Trust-Region Interior-Point Algorithms for a Class of Nonlinear Programming Problems

J. E. Dennis
Matthias Heinkenschloss
Luís Vicente

CRPC-TR95512-S
January 1995

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

Revised: May, 1995

TRUST–REGION INTERIOR–POINT SQP ALGORITHMS FOR A CLASS OF NONLINEAR PROGRAMMING PROBLEMS

J. E. DENNIS [*], MATTHIAS HEINKENSCHLOSS [†] AND LUíS N. VICENTE [‡]

Abstract. Trust–region interior–point SQP algorithms for the solution of minimization problems with equality constraints and simple bounds on some of the variables are presented. These nonlinear programs arise from the discretization of many optimal control problems. The algorithms are designed to take advantage of this structure; in particular, provision is made for user–supplied linearized state equation solvers.

The algorithms keep strict feasibility with respect to the bound constraints and use trust–region techniques to ensure global convergence. First–order convergence of these algorithms is proved for very mild conditions on the trial steps. The results given here include as special cases current results both for equality constraints and for simple bounds.

Numerical solution of an optimal control problem governed by a nonlinear heat equation is reported.

Keywords. nonlinear programming, SQP, trust–region methods, interior–point algorithms, Dikin–Karmarkar scaling, Coleman–Li scaling, simple bounds, optimal control.

AMS subject classifications. 49M37, 90C06, 90C30

1. Introduction.

In this paper we are interested in the solution of the following minimization problem

$$\begin{array}{rl} \text{minimize} & f(y,u) \\ \text{subject to} & C(y,u) = 0, \\ & u \in \mathcal{B} = \{u : \; a \le u \le b\}, \end{array} \tag{1}$$

where $y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$, $a \in (\mathbb{R} \cup \{-\infty\})^{n-m}$, $b \in (\mathbb{R} \cup \{+\infty\})^{n-m}$, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $C : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, $m < n$. The functions f and C are assumed to be at least continuously differentiable. This minimization problem often arises from the discretization of optimal control problems. In this case y is the vector of state variables, u is the vector of control variables, and $C(y,u) = 0$ is the discretized state equation. Other applications include design optimization and parameter identification problems. For convenience we write

$$x = \begin{pmatrix} y \\ u \end{pmatrix}.$$

Our interest is to design sequential quadratic programming (SQP) algorithms for this problem that use a affine–scaling interior–point approach to keep strict feasibility with respect to the simple bounds and that take advantage of the strong global convergence properties of trust–region methods for equality–constrained optimization.

[*] Department of Computational and Applied Mathematics, Rice University, Houston Texas 77251, USA, E-Mail: dennis@rice.edu. Support of this author has been provided by DOE DE–FG005–86ER25017, CRPC CCR–9120008 and AFOSR–F49620–9310212.

[†] Virginia Polytechnic Institute and State University, Interdisciplinary Center for Applied Mathematics and Department of Mathematics, Blacksburg, VA 24061–0531, USA, E-Mail: na.heinken@na-net.ornl.gov. This author was supported by the NSF under Grant DMS–9403699 and in part by the AFOSR under Grant F49620–93–1–0280.

[‡] Department of Computational and Applied Mathematics, Rice University, Houston Texas 77251, USA, E-Mail: lvicente@rice.edu. Support of this author has been provided by INVOTAN, CCLA and FLAD (Portugal) and by Department of Energy grant DE–FG03–93ER25178.

The interior–point approach used to maintain strict feasibility has the flavor of the Dikin-Karmarkar scaling. We propose two interior–point approaches. In the first one, the trust region is of the Dikin–Karmarkar type at a constraining bound, although we allow the trust radius to be greater than one. This is similar to the approaches given by Coleman and Li [4] and Dennis and Vicente [8] for minimization problems with simple bounds. The second maintains the trust region in the unscaled variables and has been suggested by Dennis and Vicente [8]. In both, the quadratic models are scaled and the step is inside the trust region and strictly inside $\mathcal{B}$. Other affine–scaling approaches for nonlinear programming have recently been proposed by Bonnans and Pola [1], Coleman and Liu [5], Li [18], [19], and Plantenga [22].

The trust–region SQP algorithms decompose a step s in $s = s^{\mathsf{n}} + s^{\mathsf{t}}$, where s^{n} is the quasi–normal component associated with the trust–region subproblem for the linearized constraints and s^{t} is the tangential component computed from the trust–region subproblem for the Lagrangian reduced to the tangent subspace. This approach is like those recently followed by several authors (see references [6], [7], [10], [17], and [21]). The components s^{n} and s^{t} can be computed in several ways (see [17] and [22]). However we have in mind nonlinear programs of the type (1) that come from the discretization of optimal control problems and where $C(y,u) = 0$ is a discretized partial differential equation. The algorithms suggested in this paper take advantage of this structure to compute the components s^{n} and s^{t} of a step s.

We prove that any sequence of iterates produced by our trust–region interior–point (TRIP) SQP algorithms has a subsequence for which the first–order Karush–Kuhn–Tucker conditions are satisfied in the limit. It is important to note that this result is obtained under very mild assumptions on the trial steps, and that the sequence of approximations to the full or reduced Hessian matrix of the Lagrangian is assumed only to be bounded. We use the theory developed by Dennis, El–Alem and Maciel [6] for equality–constrained optimization.

We implemented the TRIP SQP algorithms and solved a discretized optimal control problem governed by a nonlinear heat equation. We tested several alternatives and the numerical results given in Section 8 are quite promising.

A projected sequential quadratic programming method to solve (1) that also exploits the discretized optimal control structure has recently been proposed by Heinkenschloss [12]. His algorithm uses line searches and requires an approximation to the reduced Hessian matrix.

Reduced sequential quadratic programming methods for the solution of (1) without inequality constraints have been analyzed in a Hilbert space setting by Kupfer [15] and applied to control and parameter identification problems by Kunisch and Sachs [14] and by Kupfer and Sachs [16].

We review the notation used in this paper. The Lagrangian function $\ell : \mathbb{R}^{n+m} \longrightarrow \mathbb{R}^n$ associated with the equality constraints $C(x) = 0$ is given by $\ell(x,\lambda) = f(x) + \lambda^T C(x)$, where $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers. The Jacobian matrix of $C(x)$ is denoted by $J(x)$. We use subscripted indices to represent the evaluation of a function at a particular point of the sequences $\{x_k\}$ and $\{\lambda_k\}$. For instance, f_k represents $f(x_k)$, and ℓ_k is the same as $\ell(x_k, \lambda_k)$. The vector and matrix norms used are the ℓ_2 norms, and I_l represents the identity matrix of order l. Also $(z)_y$ and $(z)_u$ represent the subvectors of $z \in \mathbb{R}^n$ corresponding to the y and u components, respectively.

This paper is organized as follows. In Section 2 we present the structure of the problem, and in Section 3 we use this structure to derive a form of the first–order

Karush–Kuhn–Tucker conditions. In Sections 4 and 5 we describe our TRIP SQP algorithms. The convergence theory for these algorithms is given in Section 6. In Section 7 we describe algorithms to compute trial steps and the multiplier estimates. The numerical results are reported in Section 8. Finally, in Section 9 we end the paper with conclusions and a discussion of future work.

2. Structure of the minimization problem. We seek to design algorithms that exploit the structure of this problem. For this purpose let us partition the Jacobian matrix of $C(x)$ as

$$J(x) = \begin{pmatrix} C_y(x) & C_u(x) \end{pmatrix},$$

corresponding to the partitioning of x in y and u. Here $C_y(x) \in \mathrm{I\!R}^{m\times m}$ and $C_u(x) \in \mathrm{I\!R}^{m\times(n-m)}$. We assume that the matrix $C_y(x)$ is nonsingular for every x with $a \le u \le b$. Often, efficient linear system solvers are available from the application specialist for the coefficient matrix $C_y(x)$ and its transpose. We designed our algorithms to allow the use of these system solvers rather than incorporate a particular method into the optimization algorithm. This is important for many applications since the solution of these systems is very time consuming and the coefficient matrix is often not available in the explicit form. Of course, we also can furnish solvers for these systems, but our approach allows more flexibility. We also need to assume that we can multiply the matrices $C_u(x)$ and $C_u(x)^T$ times a given vector.

We say that s satisfies the linearized state equations at x if $J(x)s = -C(x)$ or equivalently if

$$\begin{pmatrix} C_y(x) & C_u(x) \end{pmatrix} \begin{pmatrix} s_y \\ s_u \end{pmatrix} = -C(x).$$

Here s is partitioned as

$$s = \begin{pmatrix} s_y \\ s_u \end{pmatrix},$$

and $s_y \in \mathrm{I\!R}^m$ and $s_u \in \mathrm{I\!R}^{n-m}$. One way to compute such an s is to set s_u to zero and to calculate the corresponding s_y:

$$s = \begin{pmatrix} -C_y(x)^{-1}C(x) \\ 0 \end{pmatrix}.$$

We also are interested in finding a matrix $W(x)$ whose columns form a basis for the null space $\mathcal{N}(J(x))$ of $J(x)$. Such a basis can be given by

$$W(x) = \begin{pmatrix} -C_y(x)^{-1}C_u(x) \\ I_{n-m} \end{pmatrix}.$$

One can see that matrix–vector multiplications of the form $W(x)^T s$ and $W(x)s_u$ involve only the solution of linear systems with the matrices $C_y(x)$ and $C_y(x)^T$.

3. First–order Karush–Kuhn–Tucker conditions. A point x_* satisfies the first–order Karush–Kuhn–Tucker (KKT) conditions if there exist $\lambda_* \in \mathbb{R}^m$ and $\mu_*^a, \mu_*^b \in \mathbb{R}^{n-m}$ such that

$$\begin{aligned}
& C(x_*) = 0, \\
& a \le u_* \le b, \\
& \begin{pmatrix} \nabla_y f(x_*) \\ \nabla_u f(x_*) \end{pmatrix} + \begin{pmatrix} C_y(x_*)^T \lambda_* \\ C_u(x_*)^T \lambda_* \end{pmatrix} - \begin{pmatrix} 0 \\ \mu_*^a \end{pmatrix} + \begin{pmatrix} 0 \\ \mu_*^b \end{pmatrix} = 0, \\
& ((u_*)_i - a_i)\,(\mu_*^a)_i = (b_i - (u_*)_i)\,(\mu_*^b)_i = 0, \; i = 1, \ldots, n-m, \\
& \mu_*^a \ge 0, \; \mu_*^b \ge 0.
\end{aligned}$$

These KKT conditions are necessary conditions for x_* to be a local solution of (1) since the invertibility of $C_y(x_*)$ and the form of the bound constraints imply the linear independence of the active constraints. We can use the structure of the problem to rewrite the first–order KKT conditions:

$$\begin{aligned}
& C(x_*) = 0, \\
& a \le u_* \le b, \\
& \lambda_* = -C_y(x_*)^{-T} \nabla_y f(x_*), \\
& a_i < (u_*)_i < b_i \quad \Longrightarrow \quad (\nabla_u \ell(x_*, \lambda_*))_i = 0, \\
& (u_*)_i = a_i \quad \Longrightarrow \quad (\nabla_u \ell(x_*, \lambda_*))_i \ge 0, \\
& (u_*)_i = b_i \quad \Longrightarrow \quad (\nabla_u \ell(x_*, \lambda_*))_i \le 0.
\end{aligned}$$

One can obtain a useful form of the first–order KKT conditions by noting that

$$\begin{aligned}
\nabla_u \ell(x_*, \lambda_*) &= \nabla_u f(x_*) + C_u(x_*)^T \lambda_* \\
&= \nabla_u f(x_*) - C_u(x_*)^T C_y(x_*)^{-T} \nabla_y f(x_*) \\
&= W(x_*)^T \nabla f(x_*).
\end{aligned}$$

In other words, $\nabla_u \ell(x_*, \lambda_*)$ is just the reduced gradient corresponding to the u variables. Hence x_* is a first–order KKT point if

$$\begin{aligned}
& C(x_*) = 0, \\
& a \le u_* \le b, \\
& a_i < (u_*)_i < b_i \quad \Longrightarrow \quad \left(W(x_*)^T \nabla f(x_*) \right)_i = 0, \\
& (u_*)_i = a_i \quad \Longrightarrow \quad \left(W(x_*)^T \nabla f(x_*) \right)_i \ge 0, \\
& (u_*)_i = b_i \quad \Longrightarrow \quad \left(W(x_*)^T \nabla f(x_*) \right)_i \le 0.
\end{aligned}$$

Now we adapt the idea of Coleman and Li [4] to this context and define $D(u)$ as a diagonal matrix whose diagonal elements are given by

$$(D(u))_{ii} = \begin{cases} (b-u)_i & \text{if } \left(W(x)^T\nabla f(x)\right)_i < 0 \text{ and } b_i < +\infty, \\ 1 & \text{if } \left(W(x)^T\nabla f(x)\right)_i < 0 \text{ and } b_i = +\infty, \\ (u-a)_i & \text{if } \left(W(x)^T\nabla f(x)\right)_i \geq 0 \text{ and } a_i > -\infty, \\ 1 & \text{if } \left(W(x)^T\nabla f(x)\right)_i \geq 0 \text{ and } a_i = -\infty, \end{cases} \tag{2}$$

for $i = 1, \ldots, n-m$. In the following proposition we give the form of the first–order KKT conditions that we use in this paper. To us, they indicate the suitability of (2) as a scaling for (1).

PROPOSITION 3.1. *The point x_* satisfies the first–order KKT conditions if*

$$\begin{aligned} &C(x_*) = 0, \\ &a \leq u_* \leq b, \\ &D(u_*)W(x_*)^T\nabla f(x_*) = 0. \end{aligned}$$

4. Decomposition of the step. The algorithms that we propose generate a sequence of iterates $\{x_k\}$ where

$$x_k = \begin{pmatrix} y_k \\ u_k \end{pmatrix},$$

and u_k is strictly feasible, i.e., $a < u_k < b$. At iteration k we are given x_k and we need to compute a trial step s_k. If s_k is accepted, we set $x_{k+1} = x_k + s_k$. Otherwise we set x_{k+1} to x_k. Each trial step s_k is decomposed as $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$, where s_k^{n} is called the quasi–normal component and s_k^{t} is the tangential component.

4.1. The quasi–normal component. Let δ_k be the trust radius at iteration k, and let r be a positive real number. We discuss the role of r later when we define the tangential component. The quasi–normal component s_k^{n} is related to the trust–region subproblem for the linearized constraints

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|J_k s^{\mathsf{n}} + C_k\|^2 \\ &\text{subject to} && \|s^{\mathsf{n}}\| \leq r\delta_k, \end{aligned}$$

and it is required to have the form

$$s_k^{\mathsf{n}} = \begin{pmatrix} (s_k^{\mathsf{n}})_y \\ 0 \end{pmatrix}. \tag{3}$$

Thus the displacement along s_k^{n} is made only in the y variables, and as a consequence, x_k and $x_k + s_k^{\mathsf{n}}$ coincide in their u components. Furthermore, the trust–region subproblem introduced above can be rewritten as

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|C_y(x_k)(s^{\mathsf{n}})_y + C_k\|^2 \\ &\text{subject to} && \|(s^{\mathsf{n}})_y\| \leq r\delta_k. \end{aligned}$$

We also need to impose on the quasi–normal component the following conditions:

$$\|s_k^{\mathsf{n}}\| \leq \kappa_1 \|C_k\| \tag{4}$$

and

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \geq \kappa_2 \|C_k\| \min\{\kappa_3 \|C_k\|, r\delta_k\}, \tag{5}$$

where κ_1, κ_2 and κ_3 are positive constants independent of k. In Section 7.1, we describe several ways of computing the quasi–normal component that satisfy the requirements (3), (4), and (5).

4.2. The tangential component. The tangential component s_k^{t} lies in the null space $\mathcal{N}(J_k)$. For this purpose we consider the matrix

$$W_k = \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k) \\ I_{n-m} \end{pmatrix},$$

whose columns form a basis for $\mathcal{N}(J_k)$. Thus we can write s_k^{t} as $W_k \bar{s}_k^{\mathsf{t}}$ for some $\bar{s}_k^{\mathsf{t}}$ in $\mathbb{R}^{n-m}$ and s_k as

$$s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}} = s_k^{\mathsf{n}} + W_k \bar{s}_k^{\mathsf{t}} = \begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1} C_u(x_k) \bar{s}_k^{\mathsf{t}} \\ \bar{s}_k^{\mathsf{t}} \end{pmatrix}.$$

From this it is clear that the $(s_k)_y$ and $(s_k)_u$ components of the trial step s_k are given by

$$\begin{aligned} (s_k)_y &= (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1} C_u(x_k) \bar{s}_k^{\mathsf{t}} = (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1} C_u(x_k)(s_k)_u, \\ (s_k)_u &= \bar{s}_k^{\mathsf{t}}. \end{aligned}$$

4.2.1. The decoupled trust–region approach. Our approach to compute the tangential component $W_k(s_k)_u$ begins like SQP. First we consider the local quadratic programming subproblem

$$\begin{array}{ll} \text{minimize} & q_k(s_k^{\mathsf{n}} + W_k s_u) \\ \text{subject to} & \sigma_k(a - u_k) \leq s_u \leq \sigma_k(b - u_k), \end{array}$$

gotten by building a quadratic model

$$q_k(s) = \ell_k + \nabla_x \ell_k{}^T s + \frac{1}{2} s^T H_k s$$

of $\ell(x_k + s, \lambda)$ about (x_k, λ_k), where H_k is an approximation to the Hessian matrix $\nabla_{xx}^2 \ell(x_k, \lambda_k)$ and λ_k represents the multiplier estimate. Here $\sigma_k \in [\sigma, 1)$ ensures that the solution to the subproblem remains strictly feasible with respect to the box constraints. The parameter $\sigma \in (0,1)$ is fixed for all k. A trivial manipulation shows that

$$q_k(s_k^{\mathsf{n}} + W_k s_u) = q_k(s_k^{\mathsf{n}}) + \bar{g}_k^T s_u + \frac{1}{2} s_u{}^T W_k^T H_k W_k s_u, \tag{6}$$

with $\bar{g}_k = W_k^T \nabla q_k(s_k^{\mathsf{n}}) = W_k^T (H_k s_k^{\mathsf{n}} + \nabla f_k)$.

We rewrite this quadratic problem in a basis $\hat{s}_u = \bar{D}_k^{-1} s_u$, where $\bar{D}_k$ is a diagonal matrix whose diagonal elements are given by

$$(\bar{D}_k)_{ii} = \begin{cases} (b - u_k)_i & \text{if } \bar{g}_i < 0 \text{ and } b_i < +\infty, \\ 1 & \text{if } \bar{g}_i < 0 \text{ and } b_i = +\infty, \\ (u_k - a)_i & \text{if } \bar{g}_i \geq 0 \text{ and } a_i > -\infty, \\ 1 & \text{if } \bar{g}_i \geq 0 \text{ and } a_i = -\infty, \end{cases} \tag{7}$$

for $i = 1, \ldots, n - m$.

This gives the local quadratic programming subproblem:

$$\begin{array}{rl} \text{minimize} & q_k(s_k^{\mathsf{n}} + W_k \bar{D}_k \hat{s}_u) \\ \text{subject to} & \sigma_k \bar{D}_k^{-1}(a - u_k) \leq \hat{s}_u \leq \sigma_k \bar{D}_k^{-1}(b - u_k), \end{array}$$

gotten by building a quadratic model

$$q_k(s_k + W_k \bar{D}_k \hat{s}_u) = q_k(s_k^{\mathsf{n}}) + (\bar{D}_k \bar{g}_k)^T \hat{s}_u + \frac{1}{2} \hat{s}_u^T \bar{D}_k W_k^T H_k W_k \bar{D}_k \hat{s}_u.$$

In this subproblem there is an *explicit* scaling given by $\bar{D}_k$. For instance, the steepest–descent direction in the ℓ_2 norm is given by $-\bar{D}_k \bar{g}_k$.

We would like to minimize this quadratic function over a trust region with the requirement that $u_k + (s_k)_u$ has to be strictly feasible. Although we do this in the original basis s_u so that we can always work with the same variables, we have inherited the scaling that is used in the basis $\hat{s}_u$. The reference trust–region subproblem that we consider, written in the original basis, is the following:

$$\begin{array}{rl} \text{minimize} & q_k(s_k^{\mathsf{n}} + W_k s_u) \\ \text{subject to} & \|S_k^{-1} s_u\| \leq \delta_k, \\ & \sigma_k(a - u_k) \leq s_u \leq \sigma_k(b - u_k), \end{array} \tag{8}$$

where δ_k is the trust radius, and S_k is a $(n - m) \times (n - m)$ nonsingular matrix. This subproblem is *implicitly* scaled. Here the scaling is of the form $\bar{D}_k^2$. For instance the direction $-D_k \bar{g}_k$ given in the $\hat{s}_u$ variables is now defined as $-\bar{D}_k(\bar{D}_k \bar{g}_k) = -\bar{D}_k^2 \bar{g}_k$ in the s_u variables.

We discuss two alternatives for S_k now.

If we continue to follow the affine–scaling idea, then we use the ellipse defined at each iteration in the original s_u coordinates by the ℓ_2 norm on these new coordinates $\hat{s}_u$ to help to enforce the bounds. In other words, we would choose $S_k = \bar{D}_k$, and the shape of the trust region would be ellipsoidal in the original basis. This has been suggested in [4].

This substitution of one ellipsoidal constraint for all the bound constraints was a prime motivation for interior–point methods. However from the beginning of the computational study of interior–point methods, it was found to be important to allow steps to past the boundary of this ellipsoid, as long as they still satisfy the subproblem bound constraints. This translates here to saying that if the trust region is to have the ellipsoidal shape, then the trust radius should be allowed to exceed one, and so the trust region really is not used to enforce the bound constraints.

The motivation for our second choice of S_k is that there is no reason to use the scaling to define the shape of the trust region if it is not useful for enforcing the bounds. In fact, there are even more good reasons not to use it here than in the linear programming problem. One of the most important is that for nonlinear programs u_* may lie strictly inside $\mathcal{B}$. This happens in problems where the bounds are really to define the region of interest. Hence our second choice of S_k is the identity matrix of order $n - m$. This has been suggested in [8].

4.2.2. The coupled trust–region approach. In the decoupled trust–region approach we impose the trust region separately on the y component of the quasi–normal step and on the u component of the tangential step. In this case there is no need to restrict the parameter r. For example, $r = 1$ is a reasonable choice.

The approach we follow in this section forces the whole trial step $s_k = s_k^{\mathsf{n}} + W_k(s_k)_u$ to lie inside the trust region of radius δ_k. In this case we need to choose r in $(0,1)$. The reference trust–region subproblem is given by

$$\begin{array}{lll} & \text{minimize} & q_k(s_k^{\mathsf{n}} + W_k s_u) \\ (9) & \text{subject to} & \left\| \begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1} C_u(x_k) s_u \\ S_k^{-1} s_u \end{pmatrix} \right\| \leq \delta_k, \\ & & \sigma_k(a - u_k) \leq s_u \leq \sigma_k(b - u_k), \end{array}$$

where S_k plays the role described in the decoupled approach. This approach is similar to those followed by many other authors for equality–constrained optimization (see references [6], [7], [10], [17], and [21]).

4.2.3. What to impose on the tangential component. Our aim will be to impose as little as possible on the tangential components. This suggests that we consider analogs for the subproblems (8) and (9) of the *fraction of Cauchy decrease* conditions for the unconstrained minimization problem.

First we consider the decoupled trust–region subproblem (8). The Cauchy step c_k^{d} is defined as the solution of

$$\begin{array}{ll} \text{minimize} & q_k(s_k^{\mathsf{n}} + W_k s_u) \\ \text{subject to} & \|S_k^{-1} s_u\| \leq \delta_k, \quad s_u \in span\{-\bar{D}_k^2 \bar{g}_k\}, \\ & \sigma_k(a - u_k) \leq s_u \leq \sigma_k(b - u_k). \end{array}$$

As in many trust–region algorithms, we require $(s_k)_u$ to give a decrease on $q_k(s_k^{\mathsf{n}} + W_k s_u)$ smaller than a uniform fraction of the decrease given by c_k^{d} for the same function $q_k(s_k^{\mathsf{n}} + W_k s_u)$. This condition is often called fraction of Cauchy decrease, and in this case is

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k(s_k)_u) \geq \beta\left(q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k c_k^{\mathsf{d}})\right), \tag{10}$$

where β is positive and fixed across all iterations. It is not difficult to see that dogleg or conjugate–gradient algorithms can compute trial steps $(s_k)_u$ conveniently that satisfy condition (10) with $\beta = 1$. We leave these issues to Section 7.2.

In a similar way, the component $(s_k)_u$ satisfies a fraction of Cauchy decrease for the coupled trust–region subproblem (9) if

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k(s_k)_u) \ge \beta\left(q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k c_k^{\mathsf{c}})\right) \tag{11}$$

for some β independent of k, where the Cauchy step c_k^{c} is the solution of

$$\begin{array}{ll} \text{minimize} & q_k(s_k^{\mathsf{n}} + W_k s_u) \\ \text{subject to} & \left\| \begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1}C_u(x_k)s_u \\ S_k^{-1}s_u \end{pmatrix} \right\| \le \delta_k, \quad s_u \in span\{-\bar{D}_k^2\bar{g}_k\}, \\ & \sigma_k(a-u_k) \le s_u \le \sigma_k(b-u_k). \end{array}$$

In Section 7.2 we show how to use conjugate–gradients to compute components $(s_k)_u$ satisfying the condition (11).

One final comment is in order. In the coupled approach the Cauchy step c_k^{c} was defined along the direction $-\bar{D}_k^2\bar{g}_k$. To simplify this discussion, suppose that there are no bounds on u. In this case the trust–region constraint is of the form $\|s_k^{\mathsf{n}} + W_k s_u\| \le \delta_k$, where W_k gives the trust region an ellipsoidal shape. The steepest descent direction for the quadratic (6) in the norm $\|W_k \cdot\|$ is given by $-(W_k^T W_k)^{-1}\bar{g}_k$. The reason why we drop the term $(W_k^T W_k)^{-1}$ is that in many applications there is no reasonable way to solve systems with $W_k^T W_k$. We will show in Section 7.2 how this affects the use of conjugate gradients (see Remark 7.2). Finally, we point out that this problem does not arise if the decoupled approach is used.

4.3. Reduced and full Hessians. In the previous section we considered an approximation H_k to the full Hessian. The algorithms and theory presented in this paper are also valid if we use an approximation $\widehat{H}_k$ to the reduced Hessian $W_k^T H_k W_k$. In this case we set

$$H_k = \begin{pmatrix} 0 & 0 \\ 0 & \widehat{H}_k \end{pmatrix}, \tag{12}$$

and due to the form of W_k, we have

$$W_k^T H_k W_k = \widehat{H}_k.$$

This allows us to see the expansion (6) in the context of a reduced Hessian approximation.

For the algorithms with reduced Hessian approximation the following observations are useful:

$$\begin{array}{rcl} H_k d & = & (0, \widehat{H}_k d_u), \\ d^T H_k d & = & d_u^T \widehat{H}_k d_u, \\ W_k^T H_k d & = & \widehat{H}_k d_u. \end{array} \tag{13}$$

5. Outline of the algorithms and general assumptions. We need to introduce a merit function and the corresponding actual and predicted reductions. The merit function used is the augmented Lagrangian

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

We follow [6] and define the actual decrease at iteration k as

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

and the predicted decrease as

$$pred(s_k;\rho_k) = L(x_k,\lambda_k;\rho_k) - \left(q_k(s_k) + \Delta\lambda_k^T(J_k s_k + C_k) + \rho_k\|J_k s_k + C_k\|^2\right),$$

with $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.

To decide whether to accept or reject a trial step s_k, we evaluate the ratio

$$\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)},$$

and to update the penalty parameter ρ_k we use the scheme proposed by El–Alem [9]. Other schemes to update the penalty parameter have been suggested in [10] and [17].

We can describe now the main procedures of the trust–region interior–point SQP algorithms and leave the computation of s_k^{n} and $(s_k)_u$ to Section 7. In this section we also suggest convenient multiplier updates.

ALGORITHM 5.1 (TRUST–REGION INTERIOR–POINT SQP ALGORITHMS).

```
1 Choose $x_0$ such that $a < u_0 < b$, pick $\delta_0 > 0$, and calculate $\lambda_0$. Set $\rho_{-1} \ge 1$
  and $\epsilon_{tol} > 0$. Choose $\alpha_1, \eta_1, \sigma, \delta_{min}, \delta_{max}, \mu$, and $r$ such that $0 < \alpha_1, \eta_1, \sigma < 1$,
  $0 < \delta_{min} \le \delta_{max}$, $\mu > 0$, and $r > 0$.
2 For $k = 0, 1, 2, \ldots$ do
  2.1 If $\|C_k\| + \|D_k W_k^T \nabla f_k\| \le \epsilon$, stop and return $x_k$ as an approximate solution for problem (1).
  2.2 Set $s_k^{\mathsf{n}} = s_k^{\mathsf{t}} = 0$.
      Compute $s_k^{\mathsf{n}}$ satisfying (3), (4), and (5).
      Compute $s_k^{\mathsf{t}} = W_k(s_k)_u$ where $(s_k)_u$ satisfies
          $$\sigma_k(a - u_k) \le (s_k)_u \le \sigma_k(b - u_k),$$
      with $\sigma_k \in [\sigma, 1)$, (10), and $\|S_k^{-1}(s_k)_u\| \le \delta_k$.
      Set $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$.
  2.3 Compute $\lambda_{k+1}$ and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.
  2.4 Set $pred(s_k;\rho_{k-1})$ to
          $$q_k(0) - q_k(s_k) - \Delta\lambda_k^T(J_k s_k + C_k) + \rho_{k-1}\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right).$$
      If $pred(s_k;\rho_{k-1}) \ge \frac{\rho_{k-1}}{2}\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right)$ then set $\rho_k = \rho_{k-1}$.
      Otherwise set
          $$\rho_k = \frac{2\left(q_k(s_k) - q_k(0) + \Delta\lambda_k^T(J_k s_k + C_k)\right)}{\|C_k\|^2 - \|J_k s_k + C_k\|^2} + \mu.$$
  2.5 If $\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} < \eta_1$, set
          $$\delta_{k+1} = \alpha_1 \max\left\{\frac{1}{r}\|s_k^{\mathsf{n}}\|, \|S_k^{-1}(s_k)_u\|\right\}$$
      and reject $s_k$.
      Otherwise accept $s_k$ and choose $\delta_{k+1}$ such that $\max\{\delta_{min}, \delta_k\} \le \delta_{k+1} \le$
      $\delta_{max}$.
```

2.6 If s_k was rejected set $x_{k+1} = x_k$. Otherwise set $x_{k+1} = x_k + s_k$.

If we use the coupled trust–region approach suggested in Section 4.2.2, then we need to restrict r to be in $(0, 1)$ and to change Steps 2.2 and 2.5. In Step 2.2 $(s_k)_u$ is now required to satisfy (11) and

$$\left\| \begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1}C_u(x_k)(s_k)_u \\ S_k^{-1}(s_k)_u \end{pmatrix} \right\| \leq \delta_k.$$

In Step 2.5, if $\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} < \eta_1$, we now set

$$\delta_{k+1} = \alpha_1 \left\| \begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1}C_u(x_k)(s_k)_u \\ S_k^{-1}(s_k)_u \end{pmatrix} \right\|.$$

It is not difficult to see that in the coupled trust–region approach we have $\|s_k\| \leq (1 + \nu_{10})\delta_k$ and $\delta_{k+1} \geq \frac{\alpha_1}{1+\nu_{10}}\|s_k\|$ in Step 2.5, where ν_{10} is a uniform bound for $\|S_k\|$, see Assumption **A.7** below. However this is not the case in the decoupled approach. Here $\|s_k\| = \|s_k^{\mathsf{n}} + W_k(s_k)_u\| \leq (r + \nu_7\nu_{10})\delta_k$ and similarly $\delta_{k+1} \geq \frac{\alpha_1}{r+\nu_7\nu_{10}}\|s_k\|$, where ν_7 is a uniform bound for $\|W_k\|$, see Assumption **A.4** below. We can combine these bounds to obtain

$$(14) \qquad \begin{array}{rcl} \|s_k\| & \leq & \max\{1 + \nu_{10}, r + \nu_7\nu_{10}\}\, \delta_k, \\[2ex] \delta_{k+1} & \geq & \min\{\frac{\alpha_1}{1+\nu_{10}}, \frac{\alpha_1}{r+\nu_7\nu_{10}}\}\, \|s_k\|. \end{array}$$

Of course the rules to update the trust radius in the previous algorithm can be much more involved but the above suffices to prove convergence results and to understand the trust–region mechanism.

As before we have the choices $S_k = \bar{D}_k$ and $S_k = I_{n-m}$.

In order to establish global convergence results we need some general assumptions. We list these assumptions below.

A.1 For all iterations k, x_k, $x_k + s_k \in \Omega$, where Ω is an open convex set of $\mathbb{R}^n$.

A.2 The functions $f(x)$, $c_i(x)$, $i = 1, \ldots, m$, are twice continuously differentiable on Ω. Here $c_i(x)$ represents the i–th component of $C(x)$.

A.3 The partial Jacobian $C_y(x)$ is nonsingular for all $x \in \Omega$.

A.4 The functions $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$, $C(x)$, $J(x)$, $\nabla^2 c_i(x)$, $i = 1, \ldots, m$, and $C_y(x)^{-1}$ are bounded in Ω.

A.5 The sequences $\{H_k\}$, $\{\lambda_k\}$ are bounded.

As consequence of **A.4** and **A.5**, there exist positive constants $\nu_0, \ldots, \nu_8$ independent of k such that $|f_k| \leq \nu_0$, $\|\nabla f_k\| \leq \nu_1$, $\|\nabla^2 f_k\| \leq \nu_2$, $\|C_k\| \leq \nu_3$, $\|J_k\| \leq \nu_4$ and $\|(\nabla^2 c_i)_k\| \leq \nu_5$, $i = 1, \ldots, m$, $\|H_k\| \leq \nu_6$, $\|C_y(x_k)^{-1}\| \leq \nu_7$, $\|W_k\| \leq \nu_7$, and $\|\lambda_k\| \leq \nu_8$.

If for some i, $a_i = -\infty$ or $b_i = +\infty$, we need to assume that $\{u_k\}$ is bounded. Thus we add the following assumption.

A.6 The sequence $\{u_k\}$ is bounded.

We need to restrict the choices of the scaling matrix S_k.

A.7 The sequences $\{\|S_k\|\}$ and $\{\|S_k^{-1}\bar{D}_k\|\}$ are bounded.

Under the Assumption **A.6**, the choices $S_k = \bar{D}_k$ and $S_k = I_{n-m}$ satisfy **A.7**.

It follows from the Assumptions **A.6** and **A.7** that $\|\bar{D}_k\| \leq \nu_9$, $\|S_k\| \leq \nu_{10}$, and $\|S_k^{-1}\bar{D}_k\| \leq \nu_{11}$, where ν_9, ν_{10} and ν_{11} are positive constants independent of k.

For the rest of this paper we suppose that general Assumptions **A.1**–**A.7** are always satisfied.

6. First–order convergence theory.

The first–order convergence result established in this section is obtained by using the convergence theory presented in [6] for the equality–constrained optimization problem. To do this we need some technical lemmas.

First we recall that the quasi–normal component s_k^{n} is assumed to satisfy the conditions (3), (4), and (5). From (5) and the fact that the tangential component lies in the null space of J_k, we obtain

$$\|C_k\|^2 - \|J_k s_k + C_k\|^2 \geq \kappa_2 \|C_k\| \min\{\kappa_3 \|C_k\|, r\delta_k\}. \tag{15}$$

In the following lemma we rewrite the fraction of Cauchy decrease conditions (10) and (11) in a more useful form for the analysis.

LEMMA 6.1. *If $(s_k)_u$ satisfies either (10) or (11) then*

$$\begin{aligned} q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k(s_k)_u) \quad \geq \quad & \kappa_4 \|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\| \cdot \\ & \min\left\{\kappa_5 \|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\|, \kappa_6 \delta_k\right\}, \end{aligned} \tag{16}$$

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

Proof. We first consider the decoupled trust region and thus assume (10). Let $\tilde{\delta}_k$ be the maximum $\|\bar{D}_k^{-1} \ \cdot\|$ norm of a step, say $(\tilde{s}_k)_u$, along $-\bar{D}_k \frac{\hat{g}_k}{\|\hat{g}_k\|}$ allowed inside the trust region given by (8). Here $\hat{g}_k = \bar{D}_k \bar{g}_k$. From Assumption **A.7**, we have

$$\delta_k = \|S_k^{-1}(\tilde{s}_k)_u\| \leq \nu_{11}\tilde{\delta}_k. \tag{17}$$

Define $\psi : \mathbb{R}^+ \longrightarrow \mathbb{R}$ as $\psi(t) = q_k(s_k^{\mathsf{n}} - tW_k\bar{D}_k\frac{\hat{g}_k}{\|\hat{g}_k\|}) - q_k(s_k^{\mathsf{n}})$. Then $\psi(t) = -\|\hat{g}_k\|t + \frac{r_k}{2}t^2$, where $r_k = \frac{\hat{g}_k^T \widetilde{H}_k \hat{g}_k}{\|\hat{g}_k\|^2}$ and $\widetilde{H}_k = \bar{D}_k W_k^T H_k W_k \bar{D}_k$. Now we need to minimize ψ in $[0, T_k]$ where T_k is given by

$$T_k = \min\left\{\tilde{\delta}_k,\ \sigma_k \min\left\{\frac{\|\hat{g}_k\|}{(\hat{g}_k)_i} : (\hat{g}_k)_i > 0\right\},\ \sigma_k \min\left\{-\frac{\|\hat{g}_k\|}{(\hat{g}_k)_i} : (\hat{g}_k)_i < 0\right\}\right\}.$$

Let t_k^* be the minimizer of ψ in $[0, T_k]$. If $t_k^* \in (0, T_k)$ then

$$\psi(t_k^*) = -\frac{1}{2}\frac{\|\hat{g}_k\|^2}{r_k} \leq -\frac{1}{2}\frac{\|\hat{g}_k\|^2}{\|\widetilde{H}_k\|}. \tag{18}$$

If $t_k^* = T_k$ then either $r_k > 0$ in which case $\frac{\|\hat{g}_k\|}{r_k} \geq T_k$ or $r_k \leq 0$ in which case $r_k T_k \leq \|\hat{g}_k\|$. In either event,

$$\psi(t_k^*) = \psi(T_k) = -T_k\|\hat{g}_k\| + \frac{r_k}{2}T_k^2 \leq -\frac{T_k}{2}\|\hat{g}_k\|. \tag{19}$$

We can combine (18) and (19) with

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k(s_k)_u) \geq \beta\left(q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k c_k^{\mathsf{d}})\right) = -\beta\psi(t_k^*)$$

to get

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k(s_k)_u) \geq \frac{1}{2}\beta\|\hat{g}_k\| \min\left\{ \frac{\|\hat{g}_k\|}{\|\widetilde{H}_k\|}, T_k \right\}.$$

The fact that $\sigma_k \geq \sigma$,

$$\min\left\{ \frac{\|\hat{g}_k\|}{(\hat{g}_k)_i} : (\hat{g}_k)_i > 0 \right\} \geq 1 \quad \text{and} \quad \min\left\{ -\frac{\|\hat{g}_k\|}{(\hat{g}_k)_i} : (\hat{g}_k)_i < 0 \right\} \geq 1,$$

implies that

$$\begin{aligned} q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k(s_k)_u) \quad &\geq \quad \tfrac{1}{2}\beta\|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\| \cdot \\ & \qquad \min\left\{ \frac{\|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\|}{\|\bar{D}_k^T W_k^T H_k W_k \bar{D}_k\|}, \min\left\{\tilde{\delta}_k, \sigma\right\} \right\}. \end{aligned} \tag{20}$$

Now we consider the coupled trust region and the condition (11). Let $\tilde{\delta}_k$ be the maximum $\|\bar{D}_k^{-1} \cdot\|$ norm of a step, say $(\tilde{s}_k)_u$, along $-\bar{D}_k \frac{\hat{g}_k}{\|\hat{g}_k\|}$ allowed inside the trust region given by (9). It is a simple matter to see that

$$\left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)(\tilde{s}_k)_u \\ S_k^{-1}(\tilde{s}_k)_u \end{pmatrix} \right\| \geq (1-r)\delta_k. \tag{21}$$

From the Assumptions **A.6** and **A.7** we have

$$\begin{aligned} \left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)(\tilde{s}_k)_u \\ S_k^{-1}(\tilde{s}_k)_u \end{pmatrix} \right\|^2 \quad &= \quad \| - C_y(x_k)^{-1}C_u(x_k)S_kS_k^{-1}(\tilde{s}_k)_u\|^2 + \|S_k^{-1}(\tilde{s}_k)_u\|^2 \\ &\leq \quad (\nu_7^2\nu_{10}^2 + 1)\|S_k^{-1}(\tilde{s}_k)_u\|^2 \\ &= \quad (\nu_7^2\nu_{10}^2 + 1)\ \|S_k^{-1}\bar{D}_k\bar{D}_k^{-1}(\tilde{s}_k)_u\|^2 \\ &\leq \quad (\nu_7^2\nu_{10}^2 + 1)\nu_{11}^2\ \|\bar{D}_k^{-1}(\tilde{s}_k)_u\|^2 \\ &= \quad (\nu_7^2\nu_{10}^2 + 1)\nu_{11}^2\ \tilde{\delta}_k^2. \end{aligned}$$

So from this and inequality (21), we get

$$\tilde{\delta}_k \geq \frac{1-r}{\nu_{11}\sqrt{\nu_7^2\nu_{10}^2 + 1}}\delta_k. \tag{22}$$

Using the arguments applied before we can show that (20) holds true.

To complete the proof, we use (17), (20), (22), the general assumptions and the fact that $\delta_k \leq \delta_{max}$ to establish (16) with $\kappa_4 = \frac{1}{2}\beta$, $\kappa_5 = \frac{1}{\nu_6\nu_7^2\nu_9^2}$ and $\kappa_6 = \min\{\frac{1}{\nu_{11}}, \frac{1-r}{\nu_{11}\sqrt{\nu_7^2\nu_{10}^2+1}}, \frac{\sigma}{\delta_{max}}\}$. □

We also need the following three inequalities.

LEMMA 6.2. *There exist positive constants κ_7, κ_8, and κ_9 independent of k such that*

$$q_k(0) - q_k(s_k^{\mathsf{n}}) - \Delta\lambda_k^T(J_ks_k + C_k) \geq -\kappa_7\|C_k\|, \tag{23}$$

$$|ared(s_k;\rho_k) - pred(s_k;\rho_k)| \le \kappa_8\|s_k\|^2 + \kappa_8\rho_k\|s_k\|^3 + \kappa_8\rho_k\|C_k\|\,\|s_k\|^2 \tag{24}$$

and

$$|ared(s_k;\rho_k) - pred(s_k;\rho_k)| \le \kappa_9\rho_k\|s_k\|^2. \tag{25}$$

Proof. For the proof of the first inequality see Lemma 7.3 in [6]. The proofs of (24) and (25) are given in [9, Lemma 6.3] and [6, Lemma 7.5], respectively. □

The following four lemmas bound the predicted decrease.

LEMMA 6.3. *If $(s_k)_u$ satisfies either (10) or (11), then the predicted decrease in the merit function satisfies*

$$\begin{aligned} pred(s_k;\rho) \;\ge\;& \kappa_4\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| \min\left\{\kappa_5\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\|, \kappa_6\delta_k\right\} \\ & -\kappa_7\|C_k\| + \rho(\|C_k\|^2 - \|J_k s_k + C_k\|^2), \end{aligned} \tag{26}$$

where $\rho \ge 1$.

Proof. The inequality (26) follows from a direct application of (23) and from the lower bound (16). □

LEMMA 6.4. *Assume that $(s_k)_u$ satisfies either (10) or (11) and that $\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| + \|C_k\| > \epsilon_{tol}$. If $\|C_k\| \le \alpha\delta_k$, where α is a positive constant satisfying*

$$\alpha \le \min\left\{\frac{\epsilon_{tol}}{3\delta_{max}}, \frac{\kappa_4\epsilon_{tol}}{3\kappa_7}\min\left\{\frac{2\kappa_5\epsilon_{tol}}{3\delta_{max}}, \kappa_6\right\}\right\}, \tag{27}$$

then we have

$$\begin{aligned} pred(s_k;\rho) \;\ge\;& \tfrac{\kappa_4}{2}\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| \min\{\kappa_5\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\|, \kappa_6\delta_k\} \\ & +\rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right), \end{aligned} \tag{28}$$

where $\rho \ge 1$.

Proof. From $\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| + \|C_k\| > \epsilon_{tol}$ and the first bound on α given by (27), we get

$$\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| > \frac{2}{3}\epsilon_{tol}. \tag{29}$$

If we use this, (26), and the second bound on α given by (27), we obtain

$$\begin{aligned} pred(s_k;\rho) \;\ge\;& \tfrac{\kappa_4}{2}\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| \min\{\kappa_5\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\|, \kappa_6\delta_k\} \\ & +\tfrac{\kappa_4\epsilon_{tol}}{3}\min\{\tfrac{2\kappa_5\epsilon_{tol}}{3}, \kappa_6\delta_k\} \\ & -\kappa_7\|C_k\| + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right) \\ \ge\;& \tfrac{\kappa_4}{2}\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\| \min\{\kappa_5\|\bar{D}_k W_k^T\nabla q_k(s_k^{\mathsf{n}})\|, \kappa_6\delta_k\} \\ & +\rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right). \end{aligned}$$

□

We can use Lemma 6.4 with $\rho = \rho_{k-1}$ and conclude that if $\|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\| + \|C_k\| > \epsilon_{tol}$ and $\|C_k\| \leq \alpha\delta_k$, then the penalty parameter at the current iteration does not need to be increased. This is equivalent to Lemma 7.7 in [6]. The next lemma states the same result as Lemma 7.8 in [6] but with a different choice of α.

LEMMA 6.5. *Let $(s_k)_u$ satisfy either (10) or (11) and suppose that $\|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\| + \|C_k\| > \epsilon_{tol}$. If $\|C_k\| \leq \alpha\delta_k$, where α satisfies (27), then there exists a positive constant $\kappa_{10} > 0$ such that*

$$pred(s_k; \rho_k) \geq \kappa_{10}\delta_k. \tag{30}$$

Proof. From (28) with $\rho = \rho_k$ and $\|\bar{D}_k \bar{g}_k\| \geq \frac{2}{3}\epsilon_{tol}$, cf. (29), we obtain

$$\begin{aligned} pred(s_k; \rho_k) &\geq \tfrac{\kappa_4 \epsilon_{tol}}{3} \min\{\tfrac{2\kappa_5\epsilon_{tol}}{3}, \kappa_6\delta_k\} \\ &\geq \tfrac{\kappa_4 \epsilon_{tol}}{3} \min\{\tfrac{2\kappa_5\epsilon_{tol}}{3\delta_{max}}, \kappa_6\}\delta_k. \end{aligned}$$

Hence (30) holds with

$$\kappa_{10} = \frac{\kappa_4\epsilon_{tol}}{3} \min\left\{\frac{2\kappa_5\epsilon_{tol}}{3\delta_{max}}, \kappa_6\right\}.$$

□

LEMMA 6.6. *The predicted decrease satisfies*

$$pred(s_k; \rho_k) \geq \frac{\rho_k}{2}\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right), \tag{31}$$

for all k.

Proof. The assertion follows directly from the scheme that updates ρ_k in Step 2.4 of Algorithms 5.1. □

Now we use the theory given by Dennis, El–Alem and Maciel [6] to state the following result.

THEOREM 6.1. *The sequences of iterates generated by the trust–region interior–point SQP Algorithms 5.1 satisfy*

$$\liminf_k \|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\| + \|C_k\| = 0.$$

Proof. Lemmas 7.9–7.13 and 8.2 as well as Theorems 8.1, 8.3 and 8.4 in [6] can be applied based on (4), (14), (15), (16), (23), (24), (25), (30), (31) and on the fact that if $\|\bar{D}_k W_k^T \nabla q_k(s_k^{\mathsf{n}})\| + \|C_k\| > \epsilon_{tol}$ and $\|C_k\| \leq \alpha\delta_k$, then the penalty parameter at the current iteration does not need to be increased. Thus this result is just a restating of Theorem 8.4 of [6]. □

Now let $D_k = D(u_k)$, where $D(u)$ is given by (2). The matrix D_k is different from $\bar{D}_k$ (given in Section 4.2.1) because we are choosing the diagonal elements based on the sign of the components of $W_k^T \nabla f_k$ and not on the sign of the elements of $W_k^T \nabla q_k(s_k^{\mathsf{n}})$ as we did when we defined $\bar{D}_k$. Now we can state the first–order convergence result that the trust–region interior–point SQP algorithms satisfy.

THEOREM 6.2. *The sequences of iterates generated by the trust–region interior–point SQP Algorithms 5.1 satisfy*

$$\liminf_k \|D_k W_k^T \nabla f_k\| + \|C_k\| = 0.$$

Proof. From Theorem 6.1 there exists a subsequence k_i such that

$$\lim_i \|\bar{D}_{k_i} W_{k_i}^T \nabla q_{k_i}(s_{k_i}^{\mathsf{n}})\| + \|C_{k_i}\| = 0.$$

We need to show that $\lim_i \|D_{k_i} W_{k_i}^T \nabla f_{k_i}\| = 0$. From (4) and the fact that $\lim_i \|C_{k_i}\| = 0$, we get $\lim_i \|s_{k_i}^{\mathsf{n}}\| = 0$. This completes the proof of the theorem. □

Although we have decided not to include it here, we can follow an argument similar to the one given for minimization with simple bounds [8], to extend Theorems 6.1 and 6.2 to scalings of the form D_k^p and $\bar{D}_k^p$ for $p \geq \frac{1}{2}$. We content ourselves with $p = 1$, which seems to us the most straightforward choice.

7. Trial steps and multiplier estimates.

When we described the trust–region interior–point SQP algorithms, we deferred the computation of the quasi–normal and tangential components and of the multiplier estimates. In the following sections we address these issues.

7.1. Computation of the quasi–normal component.

The quasi–normal component s_k^{n} is an approximate solution of the trust–region subproblem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|C_y(x_k)(s^{\mathsf{n}})_y + C_k\|^2 \\ \text{subject to} & \|(s^{\mathsf{n}})_y\| \leq r\delta_k, \end{array} \tag{32}$$

and it is required to satisfy conditions (3), (4) and (5). Property (4) is a consequence of (5). In fact, using $\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\| \leq \|C_k\|$ and the boundedness of $\{C_y(x_k)^{-1}\}$ we find that

$$\|s_k^{\mathsf{n}}\| \leq \|C_y(x_k)^{-1}\|(\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\| + \|C_k\|) \leq 2\nu_7 \, \|C_k\| \, .$$

Whether the property (5) holds depends on the way in which the quasi–normal component is computed. We will show below that (5) is satisfied for some of the most reasonable ways to compute s_k^{n}.

There are various ways to compute the quasi–normal step s_k^{n} for large scale problems. For example, one can use the conjugate–gradient method as suggested in [24] and [25], or one can use the Lanczos bidiagonalization as described in [11]. Both methods compute an approximate minimizer to the least squares functional in (32) from a subspace which contains its negative gradient $-C_y(x_k)^T C(x_k)$. Thus, the steps s_k^{n} generated by these methods satisfy $\|s_k^{\mathsf{n}}\| \leq r\delta_k$ and

$$\begin{array}{l} \frac{1}{2}\|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \\ \leq \min\{\frac{1}{2}\|C_y(x_k)s + C_k\|^2 : \; s \in span\{C_y(x_k)^T C_k\} \, , \, \|s\| \leq r\delta_k\} \, . \end{array} \tag{33}$$

We can appeal to a classical result due to Powell, see [23, Thm. 4], [20, Lemma 4.8], to show that

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \geq \frac{1}{2}\|C_y(x_k)^T C_k\| \min\left\{ \frac{\|C_y(x_k)^T C_k\|}{\|C_y(x_k)^T C_y(x_k)\|}, r\delta_k \right\}.$$

Now one can use the fact that $\{C_y(x_k)\}$ and $\{C_y(x_k)^{-T}\}$ are bounded and write

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \geq \kappa_2\|C_k\| \min\{\kappa_3\|C_k\|, r\delta_k\},$$

where κ_2 and κ_3 are positive and do not depend on k.

An alternative to the previous procedures is to compute the solution of $C_y(x_k)s = -C(x_k)$ and to scale this solution back into the trust region, i.e., to set

$$s_k^{\mathsf{n}} = \begin{pmatrix} -\xi_k C_y(x_k)^{-1}C_k \\ 0 \end{pmatrix}, \tag{34}$$

where

$$\xi_k = \begin{cases} 1 & \text{if } \|-C_y(x_k)^{-1}C_k\| \le r\delta_k, \\ \frac{r\delta_k}{\|-C_y(x_k)^{-1}C_k\|} & \text{otherwise.} \end{cases}$$

It follows from the boundedness on $\{C_y(x_k)^{-1}\}$ that s_k^{n} given by (34) satisfies the condition (4). In the following lemma we show that this choice of quasi–normal component also satisfies (5).

LEMMA 7.1. *The quasi–normal component (34) satisfies*

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \ge \kappa_2\|C_k\| \min\{\kappa_3\|C_k\|, r\delta_k\},$$

where κ_2 and κ_3 are positive constants independent of k.

Proof. A simple manipulation shows that

$$\begin{aligned} &\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \\ \ge\ & \|C_k\|^2 - \|-\xi_k C_y(x_k)C_y(x_k)^{-1}C_k + C_k\|^2 \\ \ge\ & \|C_k\|^2 - ((1-\xi_k)\|C_k\| + \xi_k\, \|-C_y(x_k)C_y(x_k)^{-1}C_k + C_k\|)^2 \\ =\ & \xi_k(2-\xi_k)\|C_k\|^2 \ \ge\ \xi_k\,\|C_k\|^2 \end{aligned}$$

We need to consider two cases. If $\xi_k = 1$, then

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \ge \|C_k\| \min\{\|C_k\|, r\delta_k\}.$$

Otherwise, $\xi_k = \frac{r\delta_k}{\|-C_y(x_k)^{-1}C_k\|}$. In this case we get

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{n}})_y + C_k\|^2 \ge \frac{1}{\nu_7}\|C_k\|\, r\delta_k \ge \frac{1}{\nu_7}\|C_k\| \min\{\|C_k\|, r\delta_k\}.$$

Thus the result holds with $\kappa_2 = \min\{1, \frac{1}{\nu_7}\}$ and $\kappa_3 = 1$. □

7.2. Computation of the tangential component. In this section we show how to derive conjugate–gradient algorithms to compute $(s_k)_u$. Let us consider first the decoupled trust–region approach given in Section 4.2.1. If we ignore the bound constraints for the moment, we can apply the conjugate–gradient algorithm proposed by Steihaug [24] and Toint [25] to solve the problem

$$\begin{aligned} \text{minimize} \quad & q_k(s_k^{\mathsf{n}}) + \left(W_k^T \nabla q_k(s_k^{\mathsf{n}})\right)^T s_u + \frac{1}{2} s_u^T W_k^T H_k W_k s_u \\ \text{subject to} \quad & \|S_k^{-1} s_u\| \le \delta_k, \end{aligned}$$

with our choices $S_k = \bar{D}_k$ and $S_k = I_{n-m}$. However we also need to incorporate the constraints

$$\sigma_k(a - u_k) \le s_u \le \sigma_k(b - u_k).$$

The algorithm is the following.

ALGORITHM 7.1 (COMPUTATION OF $s_k = s_k^{\mathsf{n}} + W_k(s_k)_u$ (DECOUPLED APPROACH)).

```
1 Set $s_u^0 = 0$, $r_0 = -W_k^T \nabla q_k(s_k^{\mathsf{n}})$, $q_0 = \bar{D}_k^2 r_0$, $d_0 = q_0$, and $\epsilon > 0$.
2 For $i = 0, 1, 2, \ldots$ do
    2.1 Compute $\gamma_i = \frac{r_i^T q_i}{d_i^T W_k^T H_k W_k d_i}$.
    2.2 Compute
          $\tau = \max\{\tau > 0 \ : \ \|S_k^{-1}(s_u^i + \tau d_i)\| \le \delta_k,$
                            $\sigma_k(a - u_k) \le s_u^i + \tau d_i \le \sigma_k(b - u_k)\}$.
    2.3 If $\gamma_i \le 0$, or if $\gamma_i > \tau$, then set $(s_k)_u = s_u^i + \tau d_i$, where $\tau$ is given as in
        2.2 and go to 3; otherwise set $s_u^{i+1} = s_u^i + \gamma_i d_i$.
    2.4 Update the residuals $r_{i+1} = r_i - \gamma_i W_k^T H_k W_k d_i$ and $q_{i+1} = \bar{D}_k^2 r_{i+1}$.
    2.5 Check truncation criteria: If $\sqrt{\frac{r_{i+1}^T q_{i+1}}{r_0^T q_0}} \le \epsilon$, set $(s_k)_u = s_u^{i+1}$ and go to 3.
    2.6 Compute $\alpha_i = \frac{r_{i+1}^T q_{i+1}}{r_i^T q_i}$ and set $d_{i+1} = q_{i+1} + \alpha_i d_i$.
3 Compute $s_k = s_k^{\mathsf{n}} + W_k(s_k)_u$ and stop.
```

Step 2 iterates entirely in the u–space. After the u–component of the step s_k has been computed, Step 3 finds its y–component. The decoupled approach allows an efficient use of an approximation $\widehat{H}_k$ to the reduced Hessian $W_k^T H_k W_k$. In this case only two linear systems are required, one with $C_y(x_k)^T$ in Step 1 and the other with $C_y(x_k)$ in Step 3. If it is the Hessian H_k that is being approximated, then the total number of linear systems is $2L_k + 2$, where L_k is the number of conjugate–gradient iterations.

One can transform this algorithm to work in the whole space rather then in the reduced space by considering the coupled trust–region approach given in Section 4.2.2. This requires the solution of two linear systems at each iteration no matter what type of Hessian approximation (reduced or full) is used. In either case the coupled approach requires a total of $2L_k + 2$ linear systems. This alternative is presented below.

ALGORITHM 7.2 (COMPUTATION OF $s_k = s_k^{\mathsf{n}} + W_k(s_k)_u$ (COUPLED APPROACH)).

```
1 Set $s^0 = s_k^{\mathsf{n}}$, $r_0 = -W_k^T \nabla q_k(s_k^{\mathsf{n}})$, $q_0 = \bar{D}_k^2 r_0$, $d_0 = W_k q_0$, and $\epsilon > 0$.
2 For $i = 0, 1, 2, \ldots$ do
    2.1 Compute $\gamma_i = \frac{r_i^T q_i}{d_i^T H_k d_i}$.
    2.2 Compute
          $\tau = \max\{\tau > 0 \ : \ \left\| \begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1} C_u(x_k) \tau (d_i)_u \\ S_k^{-1} \tau (d_i)_u \end{pmatrix} \right\| \le \delta_k,$

                    $\sigma_k(a - u_k) \le s_u^i + \tau (d_i)_u \le \sigma_k(b - u_k)\}$.
    2.3 If $\gamma_i \le 0$, or if $\gamma_i > \tau$, then stop and set $s_k = s^i + \tau d_i$, where $\tau$ is given
        as in 2.2; otherwise set $s^{i+1} = s^i + \gamma_i d_i$.
    2.4 Update the residuals $r_{i+1} = r_i - \gamma_i W_k^T H_k d_i$ and $q_{i+1} = \bar{D}_k^2 r_{i+1}$.
    2.5 Check truncation criteria: If $\sqrt{\frac{r_{i+1}^T q_{i+1}}{r_0^T q_0}} \le \epsilon$, set $s_k = s^{i+1}$ and stop.
```

2.6 Compute $\alpha_i = \frac{r_{i+1}^T q_{i+1}}{r_i^T q_i}$ and set $d_{i+1} = W_k(q_{i+1} + \alpha_i d_i)$.

Note that in Step 2 both the y– and the u–components of the step are being computed. The coupled approach is particularly suitable when an approximation to the full Hessian H_k is used. The coupled approach can also be used with an approximation $\widehat{H}_k$ to the reduced Hessian $W_k^T H_k W_k$. In this case we consider that H_k is given by (12), and we use the equalities (13) to compute the terms involving H_k in Algorithm 7.2.

Two final important remarks are in order.

REMARK 7.1. If $(W_k^T W_k)^{-1}$ was included as a preconditioner in Algorithm 7.2, then the conjugate–gradient iterates would monotonically increase in the norm $\|W_k \cdot \|$. Dropping this preconditioner means that the conjugate–gradient iterates do not necessarily increase in this norm. As result if the quasi–Newton step is inside the trust region, Algorithm 7.2 can terminate prematurely by stopping at the boundary of the trust region.

REMARK 7.2. Since the conjugate–gradient Algorithms 7.1, 7.2 start by minimizing the quadratic function $q_k(s_k^{\mathsf{n}} + W_k s_u)$ along the direction $-\bar{D}_k^2 \bar{g}_k$, it is quite clear that they produce reduced tangential components $(s_k)_u$ that satisfy (10) and (11), respectively, with $\beta = 1$.

7.3. Multiplier estimates. A convenient estimate for the Lagrange multipliers is the adjoint update

$$\lambda_k = -C_y(x_k)^{-T} \nabla_y f_k, \tag{35}$$

which we use after each successful step. However we also consider the following update:

$$\lambda_{k+1} = -C_y(x_k)^{-T} \nabla_y q_k(s_k^{\mathsf{n}}) = -C_y(x_k)^{-T} \left((H_k s_k^{\mathsf{n}})_y + \nabla_y f_k \right). \tag{36}$$

Here the use of (36) instead of

$$\lambda_{k+1} = -C_y(x_k + s_k)^{-T} \nabla_y f(x_k + s_k), \tag{37}$$

might be justified since we obtain (36) without any further cost from the first iteration of any of the conjugate–gradient algorithms described above. The updates (35), (36), and (37) satisfy the requirement given by **A.5** needed to prove first–order convergence.

8. Numerical example.

A typical application that has the structure described in this paper is the control of a heating process. In this section we introduce a simplified model for the heating of a probe in a kiln discussed in [2]. The temperature $y(x,t)$ inside the probe is governed by a nonlinear partial differential equation. The spatial domain is given by $(0,1)$. The boundary $x = 1$ is the inside of the probe and $x = 0$ is the boundary of the probe.

The goal is to control the heating process in such a way that the temperature inside the probe follows a certain desired temperature profile $y_d(t)$. The control $u(t)$ acts on the boundary $x = 0$. The problem can be formulated as follows.

$$\text{minimize} \ \frac{1}{2} \int_0^T [(y(1,t) - y_d(t))^2 + \gamma u^2(t)] dt$$

subject to

$$\begin{array}{rcl} \tau(y(x,t))\frac{\partial y}{\partial t}(x,t) & & \\ -\partial_x(\kappa(y(x,t))\partial_x y(x,t)) & = & q(x,t), \quad (x,t) \in (0,1)\times(0,T), \\ \kappa(y(0,t))\partial_x y(0,t) & = & g[y(0,t)-u(t)], \quad t\in(0,T), \\ \kappa(y(1,t))\partial_x y(1,t) & = & 0, \quad t \in (0,T), \\ y(x,0) & = & y_0(x), \quad x\in(0,1), \\ u_{low} \leq & u & \leq u_{upp}, \end{array}$$

where $y \in L^2(0,T;H^1(0,1))$, and $u \in L^2(0,T)$. The functions $\tau : \mathbb{R} \to \mathbb{R}$ and $\kappa : \mathbb{R} \to \mathbb{R}$ denote the specific heat capacity and the heat conduction, respectively, y_0 is the initial temperature distribution, q is the source term, g is a given scalar, and γ is a regularization parameter. Here $u_{low}, u_{upp} \in L^\infty(0,T)$ are given functions.

If the partial differential equation and the integral are discretized we obtain an optimization problem of the form (1). The discretization uses finite elements and was introduced in [2] (see also [12] and [16]). The spatial domain $(0,1)$ is divided into N_x subintervals of equidistant length, and the spatial discretization is done using piecewise linear finite elements. The time discretization is performed by partitioning the interval $[0,T]$ into N_t equidistant subintervals. Then the backward Euler method is used to approximate the state space in time, and piecewise constant functions are used to approximate the control space.

We implemented the TRIP SQP Algorithms 5.1 in FORTRAN 77. We use the formula (34) to compute the quasi–normal component, and Algorithms 7.1 and 7.2 to calculate the tangential component. The numerical test computations were done on a Sun Sparcstation 10 in double precision.

With this discretization scheme, $C_y(x)$ is a block bidiagonal matrix with tridiagonal blocks. Hence linear systems with $C_y(x)$ and $C_y(x)^T$ can be solved efficiently. In the implementation we use the LINPACK subroutine DGTSL to solve the tridiagonal systems. Inner products and norms used in the TRIP SQP algorithms are not Euclidean; instead we use discretizations of the $L^2(0,T)$ and $L^2(0,T;H^1(0,1))$ norms for the control and the state spaces respectively. This is important for the correct computation of the adjoint and the appropriate scaling of the problem.

In our numerical example we use the functions

$$\tau(y) = q_1 + q_2 y, \quad y \in \mathbb{R}, \quad \kappa(y) = r_1 + r_2 y, \quad y \in \mathbb{R},$$

with parameters $r_1 = q_1 = 4$, $r_2 = -1$, $q_2 = 1$. The desired and initial temperatures, and the right hand side are given by

$$\begin{array}{rcl} y_d(t) & = & 2 - e^{\eta t}, \\ y_0(x) & = & 2 + \cos \pi x, \\ q(x,t) & = & [\eta(q_1 + 2q_2) + \pi^2(r_1 + 2r_2)]e^{\eta t}\cos \pi x \\ & & -r_2\pi^2 e^{2\eta t} + (2r_2\pi^2 + \eta q_2)e^{2\eta t}\cos^2 \pi x, \end{array}$$

with $\eta = -1$. The final temperature is chosen to be $T = 0.5$ and the scalar g in the boundary condition is set to be one. The functions in this example are those used in [16, Ex. 4.1]. The size of the problem tested is $n = 2100$, $m = 2000$ corresponding to the values $N_t = 100$, $N_x = 20$.

The scheme used to update the trust radius is the following:

- If $ratio(s_k;\rho_k) < 10^{-4}$, reject s_k and set $\delta_{k+1} = 0.5\, norm(s_k)$;
- If $10^{-4} \leq ratio(s_k;\rho_k) < 0.1$, reject s_k and set $\delta_{k+1} = 0.5\, norm(s_k)$;
- If $0.1 \leq ratio(s_k;\rho_k) < 0.75$, accept s_k and set $\delta_{k+1} = \delta_k$;
- If $ratio(s_k;\rho_k) \geq 0.75$, accept s_k and set $\delta_{k+1} = 2\delta_k$;

where $ratio(s_k;\rho_k) = \frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)}$ and $norm(s_k)$ is given by

$$\max\left\{\frac{1}{r}\|s_k^{\mathsf{n}}\|, \|S_k^{-1}(s_k)_u\|\right\}$$

in the decoupled approach and by

$$\left\|\begin{pmatrix} (s_k^{\mathsf{n}})_y - C_y(x_k)^{-1}C_u(x_k)(s_k)_u \\ S_k^{-1}(s_k)_u \end{pmatrix}\right\|$$

in the coupled approach.

We have used $r = 0.5$ in the decoupled approach and $r = 1$ in the coupled approach; $\sigma_k = \sigma = 0.99995$ for all k; $\delta_0 = 1$ as initial trust radius; $\rho_{-1} = 1$ and $\mu = 10^{-2}$ in the penalty scheme. The tolerances used were $\epsilon = 10^{-8}$ for the main iteration and $\epsilon = 10^{-4}$ for the conjugate–gradient iteration. The upper and lower bounds were $b_i = 10^{-2}$, $a_i = -1000$, $i = 1, \ldots, n-m$. The starting vector was $(y_0, u_0) = (0, 0)$.

For both, the decoupled and the coupled approaches, we used approximations to reduced and to full Hessians. We approximate these matrices with the limited memory BFGS representations given in [3] with a memory size of 5 pairs of vectors. The initial approximation chosen was γI_{n-m} for the reduced Hessian and γI_n for the full Hessian, where γ is the regularization parameter in the objective function.

In our implementation we use the following form of the diagonal matrices D_k and $\bar{D}_k$

$$(D_k)_{ii} = \begin{cases} \min\{1, (b - u_k)_i\} & \text{if} \ \left(W_k^T \nabla f_k\right)_i < 0, \\ \\ \min\{1, (u_k - a)_i\} & \text{if} \ \left(W_k^T \nabla f_k\right)_i \geq 0, \end{cases} \tag{38}$$

$$(\bar{D}_k)_{ii} = \begin{cases} \min\{1, (b - u_k)_i\} & \text{if} \ \bar{g}_i < 0, \\ \\ \min\{1, (u_k - a)_i\} & \text{if} \ \bar{g}_i \geq 0, \end{cases} \tag{39}$$

for $i = 1, \ldots, n-m$. This form of D_k and $\bar{D}_k$ gives a better transition between the infinite and finite bound and is less sensitive to the introduction of meaningless bounds. Proposition 3.1 and the convergence result given in Theorem 6.2 hold true with D_k and $\bar{D}_k$ given by (38) and (39) respectively.

The results are shown in Tables 1, 2 corresponding to the values $\gamma = 10^{-2}$ and $\gamma = 10^{-3}$, respectively. There were no rejected steps. The different alternatives tested performed quite similarly. The decoupled approach with reduced Hessian approximation seems to be the best for this example. Note that in this case the computation of each trial step costs only three linear systems with $C_y(x_k)$ and $C_y(x_k)^T$, one to compute the quasi–normal component and two for the computation of the tangential component.

Table 1
Numerical results for $\gamma = 10^{-2}$.

	Decoupled		Coupled	
	Reduced $\widehat{H}_k$	Full H_k	Reduced $\widehat{H}_k$	Full H_k
number of iterations k^*	13	16	19	19
$\lVert C_{k^*}\rVert$	$.8638E-10$	$.4426E-10$	$.5768E-12$	$.1216E-10$
$\lVert D_{k^*}W_{k^*}^T\nabla f_{k^*}\rVert$	$.1273E-08$	$.5574E-08$	$.3221E-09$	$.3145E-08$
$\lVert s_{k^*-1}\rVert$	$.3405E-04$	$.3697E-04$	$.4641E-05$	$.1629E-04$
δ_{k^*-1}	$.8192E+04$	$.6554E+05$	$.5243E+06$	$.5243E+06$
ρ_{k^*-1}	$.1000E+01$	$.1000E+01$	$.1000E+01$	$.1000E+01$

Table 2
Numerical results for $\gamma = 10^{-3}$.

	Decoupled		Coupled	
	Reduced $\widehat{H}_k$	Full H_k	Reduced $\widehat{H}_k$	Full H_k
number of iterations k^*	15	19	16	21
$\lVert C_{k^*}\rVert$	$.2518E-09$	$.4550E-10$	$.1739E-09$	$.3699E-10$
$\lVert D_{k^*}W_{k^*}^T\nabla f_{k^*}\rVert$	$.9276E-08$	$.2780E-09$	$.1967E-09$	$.4887E-10$
$\lVert s_{k^*-1}\rVert$	$.1024E-03$	$.3620E-04$	$.5402E-04$	$.2338E-04$
δ_{k^*-1}	$.3277E+05$	$.5243E+06$	$.6554E+05$	$.2097E+07$
ρ_{k^*-1}	$.1000E+01$	$.1000E+01$	$.1000E+01$	$.1000E+01$

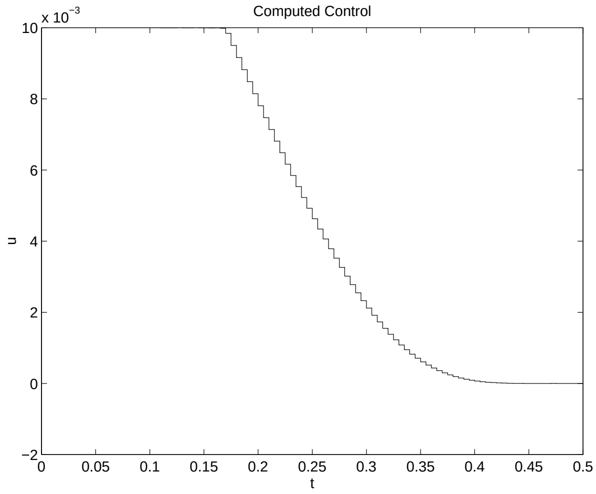

FIG. 1. *Coleman–Li scaling.*

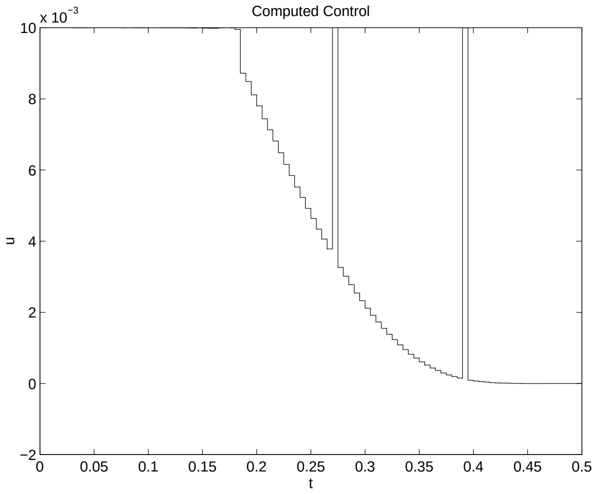

FIG. 2. *Dikin–Karmarkar scaling.*

We made an experiment to compare the use of the Coleman–Li scaling with the Dikin–Karmarkar scaling. The latter scaling is given by

$$(K_k)_{ii} = (\bar{K}_k)_{ii} = \min\{1, (u_k - a)_i, (b - u_k)_i\} \tag{40}$$

and has no dual information built in. We ran the TRIP SQP algorithm with the decoupled and reduced Hessian approximation and (38), (39) replaced by (40). The algorithm took only 11 iterations to reduce $\|K_k W_k^T \nabla f_k\| + \|C_k\|$ to 10^{-8}. However as we can see from the plots of the controls in Figures 1 and 2 the algorithm did not find the correct solution when it used the Dikin–Karmarkar scaling (40). Some of the variables are at the wrong bound corresponding to negative multipliers.

9. Conclusions. In this paper we have introduced and analyzed some trust–region interior–point SQP algorithms for an important class of nonlinear programming problems that appear in many engineering applications. These algorithms use the structure of the problem, and they combine trust–region techniques for equality–constrained optimization with a affine–scaling interior–point approach for simple bounds. We have proved a first–order convergence result for these algorithms that includes as special cases both the results established for equality constraints [6] and those for simple bounds [4], [8].

We implemented the trust–region interior–point SQP algorithms and tested them on a specific optimal control problem governed by a nonlinear heat equation. The numerical results were quite satisfactory.

We are investigating extensions of these algorithms to handle bounds on the state variables y. We are also developing an inexact analysis to deal with trial step computations that allow for inexact linear system solvers and inexact directional derivatives [13]. The formulation and analysis of these methods in an infinite dimensional framework is also part of our current studies.

Acknowledgements. This research collaboration on the tailoring of current optimization algorithm design to exploit the structure of problems in computational design and control was begun as the result of conversations held during the April 1994 Workshop on Optimal Design and Control held in Blacksburg at Virginia Polytechnic

Institute and State University. This workshop was organized by the Interdisciplinary Center for Applied Mathematics at Virginia Tech and sponsored by the AFOSR.

REFERENCES

[1] J. F. Bonnans and C. Pola, *A trust region interior point algorithm for linearly constrained optimization*, Tech. Rep. 1948, INRIA, 1993.

[2] J. Burger and M. Pogu, *Functional and numerical solution of a control problem originating from heat transfer*, J. Optim. Theory Appl., 68 (1991), pp. 49–73.

[3] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi–newton matrices and their use in limited memory methods*, Math. Programming, 63 (1994), pp. 129–156.

[4] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, Tech. Rep. TR93–1342, Department of Computer Science, Cornell University, 1993.

[5] T. F. Coleman and J. Liu, *An interior Newton method for quadratic programming*, Tech. Rep. TR93–1388, Department of Computer Science, Cornell University, 1993.

[6] J. E. Dennis, M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust–region–based algorithms for equality constrained optimization*, Tech. Rep. TR92–28, Department of Computational and Applied Mathematics, Rice University, 1992.

[7] J. E. Dennis and L. N. Vicente, *On the convergence theory of general trust–region–based algorithms for equality–constrained optimization*, Tech. Rep. TR94–36, Department of Computational and Applied Mathematics, Rice University, 1994.

[8] ———, *Trust–region interior–point algorithms for minimization problems with simple bounds*, Tech. Rep. TR94–42, Department of Computational and Applied Mathematics, Rice University, 1994.

[9] M. El-Alem, *A global convergence theory for the Celis–Dennis–Tapia trust–region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[10] ———, *A robust trust–region algorithm with a non–monotonic penalty parameter scheme for constrained optimization*, Tech. Rep. TR92–30, Department of Computational and Applied Mathematics, Rice University, 1992. To appear in SIAM J. Optim.

[11] G. H. Golub and U. von Matt, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.

[12] M. Heinkenschloss, *Projected sequential quadratic programming methods*, Tech. Rep. ICAM 94–05–02, Department of Mathematics, Virginia Polytechnic Institute and State University, 1994. To appear in SIAM J. Optimization.

[13] M. Heinkenschloss and L. N. Vicente, *An inexact analysis for trust–region interior–point SQP algorithms.* In preparation.

[14] K. Kunisch and E. Sachs, *Reduced SQP methods for parameter identification problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1793–1820.

[15] F.-S. Kupfer, *Reduced Successive Quadratic Programming in Hilbert Space with Applications to Optimal Control*, PhD thesis, Universität Trier, Fb–IV, Mathematik, D–54286 Trier, Germany, 1992.

[16] F.-S. Kupfer and E. W. Sachs, *Numerical solution of a nonlinear parabolic control problem by a reduced SQP method*, Computational Optimization and Applications, 1 (1992), pp. 113–135.

[17] M. Lalee, J. Nocedal, and T. Plantenga, *On the implementation of an algorithm for large–scale equality constrained optimization.* 1994.

[18] Y. Li, *On global convergence of a trust region and affine scaling method for nonlinearly constrained minimization*, Tech. Rep. CTC94TR197, Advanced Computing Research Institute, Cornell University, 1994.

[19] ———, *A trust region and affine scaling method for nonlinearly constrained minimization*, Tech. Rep. CTC94TR198, Advanced Computing Research Institute, Cornell University, 1994.

[20] J. J. Moré, *Recent developments in algorithms and software for trust regions methods*, in Mathematical programming. The state of art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer Verlag, New York, 1983, pp. 258–287.

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

[22] T. Plantenga, *Large-scale nonlinear constrained optimization using trust regions*, PhD thesis,

Northwestern University, Evanston, Illinois, 1994.

[23] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970.

[24] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[25] 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.