

**Analysis of Inexact Trust-Region
SQP Algorithm**

*Matthias Heinkenschloss and Luis N.
Vicente*

**CRPC-TR99796
September 1999**

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

ANALYSIS OF INEXACT TRUST-REGION SQP ALGORITHMS

MATTHIAS HEINKENSCHLOSS * AND LUÍS N. VICENTE †

Abstract. In this paper we study the global convergence behavior of a class of composite-step trust-region SQP methods that 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. 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

1. Introduction. In this paper we study a class of optimization algorithms for the solution of minimization problems with nonlinear equality constraints that allow the use of inexact information. The problems we are interested in are of the form

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

where $y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $C : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m < n$, and f and C are assumed to be at least twice continuously differentiable functions. This problem formulation is motivated by applications involving discretized optimal control problems, parameter identification problems and inverse problems, and design optimization. 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.

Sequential quadratic programming (SQP) algorithms compute an approximate solution of (1.1) by solving a sequence of quadratic programming problems (QPs) in which the quadratic objective function is a model of the Lagrangian function

$$(1.2) \quad \ell(y, u, \lambda) = f(y, u) + \lambda^T C(y, u)$$

corresponding to (1.1), and in which the linear constraints are related to

$$(1.3) \quad C_y(y_k, u_k)s_y + C_u(y_k, u_k)s_u + C(y_k, u_k) = 0.$$

The matrices $C_y(y, u)$ and $C_u(y, u)$ denote the partial Jacobians with respect to y and u , respectively; the Jacobian of C is $J(y, u) = (C_y(y, u) \mid C_u(y, u))$. In the applications mentioned above, (1.3) represents the linearized state equations. We assume that $C_y(y_k, u_k) \in \mathbb{R}^{m \times m}$ is invertible. Many implementations of SQP methods, in particular the composite-step trust-region SQP algorithms considered in this paper, require the solution of linear equations of the type

$$(1.4) \quad C_y(y_k, u_k)z = d \quad \text{or} \quad C_y(y_k, u_k)^T z = d.$$

*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, 3000 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.

In optimal control, parameter identification, or optimal design problems the equations (1.4) are related to the linearized state equations and the adjoint equations, respectively. For problems governed by partial differential equations these equations are often not solved exactly using sparse matrix decompositions, but they are solved iteratively. In this case exact solutions of (1.4) are not available and one has to decide how accurately these equations have to be solved. Ideally one wants to allow for coarse but fast solutions of these systems when the SQP iterates (y_k, u_k) are far away from a solution and tighten the accuracy requirements as necessary. 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, 19]. Generalizations of the inexact Newton method concepts to the local convergence analysis of inexact SQP methods can be found, e.g., in [8, 9, 14, 18, 20]. In this paper we will give a global convergence analysis of a class of inexact trust-region SQP methods for (1.1). More specifically, we consider composite-step trust-region SQP algorithms, 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.

The main motivation of this paper is the control of inexactness arising from iterative system solves (1.4) in trust-region SQP methods. However, our assumptions on the inexactness are not based on this particular source of inexactness. Therefore our results are applicable more broadly, e.g., if finite difference approximations of derivatives are used. Our results can also be extended to affine-scaling interior-point trust-region SQP methods. More details on such an extension will be given in section 6.

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 analysis given in [10]. Our inexact trust-region SQP algorithms and their global convergence analysis 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.

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. The $l \times l$ identity matrix is denoted by I_l .

2. Trust-Region Methods for the Black-Box Formulation with Inexactness. 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 reduced problem

$$(2.1) \quad \min \hat{f}(u) = f(y(u), u).$$

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

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

and

$$\nabla^2 \hat{f}(u) = W(y, u)^T \nabla \ell(y, u, \lambda) W(y, u) \Big|_{y=y(u), \lambda=\lambda(u)},$$

where

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

and $\lambda(u)$ solves $C_y(y(u), u)\lambda = -\nabla_y f(y(u), u)$. For details see, e.g., [11, 16].

A trust-region method for the solution of (2.1) constructs a quadratic model

$$\hat{m}_k(s_u) = \hat{f}_k + \hat{g}_k^T s_u + \frac{1}{2} s_u^T \hat{H}_k s_u$$

of the function \hat{f} around the current iterate u_k . Here \hat{g}_k is an approximation of $\nabla \hat{f}(u_k)$ and \hat{H}_k replaces $\nabla^2 \hat{f}(u_k)$. Then one computes an approximate solution $(s_u)_k$ of

$$\min_{\|s_u\| \leq \Delta_k} \hat{m}_k(s_u).$$

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

$$\frac{\widehat{\text{ared}}_k}{\widehat{\text{pred}}_k} = \frac{\hat{f}(u_k) - \hat{f}(u_k + (s_u)_k)}{\hat{m}_k(0) - \hat{m}_k((s_u)_k)}$$

of actual decrease and predicted decrease.

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 $\tilde{y}(u_k)$ of $y_k = y(u_k)$ is computed by applying an iterative method to $C(y, u_k) = 0$. In this case the function evaluations $\hat{f}(u_k)$ and $\hat{f}(u_k + (s_u)_k)$ in the computation of the actual decrease will not be known exactly. Moreover, inexactness in $y(u_k)$ will also affect the gradient computation (see (2.2)). In addition, this gradient computation also requires the solution of a linear system of the form $C_y(y_k, u_k)^T z = d$. Suppose that such systems are solved iteratively, a procedure that will introduce another source of inexactness in the gradient. How does one need to control the inexactness in function values and gradients?

The influence of inexact gradient information is analyzed in [3] and in [5, § 8.4]. Let $\eta_2 \in (0, 1)$ be the constant so that the trust-region radius is reduced only if $\widehat{\text{ared}}_k / \widehat{\text{pred}}_k < \eta_2$. In [3] it is shown that if the relative gradient error satisfies

$$(2.4) \quad \|\hat{g}_k - \nabla \hat{f}(u_k)\| / \|\nabla \hat{f}(u_k)\| \leq \zeta < 1 - \eta_2,$$

then global convergence of the trust-region algorithm to stationary points can be guaranteed. In [5, § 8.4] the quotient

$$\|\hat{g}_k - \nabla \hat{f}(u_k)\| / \|\hat{g}_k\|$$

is used to measure the error in the gradient and the accuracy requirement is adjusted appropriately. The inexactness criteria in [3] and [5, § 8.4] guarantee that a successful trust-region step can be computed if the trust-region radius is sufficiently small, provided the current iterate is not a stationary point. In both analyses, [3] and [5, § 8.4], the accuracy requirements for the gradient approximation are rather weak.

The influence of inexact function evaluations is analyzed in [5, § 10.6]. It is sufficient that both errors

$$|f(\tilde{y}(u_k), u_k) - f(y(u_k), u_k)|$$

and

$$|f(\tilde{y}(u_k + (s_u)_k), u_k + (s_u)_k) - f(y(u_k + (s_u)_k), u_k + (s_u)_k)|$$

are less than $\eta_0 \widehat{\text{pred}}_k$, where $\eta_0 < \frac{1}{2}\eta_1$ and $\eta_1 \in (0, \eta_2)$ is the constant so that trust-region steps with $\widehat{\text{ared}}_k / \widehat{\text{pred}}_k < \eta_1$ are rejected. In particular, these accuracy requirements guarantee that if the ratio of actual and predicted decreases indicates acceptance of the step, i.e., if $\widehat{\text{ared}}_k / \widehat{\text{pred}}_k > \eta_1$, where $\widehat{\text{ared}}_k$ is computed with the inexact function values, then one still obtains a sufficient decrease $\hat{f}(u_k) - \hat{f}(u_k + (s_u)_k) > (\eta_1 - 2\eta_0) \widehat{\text{pred}}_k$ in the exact function values. Note also that the accuracy requirement for $f(\tilde{y}(u_k), u_k)$ depends on the trust-region step $(s_u)_k$, which is not known when $f(\tilde{y}(u_k), u_k)$ is computed the first time. Therefore, $f(\tilde{y}(u_k), u_k)$ 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. 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 (1.2) 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

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

where H_k is a symmetric approximation to the Hessian $\nabla_{xx}^2 \ell(x_k, \lambda_k)$ 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 $\|s\|_2 \leq \Delta_k$ 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 . In the context of the problem (1.1), this decomposition takes the form

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

(see [11]). Again, we assume that $C_y(x_k)$ is invertible. First, composite-step trust-region algorithms compute a so-called quasi-normal step s_k^n , where $(s_y^n)_k$ is an approximate solution of

$$\begin{aligned} \min \quad & \|C_y(x_k) s_y^n + C(x_k)\|_2, \\ \text{s.t.} \quad & \|s_y^n\|_2 \leq \Delta_k \end{aligned} \tag{3.3}$$

and $(s_u^n)_k = 0$, and then they compute a so-called tangential step s_k^t as an approximate solution of

$$(3.4) \quad \begin{aligned} \min \quad & q_k(s_k^n + s^t) \\ \text{s.t.} \quad & C_y(x_k)s_y^t + C_u(x_k)s_u = 0, \\ & \|s_u\|_2 \leq \Delta_k. \end{aligned}$$

From the constraints in (3.4) we see that $s_y^t = -C_y(x_k)^{-1}C_u(x_k)s_u$. Therefore we can write $s^t = W_k s_u$,

$$(3.5) \quad \begin{aligned} q_k(s_k^n + s^t) &= \ell(x_k, \lambda_k) + \nabla_x \ell_k^T s_k^n + \frac{1}{2} s_k^n^T H_k s_k^n \\ &\quad + (W_k^T (H_k s_k^n + \nabla_x \ell_k))^T s_u + \frac{1}{2} s_u^T W_k^T H_k W_k s_u \\ &= q_k(s_k^n) + (W_k^T \nabla q_k(s_k^n))^T s_u + \frac{1}{2} s_u^T W_k^T H_k W_k s_u, \end{aligned}$$

and pose the problem (3.4) entirely in s_u :

$$(3.6) \quad \begin{aligned} \min \quad & \hat{q}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^n) + (W_k^T \nabla q_k(s_k^n))^T s_u + \frac{1}{2} (s_u)^T W_k^T H_k W_k (s_u) \\ \text{s.t.} \quad & \|s_u\|_2 \leq \Delta_k. \end{aligned}$$

Note that the distinction between $q_k(s_k^n + s^t)$ and $\hat{q}_k(s_u)$ is only important in the inexact case. We have that $\hat{q}_k(s_u) = q_k(s_k^n + W_k s_u)$. Thus, if $s^t = W_k s_u$, then $\hat{q}_k(s_u) = q_k(s_k^n + s^t)$. However, if inexact information is used, $s^t \neq W_k s_u$ in general. Reduced SQP algorithms do not approximate the Hessian $\nabla_{xx}^2 \ell(x_k, \lambda_k)$ but the reduced Hessian $W_k^T \nabla_{xx}^2 \ell(x_k, \lambda_k) W_k$. In this case $W_k^T H_k W_k$ in (3.6) is replaced by the reduced Hessian approximation \hat{H}_k and the term $H_k s_k^n$ in $\nabla q_k(s_k^n)$ 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. In (3.4) we can also replace the trust-region constraint $\|s_u\|_2 \leq \Delta_k$ by $\|s_k^n + s^t\|_2 \leq \Delta_k$. The two choices, which lead to the decoupled and the coupled approaches, are discussed in [11]. We will address the coupled approach, which uses the trust-region constraint $\|s_k^n + s^t\|_2 \leq \Delta_k$, in section 5.2.

The subproblems (3.3) for the quasi-normal and (3.6) for the tangential step are not solved exactly. Rather coarse solutions are sufficient to guarantee basic global convergence. The quasi-normal component s_k^n is required to satisfy

$$(3.7) \quad \|s_k^n\| \leq \kappa_1 \|C_k\|$$

and

$$(3.8) \quad \|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2 \geq \kappa_2 \|C_k\| \min\{\kappa_3 \|C_k\|, \Delta_k\},$$

where κ_1 , κ_2 , and κ_3 are positive constants independent of k . The property (3.7) is actually a consequence of (3.8), see section 5.1, but it is included to be consistent with [10], where more general quasi-normal steps are allowed. Condition (3.8) is a practical consequence of the fraction of Cauchy decrease condition for the trust-region problem (3.3). The two conditions (3.7), (3.8) can be satisfied by a number of algorithms (see [10, 11] and section 5).

The tangential component $(s_u)_k$ has to satisfy a fraction of Cauchy decrease condition associated with the trust-region subproblem (3.6). In other words, $(s_u)_k$ has to provide as much decrease in the quadratic $\hat{q}_k(s_u)$ as the decrease achieved in the direction $-\nabla \hat{q}_k(0) = -W_k^T \nabla q_k(s_k^n)$ inside the trust region. It can be proved that such a condition implies

$$(3.9) \quad \hat{q}_k(0) - \hat{q}_k((s_u)_k) \geq \kappa_4 \|W_k^T \nabla q_k(s_k^n)\| \min \left\{ \kappa_5 \|W_k^T \nabla q_k(s_k^n)\|, \kappa_6 \Delta_k \right\},$$

where κ_4 , κ_5 , and κ_6 are positive constants independent of k . Algorithms that generate steps satisfying (3.9) are discussed in [10, 11] and will be commented on in section 5.

To decide about acceptance of the step $s_k = s_k^n + s_k^t$ we follow [10] and use the augmented Lagrangian merit function

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

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

$$(3.10) \quad \text{ared}(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - L(x_k + s_k, \lambda_{k+1}; \rho_k),$$

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

$$\begin{aligned} \text{pred}(s_k; \rho_k) &= L(x_k, \lambda_k; \rho_k) \\ &\quad - (q_k(s_k) + \Delta \lambda_k^T (J_k s_k + C_k) + \rho_k \|J_k s_k + C_k\|^2), \end{aligned}$$

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

$$(3.11) \quad \text{pred}(s_k; \rho_k) = \text{pred}(s_k^n, (s_u)_k; \rho_k),$$

where $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ is defined by

$$\begin{aligned} \text{pred}(s_k^n, (s_u)_k; \rho_k) &= \hat{q}_k(0) - \hat{q}_k((s_u)_k) \\ (3.12) \quad &+ q_k(0) - q_k(s_k^n) - (\Delta \lambda_k)^T (C_y(x_k)(s_y^n)_k + C_k) \\ &+ \rho_k (\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2). \end{aligned}$$

Recall that $\hat{q}_k((s_u)_k) = q_k(s_k^n + W_k(s_u)_k)$ (see (3.6)). In the case of exact computations the distinction between s_k and s_k^n in the definition of the predicted decrease is not necessary. However, this subtlety is important for the inexact computations discussed in the following section. We also note that if

$$(3.13) \quad r_k^t = C_y(x_k)(s_y^t)_k + C_u(x_k)(s_u)_k,$$

then

$$\begin{aligned} (3.14) \quad \text{pred}(s_k; \rho_k) - \text{pred}(s_k^n, (s_u)_k; \rho_k) \\ = -(\Delta \lambda_k)^T (r_k^t) - \rho_k \|r_k^t\|^2 - 2\rho_k (r_k^t)^T (C_y(x_k)(s_y^n)_k + C_k). \end{aligned}$$

Because of the requirements (3.8), (3.9) on the quasi-normal step and tangential step, respectively, we have that $\hat{q}_k(0) - \hat{q}_k((s_u)_k) + \rho_k (\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2) > 0$, provided x_k does not satisfy the first-order necessary optimality conditions (3.1). To ensure that $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ is sufficiently positive the penalty parameter ρ_k is increased if necessary.

The trust-region SQP algorithms are stated next. 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 $\alpha_1, \eta_1, \Delta_{min}$, 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 $k = 0, 1, 2, \dots$ do
 - 2.1 Compute s_k^n satisfying (3.7) and (3.8).

- 2.2 Compute $W_k^T \nabla q_k(s_k^n)$.
- 2.3 If $\|C_k\| + \|W_k^T \nabla q_k(s_k^n)\| \leq \epsilon_{tol}$, stop and return x_k as an approximate solution for problem (1.1).
- 2.4 Compute $(s_u)_k$ satisfying (3.9).
- 2.5 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.
- 2.6 Update the penalty parameter.
 If $\text{pred}(s_k^n, (s_u)_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2} \left(\|C_k\|^2 - \|C_y(x_k)s_k^n + C_k\|^2 \right)$, then set $\rho_k = \rho_{k-1}$.
 Otherwise set

$$\rho_k = \frac{2(-\hat{q}_k(0) + \hat{q}_k((s_u)_k) - q_k(0) + q_k(s_k^n) + \Delta\lambda_k^T (C_y(x_k)(s_y^n)_k + C_k))}{\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2} + \bar{\rho}.$$
- 2.7 Compute $(s_y^t)_k = -C_y(x_k)^{-1} C_u(x_k)(s_u)_k$.
- 2.8 If $\text{ared}(s_k; \rho_k) / \text{pred}(s_k^n, (s_u)_k; \rho_k) < \eta_1$, set

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

Step 2.7 of the algorithm deserves to be commented on. The u -component of the tangential step s_k^t is computed in step 2.4. If an approximation H_k of the full Hessian is used or if the coupled trust-region radius $\|s_k^n + s^t\| \leq \Delta_k$ is employed, then the y -component $(s_y^t)_k$ of the tangential step will be computed along with $(s_u)_k$ in step 2.4. If exact problem information is used, $(s_y^t)_k$ does not need to be recomputed, of course. However, if inexact problem information is used, then, in step 2.4, we may compute a y -component along with the u component that is sufficiently accurate to guarantee a condition corresponding to (3.9), but it may not be accurate enough to guarantee global convergence. In fact, our accuracy requirement on $(s_y^t)_k$ in presence of inexact problem information will depend on $\text{pred}(s_k^n, (s_u)_k; \rho_k)$. If an approximation of $(s_y^t)_k$ has already been computed in step 2.4, one has to check in step 2.7 if it satisfies the accuracy requirement on the tangential step and, if necessary, one has to recompute the inexact $(s_y^t)_k$. Thus, the addition of step 2.7 is mainly necessary for the inexact case.

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 $f, c_i, i = 1, \dots, m$ 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 $f, \nabla f, \nabla^2 f, C, J, \nabla^2 c_i, i = 1, \dots, m$, 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 3.1. *Let assumptions A.1-A.4 hold. The sequences of iterates generated by the trust-region SQP algorithms 3.1 satisfy*

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

4. Trust–Region SQP Methods with Inexactness. We now turn to the case of inexact problem information, in particular to the inexact solution of the linear systems (1.4) and to the inexact calculation of first–order derivatives of f and c with respect to y and u . This inexactness will effect the steps 2.1, 2.2, 2.4, 2.6, 2.7, and 2.8 in algorithm 3.1.

The assumptions (3.7) and (3.8) on the quasi–normal step turn 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.

Step 2.2 in algorithm 3.1 requires the calculation of the reduced gradient $\nabla \hat{q}_k(0) = W_k^T \nabla q_k(s_k^n)$. We assume that instead of the exact reduced gradient we can only compute an approximation \hat{g}_k . What is the accuracy requirement on \hat{g}_k ? If (y_k, u_k) were feasible, i.e., if $C(y_k, u_k) = 0$, then $s_k^n = 0$ (see (3.7)) and $\hat{f}(u_k) = W_k^T \nabla q_k(s_k^n)$. 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

$$\|\hat{g}_k - W_k^T \nabla q_k(s_k^n)\| \leq \zeta_1 \|\hat{g}_k\|$$

with some $\xi_1 < 1$ which is related to the parameters in the trust–region algorithm. We need a stronger condition, namely

$$(4.1) \quad \|\hat{g}_k - W_k^T \nabla q_k(s_k^n)\| \leq \xi_1 \min \{\|\hat{g}_k\|, \Delta_k\},$$

where $\xi_1 > 0$. In (4.1) 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 $\|\hat{g}_k\|$ and Δ_k .¹ The u –component of the step is now computed as the approximate solution of

$$(4.3) \quad \begin{aligned} \min \quad & \hat{m}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^n) + \hat{g}_k^T s_u + \frac{1}{2} s_u^T \widetilde{W_k^T H_k W_k} s_u \\ \text{s.t.} \quad & \|s_u\|_2 \leq \Delta_k \end{aligned}$$

(cf. (3.6)). In (4.3) the symbol \sim over $\widetilde{W_k^T H_k W_k}$ indicates that the reduced Hessian approximation may be inexact. The approximate solution $(s_u)_k$ of (4.3) computed in step 2.4 of algorithm 3.1 must provide a fraction of Cauchy decrease on this approximate model:

$$(4.4) \quad \hat{m}_k(0) - \hat{m}_k((s_u)_k) \geq \kappa_4 \|\hat{g}_k\| \min \left\{ \kappa_5 \|\hat{g}_k\|, \kappa_6 \Delta_k \right\},$$

where, as in (3.9), κ_4 , κ_5 , and κ_6 are positive constants independent of k . The reduced quadratic model \hat{m}_k in (4.4) is given by (4.3). In addition, we require that

$$(4.5) \quad (s_u)_k^T \widetilde{W_k^T H_k W_k} (s_u)_k \leq \zeta_2 \|(s_u)_k\|^2$$

for some $\zeta_2 > 0$ independent of k . If $\widetilde{W_k^T H_k W_k}$ is evaluated exactly, then (4.5) is implied by assumption A.4. A step satisfying (4.4) can be computed, e.g., using the conjugate gradient method with a trust–region modification due to Steihaug and Toint (see section 5.2).

¹We remark that, instead of (2.4), the inexactness condition

$$(4.2) \quad \|\hat{g}_k - \hat{f}(u_k)\| \leq \zeta_1 \min \{\|\hat{g}_k\|, \Delta_k\},$$

where $\xi_1 > 0$, will also give the classical \liminf global convergence result for the unconstrained problem (2.1). This may be seen using the proof in [21, Th. 4.10] and applying (4.2) in the estimate for $|\psi_k(s_k) - \nabla f(x_k)^T s_k|$ on page 278 of [21].

Suppose that we have computed s_k^n and $(s_u)_k$ so that (3.7), (3.8), (4.4), and (4.5) are satisfied. Now, we need to compute $(s_y^t)_k$. In exact computations, the tangential step s_k^t lies in the null space of the Jacobian matrix J_k . Since

$$(4.6) \quad C_y(x_k)s_y^t = -C_u(x_k)(s_u)_k$$

will be solved inexactly, s_k^t will no longer be in the null space of J_k and (3.11) will not be true anymore. Instead we have (3.14), where r_k^t is the (nonzero) residual of (4.6) as given in (3.13). For convenience of the presentation, we define

$$(4.7) \quad \text{rpred}(r_k^t; \rho_k) = -(\Delta\lambda_k)^T(r_k^t) - \rho_k \|r_k^t\|^2 - 2\rho_k(r_k^t)^T(C_y(x_k)(s_y^n)_k + C_k).$$

We measure the accuracy of s_k^t by the size of r_k^t . How accurately does s_k^t need to be computed? One approach is to enforce that the entire step $s_k = s_k^n + s_k^t$ gives a reduction in the linearized constraint residual that is a fraction of the reduction achieved by the quasi-normal step, i.e., to compute $(s_y^t)_k$ such that

$$(4.8) \quad \|C_k\|^2 - \|J_k(s_k^n + s_k^t) + C_k\|^2 \leq \zeta_3 (\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2)$$

with $\zeta_3 \in (0, 1)$. This requirement is very stringent if s_y^n achieved little decrease, which will, e.g., be the case if $\|C_k\|$ is small. However, if the point (y_k, u_k) is close to being feasible but is far away from the optimum, it seems unnecessary to maintain a high accuracy in the linear system solve for $(s_y^t)_k$. Therefore we propose a requirement for the accuracy of $(s_y^t)_k$ that takes into account both the feasibility and the optimality of (y_k, u_k) . To accomplish this, we use the predicted decrease $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ in (3.12), where \hat{q}_k is replaced by \hat{m}_k :

$$(4.9) \quad \begin{aligned} \text{pred}(s_k^n, (s_u)_k; \rho_k) &= \hat{m}_k(0) - \hat{m}_k((s_u)_k) \\ &+ q_k(0) - q_k(s_k^n) - (\Delta\lambda_k)^T(C_y(x_k)(s_y^n)_k + C_k) \\ &+ \rho_k (\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2). \end{aligned}$$

This predicted reduction depends only on s_k^n and $(s_u)_k$. Because of the requirements (3.8) and (4.4) on s_k^n and $(s_u)_k$, respectively, we have that $\hat{m}_k(0) - \hat{m}_k((s_u)_k) > 0$ and $\rho_k (\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2) > 0$, provided x_k does not satisfy the first-order necessary optimality conditions (3.1). As in the exact case we can update the penalty parameter ρ_k to ensure that $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ is sufficiently positive. However, in presence of inexactness, the quadratic model of the Lagrangian is

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

To ensure that the inexactness in the tangential step $(s_y^t)_k$ does not dominate this quadratic model, we require that

$$(4.10) \quad |\text{rpred}(r_k^t; \rho_k)| \leq \eta_0 \text{pred}(s_k^n, (s_u)_k; \rho_k),$$

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

$$(4.11) \quad \|r_k^t\| \leq \zeta_4 \Delta_k \max\{\|s_k^n\|, \|(s_u)_k\|\}$$

for some constant $\zeta_4 > 0$ independent of k .² Inequalities (4.10) and (4.11) are satisfied for the exact solution of (4.6). We note that the amount of positivity in $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ is determined

²Notice that $(s_y^t)_k = -C_y(x_k)^{-1}C_u(x_k)(s_u)_k + C_y(x_k)^{-1}r_k^t$ and that (4.11) implies

$$(4.12) \quad \|C_y(x_k)^{-1}r_k^t\| \leq \zeta_5 \Delta_k$$

by the reductions $\hat{m}_k(0) - \hat{m}_k((s_u)_k)$ and $\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2$. Therefore, if $\|C_k\|^2 - \|C_y(x_k)(s_y^n)_k + C_k\|^2$ is small, but $\hat{m}_k(0) - \hat{m}_k((s_u)_k)$ is large (which is likely the case at a point x_k that is almost feasible, but away from being optimal) the accuracy requirement on $(s_y^t)_k$ is much weaker than the requirement (4.8). Our criteria also seems to be close to the SQP philosophy which allows to trade gains in feasibility for gains in optimality. Another point worth noting is that inaccuracy in $(s_y^t)_k$ does not enter the penalty parameter update. If it were, 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.

Since s_k^n , $(s_u)_k$ and $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ are known, a step $(s_y^t)_k$ with (4.10) and (4.11) can be computed.

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 3.1, 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.

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

ALGORITHM 4.1 (Inexact Trust-Region SQP Algorithms).

- 1 The same initializations as in step 1 of algorithm 3.1.
- 2 For $k = 0, 1, 2, \dots$ do
 - i2.1 Compute s_k^n satisfying (3.7) and (3.8).
 - i2.2 Compute an approximation \hat{g}_k to $W_k^T \nabla q_k(s_k^n)$ satisfying (4.1).
 - i2.3 If $\|C_k\| + \|\hat{g}_k\| \leq \epsilon_{tol}$, stop and return x_k as an approximate solution for problem (1.1).
 - i2.4 Compute $(s_u)_k$ satisfying (4.4).
 - i2.5 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.
 - i2.6 Update the penalty parameter as in step 2.6, using $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ in (4.9).
 - i2.7 Compute $(s_y^t)_k$ so that the residual vector r_k^t satisfies (4.10) and (4.11).
 - i2.8 Compute $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ using (4.9).
If $\text{ared}(s_k; \rho_k) / \text{pred}(s_k^n, (s_u)_k; \rho_k) < \eta_1$, set

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

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

The global convergence property of the inexact trust-region SQP algorithms 3.1 is stated in the following theorem.

THEOREM 4.1. *Let assumptions A.1–A.4 hold. The sequences of iterates generated by the inexact trust-region SQP algorithms 4.1 satisfy*

$$(4.13) \quad \liminf_{k \rightarrow \infty} (\|\hat{g}_k\| + \|C_k\|) = 0.$$

for some $\zeta_5 > 0$, in other words, that the norm of the residual (tangential) step $C_y(x_k)^{-1} r_k^t$ is bounded by a constant time the trust-region radius. If we view $C_y(x_k)^{-1} r_k^t$ as a second (tangential) step, or as a spacer (tangential) step, we then identify (4.12) 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].

Furthermore, we have

$$(4.14) \quad \liminf_{k \rightarrow \infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| \right) = 0.$$

Proof. The proof of (4.13) follows the convergence analysis given in [10] with the predicted decrease used there always replaced by $\text{pred}(s_k^n, (s_u)_k; \rho_k)$ as defined in (4.9). 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), (4.3), 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 $\text{rpred}(r_k^t; \rho_k)$ 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

$$(4.15) \quad \begin{aligned} & |\text{ared}(s_k; \rho_k) - \text{pred}(s_k^n, (s_u)_k; \rho_k) - \text{rpred}(r_k^t; \rho_k)| \\ & \leq \kappa_9 \Delta_k \|s_k\| + \kappa_{10} \rho_k \|s_k\|^3 + \kappa_{11} \rho_k \|s_k\|^2 \|C_k\| \end{aligned}$$

instead of [10, L. 7.4], and

$$(4.16) \quad |\text{ared}(s_k; \rho_k) - \text{pred}(s_k^n, (s_u)_k; \rho_k) - \text{rpred}(r_k^t; \rho_k)| \leq \kappa_{12} \rho_k \Delta_k \|s_k\|$$

instead of [10, L. 7.5].

The estimates (4.15) and (4.16) can be verified as follows. Using the definitions (4.7) and (4.9), we can see that

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

With the definition (3.10) of the actual decrease, the previous identity, and $W_k^T (H_k s_k^n + \nabla_x \ell_k) = W_k^T \nabla q_k(s_k^n)$ we obtain

$$\begin{aligned} & \text{ared}(s_k; \rho_k) - (\text{pred}(s_k^n, (s_u)_k; \rho_k) + \text{rpred}(r_k^t; \rho_k)) \\ & = \ell(x_k, \lambda_k) + \rho_k \|C_k\|^2 - \ell(x_{k+1}, \lambda_{k+1}) - \rho_k \|C_{k+1}\|^2 \\ & \quad - \text{pred}(s_k^n, (s_u)_k; \rho_k) - \text{rpred}(r_k^t; \rho_k) \\ & = \ell(x_k, \lambda_k) - \ell(x_{k+1}, \lambda_k) + \ell(x_{k+1}, \lambda_k) - \ell(x_{k+1}, \lambda_{k+1}) \end{aligned}$$

$$\begin{aligned}
& + (H_k s_k^n + \nabla_x \ell_k)^T W_k(s_u)_k + \frac{1}{2} (s_u)_k^T W_k^T H_k W_k(s_u)_k + \nabla_x \ell_k^T s_k^n + \frac{1}{2} s_k^{nT} H_k s_k^n \\
& + (\widehat{g}_k - W_k^T \nabla q_k(s_k^n))^T (s_u)_k + \frac{1}{2} (s_u)_k^T W_k^T \widetilde{H}_k W_k(s_u)_k - \frac{1}{2} (s_u)_k^T W_k^T H_k W_k(s_u)_k \\
& + \Delta \lambda_k^T (J_k s_k + C_k) - \rho_k (\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2) \\
& = -\ell(x_{k+1}, \lambda_k) + q_k(s_k) - q_k(s_k) + \widehat{q}_k((s_u)_k) \\
& + (\widehat{g}_k - W_k^T \nabla q_k(s_k^n))^T (s_u)_k + \frac{1}{2} (s_u)_k^T W_k^T \widetilde{H}_k W_k(s_u)_k - \frac{1}{2} (s_u)_k^T W_k^T H_k W_k(s_u)_k \\
(4.17) \quad & + \Delta \lambda_k^T (-C_{k+1} + J_k s_k + C_k) - \rho_k (\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2).
\end{aligned}$$

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

$$(4.18) \quad | -\ell(x_{k+1}, \lambda_k) + q_k(s_k) | \leq \frac{1}{2} \|H_k - \nabla_{xx}^2 \ell(x_k + t_k^1 s_k, \lambda_k)\| \|s_k\|^2$$

with some $t_k^1 \in (0, 1)$. Using the definitions (3.2) and (3.6) of q_k and \widehat{q}_k , respectively, (3.5), and (3.13) we find that

$$\begin{aligned}
& | -q_k(s_k) + \widehat{q}_k((s_u)_k) | \\
& \leq \|H_k s_k^n - \nabla_x \ell(x_k, \lambda_k)\| \|s_k^t - W_k(s_u)_k\| + \frac{1}{2} \|H_k\| \|s_k^t\|^2 + \frac{1}{2} \|W_k^T H_k W_k\| \|(s_u)_k\|^2 \\
(4.19) \quad & \leq \|H_k s_k^n - \nabla_x \ell(x_k, \lambda_k)\| \|C_y(x_k)\| \|r_k^t\| + \frac{1}{2} \|H_k\| \|s_k^t\|^2 + \frac{1}{2} \|W_k^T H_k W_k\| \|(s_u)_k\|^2.
\end{aligned}$$

The inequalities (4.1), (4.5), and assumption A.4 give

$$\begin{aligned}
& (\widehat{g}_k - W_k^T \nabla q_k(s_k^n))^T (s_u)_k + \frac{1}{2} (s_u)_k^T W_k^T \widetilde{H}_k W_k(s_u)_k - \frac{1}{2} (s_u)_k^T W_k^T H_k W_k(s_u)_k \\
(4.20) \quad & \leq \zeta_1 \Delta_k \|(s_u)_k\| + (\zeta_2 + \|W_k^T H_k W_k\|) \|(s_u)_k\|^2.
\end{aligned}$$

Using Taylor expansion we obtain

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

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

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

If we insert (4.18)–(4.21) into (4.17) and use assumptions A.3, A.4 and (4.11), we arrive at the desired estimate (4.15) for some positive constants κ_9 , κ_{10} , and κ_{11} . Inequality (4.16) is then a direct consequence of inequality (4.15) 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 (4.15) with (4.10), yields

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

Similarly, combining (4.16) with (4.10), gives

$$(4.23) \quad |\text{ared}(s_k; \rho_k) - \text{pred}(s_k^n, (s_u)_k; \rho_k)| \leq \kappa_{12} \rho_k \Delta_k \|s_k\| + \eta_0 |\text{pred}(s_k^n, (s_u)_k; \rho_k)|.$$

The estimates (4.22) and (4.23) are used in the analysis only when rejection occurs in step i2.8. If s_k is rejected, we know that

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

which in our inexact context implies

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

Thus, when the estimate (4.16) is required, we obtain

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

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 (4.15).

The proof of (4.14) follows from the conjunction of (4.13) with (4.1) and (3.7). \square

5. Implementation in the Presence of Inexactness.

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 conditions (3.7) and (3.8). The property (3.7) is a consequence of (3.8). In fact, using $\|C_y(x_k)(s_k^n)_y + C_k\| \leq \|C_k\|$ and the boundedness of $\{C_y(x_k)^{-1}\}$ we find that

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

Whether the property (3.8) holds depends on the way in which the quasi-normal component is computed. We mention a few methods that are suitable for the large-scale case, that do not require the matrix $C_y(x_k)$ in explicit form, and that guarantee (3.8).

If $\|s_k^n\| \leq \Delta_k$ satisfies the fraction of Cauchy decrease condition

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

then a classical result due to Powell [22, Th. 4] (see also [5, § 6.3], [21, L. 4.8]) shows that (3.8) is satisfied. The papers [15], [24] describe two iterative methods based on Krylov subspaces for the computation of steps $(s_y^n)_k$ satisfying

$$\|C_k\|^2 - \|C_y(x_k)(s_k^n)_y + C_k\|^2 \geq \beta \left(\|C_k\|^2 - \|C_y(x_k)(s_y^n)_* + C_k\|^2 \right),$$

where $(s_y^n)_*$ is the solution of (3.3). In particular these steps also satisfy (3.8). The iterative method in [24] uses a restart technique that allows specification of storage limitations by the user, which is important for large scale problems. The iterative methods in [15] and in [24] 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 $C_y(x_k)^T u$ is more expensive than the application of $C_y(x_k)v$, and therefore it may be more efficient to use methods that avoid the use of $C_y(x_k)^T u$. In this case one can apply nonsymmetric Krylov subspace methods based on minimum residual approximations, such as GMRES [23]. 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 is used and if

$$(5.2) \quad \frac{1}{2} C_k^T \left(C_y(x_k)^T + C_y(x_k) \right) C_k \geq \beta \|C_k\|^2$$

holds with $\beta > 0$, then (3.8) 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 [27].

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

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

The step s_k^n also satisfies (3.8) (see [27]).

5.2. Computation of the tangential component. An approximate solution s_u of (4.3) that satisfies (4.4) can be computed, e.g., using the conjugate gradient (cg) method with a modification as suggested by Steihaug [25] and Toint [26]. Here the cg method with starting value $s_u = 0$ is applied to the minimization of \hat{m}_k . The conjugate gradient method is stopped if an approximate minimum of the quadratic model \hat{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 \hat{m}_k and therefore (4.4) 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 $W_k^T H_k W_k = \hat{H}_k$, then the conjugate gradient method ensures that \hat{m}_k decreases monotonically and (4.4) 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 \hat{m}_k at the first Steihaug–Toint cg iterate s_u^1 and at the final Steihaug–Toint cg iterate s_u^f . If $\hat{m}_k(s_u^f) \leq \hat{m}_k(s_u^1)$, then $(s_u)_k = s_u^f$; otherwise $(s_u)_k = s_u^1$.

In section 3 we have already pointed out that the trust-region constraint $\|s_u\|_2 \leq \Delta_k$ in (3.4) can be replaced by $\|s_k^n + s^t\|_2 \leq \Delta_k$. This leads to the so-called coupled approach. The trade-offs between the two choices are discussed in [11, § 5.2.2]. In the coupled trust-region approach we replace (3.6) by

$$(5.3) \quad \begin{aligned} \min \quad & \hat{q}_k(s_u) \stackrel{\text{def}}{=} q_k(s_k^n) + (W(x_k)^T \nabla q_k(s_k^n))^T s_u + \frac{1}{2} s_u^T W(x_k)^T H_k W(x_k) s_u \\ \text{s.t.} \quad & \left\| \begin{pmatrix} (s_y^n)_k \\ 0 \end{pmatrix} + \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k) s_u \\ s_u \end{pmatrix} \right\|_2 \leq \Delta_k. \end{aligned}$$

and formulate the trust-region constraint in (4.3) according to (5.3). In that case the computation of $C_y(x_k)^{-1} C_u(x_k)$ can be performed inexactly. In the coupled case, the first part of step i2.8 in algorithm 4.1 has to be replaced by

i2.8 If $\text{ared}(s_k; \rho_k) / \text{pred}(s_k^n, (s_u)_k; \rho_k) < \eta_1$, set

$$\delta_{k+1} = \alpha_1 \max \{ \|s_k^n\|, \|s_k\| \}$$

and reject s_k .

6. Conclusions. In this paper we have investigated composite-step trust-region SQP algorithms for the solution of a class of optimization problems with inexact problem information. 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 (1.4) 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 therefore our results are applicable more broadly. The trust-region SQP methods for (1.1) can be generalized to problems with additional simple bounds on u using affine-scaling interior-point methods due to [4]. Such a generalization (assuming exact problem information) is given in [11]. The analysis of inexact trust-region SQP algorithms performed in this paper can also be generalized to affine-scaling interior-point trust-region SQP algorithms. In fact, the predecessor [17] of this paper contains many of the technical details, although the assumptions on the inexactness made in [17] are stronger than those in this paper.

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 Newton–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] T. F. COLEMAN AND Y. LI, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optimization, 6 (1996), pp. 418–445.
- [5] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Trust–Region Methods*, SIAM, Philadelphia, 1999. To appear.
- [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. xx–yy.
- [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] P. DEUFLHARD, *Global inexact Newton methods for very large scale nonlinear problems*, Impact of Computing in Science and Engineering, 4 (1991), pp. 366–393.
- [13] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optimization, 4 (1994), pp. 393–422.
- [14] 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.
- [15] G. H. GOLUB AND U. VON MATT, *Quadratically constrained least squares and quadratic problems*, Numerische Mathematik, 59 (1991), pp. 561–580.

- [16] M. HEINKENSCHLOSS, *Projected sequential quadratic programming methods*, SIAM J. Optimization, 6 (1996), pp. 373–417.
- [17] 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>.
- [18] 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.
- [19] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.
- [20] F. LEIBFRTZ AND E. W. SACHS, *Inexact SQP interior–point methods and large–scale control problems*, SIAM J. Control and Optimization, (1999). To appear.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] D. C. SORENSEN, *Minimization of a large scale quadratic function subject to an spherical constraint*, SIAM J. Optimization, 7 (1997), pp. 141–161.
- [25] T. STEihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
- [26] 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.
- [27] 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.