A TRUST REGION METHOD FOR PARABOLIC BOUNDARY CONTROL PROBLEMS*

C.T. KELLEY† AND E.W. SACHS‡

Abstract. In this paper we develop a trust region algorithm for constrained parabolic boundary control problems. The method is a projected form of the Steihaug trust-region-CG method with a smoothing step added at each iteration to improve performance in the global phase and provide mesh-independent sup-norm convergence in the terminal phase.

Key words. Trust region methods, inexact Newton methods, optimal control

AMS subject classifications. 49K20, 65F10, 49M15, 65J15, 65K10

1. Introduction. In this paper we show how methods that combine CG iteration and trust region globalization for optimization problems subject to simple bounds can be applied in a infinite dimensional setting to parabolic optimal control problems. This paper addresses the global convergence questions left open by our previous work [11], [13], [14], on fast multilevel algorithms for local convergence by showing how trust region-CG algorithms can solve the coarse mesh problems needed to initialize the multilevel method in an efficient and mesh-independent way. The algorithm uses the postsmoothing step from [14] and [11] to improve the performance of the iteration.

For unconstrained problems, our approach differs from [18] only in that after a step and trust region radius have been accepted, a smoothing iteration like those in [11] and [14] is attempted. Unlike our previous work, however, an Armijo [1] line search is added to the smoothing step to ensure decrease in the objective function. This new form of the smoothing step is a scaled steepest descent algorithm. The local theory from [11] and [14] implies that full smoothing steps are taken near the solution.

The effect of this in the infinite dimensional case is to allow one to make sup-norm error estimates in the terminal phase of the iteration [11], [14]. In the constrained case, we differ from the algorithm in [8] in more ways. We use L^2 trust region and solve unconstrained trust region problems, using the reduced Hessian at the current point to build the quadratic model. The reason for this is to make the trust region problem as easy to solve as possible and to eliminate the need to explicitly compute a generalized Cauchy point. We update the active set after the trust region step has been computed with a scaled projected gradient step (similar to [8]). The scaling serves the purpose of becoming an inexact implementation of the algorithm in [11] and [14] when full steps are taken. We then obtain fast local convergence in the L^∞ norm. Our local convergence theory does not depend on identification of the active set in finitely many iterations but rather applies the measure-theoretic ideas in [14]. Hence, our trust region algorithm becomes an inexact projected Newton method in the terminal phase of the iteration with local convergence properties covered by

*Version of April 20, 1997.

† North Carolina State University, Department of Mathematics and Center for Research in Scientific Computation, Box 8205, Raleigh, N. C. 27695-8205 (Tim_Kelley@ncsu.edu). The research of this author was supported by National Science Foundation grant #DMS-9321938 and North Atlantic Treaty Organization grant #CRG 920067. Computing was partially supported by an allocation of time from the North Carolina Supercomputing Center.

‡Universität Trier, FB IV – Mathematik and Graduiertenkolleg Mathematische Optimierung, 54296 Trier, Germany (sachs@uni-trier.de). The research of this author was supported by North Atlantic Treaty Organization grant #CRG 920067.

the theory developed in [3], [11], and [14].

We consider the problem of minimizing

$$f(u) = \frac{1}{2}\int_0^1 (y(u;T,x) - z(x))^2\,dx + \frac{\alpha}{2}\int_0^T u^2(t)\,dt, \tag{1.1}$$

where $\alpha > 0$ is given and $y(t,x) = y(u;t,x)$ is the solution to the nonlinear parabolic problem

$$\begin{array}{l} y_t(t,x) = y_{xx}(t,x), \quad 0 < x < 1, \quad 0 < t < T, \\ y(0,x) = y_0(t), \quad 0 < x < 1, \\ y_x(t,0) = 0, \quad y_x(t,1) = g(y(t,1)) + u(t), \quad 0 < t < T. \end{array} \tag{1.2}$$

In (1.1)–(1.2) u is constrained to be in the set

$$\mathcal{U} = \{u \in L^\infty([0,T]) \mid u_{min}(t) \le u(t) \le u_{max}(t), \text{for } \textit{a. e. } t \in [0,T]\} \tag{1.3}$$

for fixed $u_{min}, u_{max} \in L^\infty([0,T])$ and the nonlinear function g is assumed to satisfy

$$g \in C^2(R), \quad g', g'' \in L^\infty(R). \tag{1.4}$$

See [21] for examples of applications.

The gradient of f in $L^2([0,T])$ is

$$(\nabla f(u))(t) = \alpha u(t) + d(t,1), \tag{1.5}$$

where $d(t,x)$ is the solution of the adjoint problem

$$\begin{array}{l} -d_t(t,x) = d_{xx}(t,x), \quad 0 < x < 1, \quad 0 < t < T \\ d(T,x) = y(T,x) - z(x), \quad 0 < x < 1, \\ d_x(t,0) = 0, \quad d_x(t,1) = g'(y(t,1))d(t,1), \quad 0 < t < T. \end{array} \tag{1.6}$$

The map $u \to d(t,1)$ is completely continuous as a map on $C[0,T]$ and as a map from $L^q[0,T]$, $1 \le q \le \infty$, to $C[0,T]$, [17]. We will make use of these compactness properties in this paper.

We base our methods on the work in [7], [8], and [18] (where $u_{max} = +\infty$ and $u_{min} = -\infty$ for unconstrained problems). These methods solve the trust region problem and by searching along the piecewise linear path having the CG iterates as nodes, terminating either on the trust region boundary, with an inexact Newton step, or a direction of negative curvature. The compactness of the map $u \to d(t,1)$ insures that the performance of the CG iteration is independent of the discretization. This is consistent with results on linear equations and CG (see [9] and [20]). Another benefit of the compactness is that the reduced Hessian of f is a compact perturbation of a constant multiple of the identity and hence no preconditioning is needed for fast convergence.

We close this section with some notation and definitions.

Let $(\cdot,\cdot)$ denote the inner product in L^2 or the Euclidean inner product in any finite dimensional space. We denote the L^2-norm by $\|\cdot\|_2$ and the L^∞-norm by $\|\cdot\|_\infty$.

We let $\mathcal{P}$ be the L^2 projection onto $\mathcal{U}$ defined for any measurable u on $[0,T]$ and almost every $t \in [0,T]$ as

$$(\mathcal{P}u)(t) = \begin{cases} u_{min}(t), & \text{if } u(t) \le u_{min}(t), \\ u(t), & \text{if } u_{min}(t) < u(t) < u_{max}(t), \\ u_{max}(t), & \text{if } u(t) \ge u_{max}(t). \end{cases} \tag{1.7}$$

We define

$$F(u)(t) = u(t) - \mathcal{P}(u(t) - \nabla f(u)(t)). \tag{1.8}$$

The nonsmooth nonlinear equation $F(u) = 0$ is a necessary condition for stationarity [2].

The map $\mathcal{K}_0$, given by

$$\mathcal{K}_0(u) = -\alpha^{-1} d(t,1) = u - \alpha^{-1} \nabla f(u), \tag{1.9}$$

is a completely continuous map from $L^q[0,T] \to C[0,T]$ for any $1 \leq q \leq \infty$ [11], [14].

For $u \in \mathcal{U}$, we define the active set for u as

$$\begin{aligned} \mathcal{A}(u) &= \{t \,|\, u(t) = u_{max},\ \mathcal{K}_0(u)(t) \geq u_{max}(t)\} \\ &\cup \{t \,|\, u(t) = u_{min},\ \mathcal{K}_0(u)(t) \leq u_{min}(t)\} \end{aligned} \tag{1.10}$$

and the inactive $\mathcal{I}(u)$ set as $[0,T] \setminus \mathcal{A}(u)$. It is clear that for any $\lambda > 0$

$$u(t) = \mathcal{P}(u(t) - \lambda \nabla f(u)(t)) \text{ for all } t \in \mathcal{A}(u). \tag{1.11}$$

2. Algorithms. The algorithms are all based on the trust region-CG method in [18] and the general convergence analysis in [19]. The trust region problem is solved approximately by using a piecewise linear path whose nodes are the CG iterates. This approximate solution of the trust region problem is used in a standard way [10], [18], [16], to test for sufficient decrease and adjust the trust region radius. We incorporate the CG-TR method into the inexact projected Newton approach of [11] to give a superlinearly convergent algorithm.

2.1. Inexact Projected Newton Algorithm. To specify the algorithm we must define projections that correspond to the active and inactive set. For any measurable $S \subset [0,T]$ we define the multiplication operator $\mathcal{P}_S$ by

$$\mathcal{P}_S u(t) = \chi_S(t) u(t), \tag{2.1}$$

where χ_S is the characteristic function of S. In particular, if $u \in \mathcal{U}$ and $\mathcal{A}$ and $\mathcal{I}$ are approximations to $\mathcal{A}(u)$ and $\mathcal{I}(u)$, we will use

$$\mathcal{P}_{\mathcal{A}} w(t) = \chi_{\mathcal{A}}(t) w(t) \text{ and } \mathcal{P}_{\mathcal{I}} w(t) = \chi_{\mathcal{I}}(t) w(t). \tag{2.2}$$

We follow [14] and [11] and approximate the active set by

$$\begin{aligned} \mathcal{A} = \mathcal{A}_\epsilon(u) &= \{t \,|\, u(t) = u_{max},\ \mathcal{K}_0(u)(t) \geq u_{max}(t) + \epsilon\} \\ &\cup \{t \,|\, u(t) = u_{min},\ \mathcal{K}_0(u)(t) \leq u_{min}(t) - \epsilon\} \end{aligned} \tag{2.3}$$

and let

$$\mathcal{I} = \mathcal{I}_\epsilon(u) = [0,T] \setminus \mathcal{A}_\epsilon(u).$$

The parameter $\epsilon > 0$ may be adjusted as the iteration progresses to give local superlinear convergence [11], [14].

Note that for all $\epsilon > 0$ we have

$$\mathcal{A}_\epsilon(u) \subset \mathcal{A}(u)$$

and hence

$$u(t) = \mathcal{P}(u(t) - \lambda \nabla f(u)(t)) \text{ for all } \lambda > 0 \text{ and } t \in \mathcal{A}_\epsilon(u). \tag{2.4}$$

In the constrained case, the necessary conditions for optimality can be expressed as a nondifferentiable compact fixed point problem

$$u = \mathcal{K}(u) \tag{2.5}$$

where

$$\mathcal{K}(u) = \mathcal{P}(\mathcal{K}_0(u)).$$

Recall from § 1 that the map $\mathcal{K}_0$ (and hence $\mathcal{K}$) is a compact map from $L^q[0,T]$ to $C[0,T]$ for any $1 \le q \le \infty$. In that sense $\mathcal{K}$ is a smoother. We will use that property to (1) improve the global convergence properties of our proposed algorithm and (2) provide a uniform norm local convergence theory as in [14] and [11].

We define the reduced Hessian $\mathcal{R}(u_c)$ at u_c by

$$\mathcal{R}(u_c) = \mathcal{P}_\mathcal{A} + \mathcal{P}_\mathcal{I} \nabla^2 f(u_c) \mathcal{P}_\mathcal{I} \tag{2.6}$$

with $\mathcal{I} = \mathcal{I}_\epsilon(u)$.

The inexact projected Newton algorithms from [11] and [14] have several stages. We describe the one from [11] here in terms of the transition from a current approximation u_c to a new one u_+. The understanding here is that the parameter ϵ in the approximation to the active set $\mathcal{A}_\epsilon(u_c)$ and the forcing term η in the inexact Newton process change as the iteration progresses.

ALGORITHM 2.1. `pnstep`$(u_c, u_+, f, \epsilon_c, \eta_c)$

1. **Identification:** *Given* u_c *and* ϵ_c *set* $\mathcal{I} = \mathcal{I}_\epsilon(u_c)$.
2. **Error Reduction:** *Find* $s \in Im\mathcal{P}_\mathcal{I}$ *which satisfies*

$$\|\mathcal{R}(u_c)s + \mathcal{P}_\mathcal{I} \nabla f(u_c)\|_2 < \eta_c \|\mathcal{P}_\mathcal{I} \nabla f(u_c)\|_2. \tag{2.7}$$

Set

$$u_{1/2} = \mathcal{P}(u_c + s)$$

to reduce the error in L^2.

3. **Postsmoothing:** *Set* $u_+ = \mathcal{K}(u_{1/2})$ *to recover convergence* $C[0,T]$.

In the context of this paper, in which global convergence is the issue, Algorithm `pnstep` presents two problems. Firstly, the smoothing step is a scaled gradient projection step and may lead to dramatic increases in the objective function when u_c is far from the solution. We remedy this by adding an Armijo line search to this phase of the algorithm but do not demand $f(u_+) < f(u_{1/2})$, which may never be possible, but only that $f(u_+) < f(u_c)$ by a certain small amount. The results in [11] and [14] insure that if u_c is sufficiently near the solution, then the full smoothing step will be accepted and hence the fast local convergence (the precise speed of convergence depends on the choice of forcing term η) will not be effected by the line search. Secondly, there is no guarantee that the reduced Hessian will be positive definite. We address this problem with an inexact trust region algorithm that will exploit any negative curvature direction that it finds.

As is standard we use the measure of nonstationarity

$$\sigma(u) = \|u - \mathcal{P}(u - \nabla f(u))\|_2 = \|F(u)\|_2. \tag{2.8}$$

For example, a locally convergent algorithm using Algorithm `pnstep` is

ALGORITHM 2.2. `pnlocal`$(u, f, \epsilon_0, \eta_0, \sigma_0)$

1. *If* $\sigma(u) \leq \sigma_0$*, terminate the iteration.*
2. $\epsilon = \min(\epsilon_0, \sigma(u)^{1/2})$, $\eta = \min(\eta_0, \sigma(u)^{1/2})$
3. *Take an inexact step* `pnstep`$(u, u_{1/2}, f, \epsilon, \eta)$.
4. $u = u_+$. *Go to step 1.*

The values of η and ϵ in step 2 will ensure superlinear convergence with q-order $5/4$, [11], [14].

Under standard assumptions, [14], [11], Algorithm `pnlocal` will produce iterates that converge locally q-superlinearly (in the L^∞ norm) to a minimizer. Q-linear convergence can be obtained if the formula for η in step 2 is replaced by $\eta = \eta_0$ and η_0 is sufficiently small. The purpose of this paper is to develop a trust region globalization for this algorithm that preserves the L^∞ norm local convergence in the terminal phase while converging globally in L^2.

2.2. Solution of the Trust Region Problem. We use a standard solver form [18] for our unconstrained trust region subproblems. The inputs to Algorithm `trcg`, which approximately solves the trust region problem, are the current point u, the objective f, a preconditioner M, the forcing term η, the current trust region radius Δ, and a limit on the number of iterations $kmax$. The output is the approximate solution of the trust region problem d. We formulate the algorithm using the preconditioned CG framework from [12].

We will assume for the present that gradients are computed exactly and that Hessian-vector product $\nabla^2 f(u)w$ is approximated by the difference quotient

$$D_h^2 f(u:w) = \begin{cases} 0 & w = 0 \\ \dfrac{\nabla f(u + h\|u\|_2 w/\|w\|_2) - \nabla f(u)}{h\|u\|_2/\|w\|_2} & w, u \neq 0 \\ \dfrac{\nabla f(hw/\|w\|_2) - \nabla f(0)}{h/\|w\|_2} & u = 0, w \neq 0. \end{cases} \tag{2.9}$$

We present the algorithm of [18] for approximate solution of the trust region problem

$$\min_{\|d\|_2 \leq \Delta} (g, d) + .5(d, Bd).$$

In practice the action of B on a vector may be inaccurate and even nonlinear, as would be the case with $Bw = D_h^2 f(u:w)$. However, the effects of such inaccuracy are simply that the method is equivalent to one in which the Hessian is a difference approximation based on the basis for the Krylov subspace.

ALGORITHM 2.3. `trcg`$(d, u, g, B, M, \eta, \Delta, kmax)$

1. $r = -g$, $\rho_0 = \|r\|_2^2$, $k = 1$, $d = 0$
2. *Do While* $\sqrt{\rho_{k-1}} > \eta\|g\|_2$ *and* $k < kmax$
   (a) $z = Mr$
   (b) $\tau_{k-1} = (z, r)$
   (c) *if* $k = 1$ *then* $\beta = 0$ *and* $p = z$
   else
   $\beta = \tau_{k-1}/\tau_{k-2}$, $p = z + \beta p$

(d) $w = Bp$
If $(p, w) \leq 0$ *then*
Find τ *such that* $\|d + \tau p\|_2 = \Delta$
$d = d + \tau p$*; return*
(e) $\alpha = \tau_{k-1}/(p, w)$
(f) $r = r - \alpha w$
(g) $\rho_k = (r, r)$
(h) $\hat{d} = d + \alpha p$
(i) *If* $\|\hat{d}\|_2 > \Delta$ *then*
Find τ *such that* $\|d + \tau p\|_2 = \Delta$
$d = d + \tau p$*; return*
(j) $d = \hat{d}$*;* $k = k + 1$

Trust region algorithms for bound constrained problems have been analyzed in considerable generality in [7]. A concrete algorithm, proposed in [8], follows a piecewise linear path in a search for a generalized Cauchy point, freezes the active set at that point, and then solves the trust region problem approximately on the current active set. This process is important for the theory in [7] not only because it guarantees Cauchy decrease but also for the proof of superlinear convergence after the active set has been identified.

In the problems considered here, where there is a continuum of constraints, it is not clear how to use the method of [8] because the active set, being uncountable, will never be fully identified, and the construction of a path on which to search for a Cauchy point would lead to infinitely many knots to test. Instead we solve an unconstrained trust region problem for a reduced quadratic model and project the solution of that problem onto the active set.

Our approach to minimization of the reduced quadratic model also differs from that in [8]. In that paper, and in the convergence analysis in [7], the fact that all norms in finite dimension are equivalent was used to justify l^∞ trust region bounds. We use the standard L^2 trust region and therefore do not include the constraints explicitly in the trust region. We then use a smoothing step to deal with the nonequivalence of norms and recover fast uniform convergence in the terminal phase of the iteration.

Given $u_c \in \mathcal{U}$ and ϵ we consider the reduced quadratic model

$$m_c(u) = f(u_c) + (\mathcal{P}_{\mathcal{I}} \nabla f(u_c), u - u_c) + (u - u_c, \mathcal{P}_{\mathcal{I}} \mathcal{R}(u_c) \mathcal{P}_{\mathcal{I}}(u - u_c))/2. \tag{2.10}$$

In (2.10), the reduced Hessian $\mathcal{R}$ is given by (2.6). Note that the action of $\mathcal{R}(u)$ on a function can easily be computed by differences. This is a somewhat nonstandard model in that $\mathcal{P}_{\mathcal{I}} \nabla f(u)$ is used in the first order part of (2.10) rather than $\nabla f(u)$ and

$$\mathcal{P}_{\mathcal{I}} \mathcal{R}(u_c) = \mathcal{P}_{\mathcal{I}} \mathcal{R}(u_c) \mathcal{P}_{\mathcal{I}}$$

instead of $\mathcal{R}(u_c)$. The reasons for this are that this model performed better in our numerical experiments and also makes a smoother transition to a fast local algorithm that can be analyzed with the ideas from [11] and [14]. The nonstandard quadratic term presents no problems, however the linear term must be accounted for in the analysis, but, as we shall show next, this is easy to do because our algorithm is a dog-leg.

The analysis of the global convergence is not affected by the linear term because, in view of (2.4), the model we use and the more standard quadratic model

$$m_s(u) = f(u_c) + (\nabla f(u_c), u - u_c) + (u - u_c, \mathcal{P}_\mathcal{I}\mathcal{R}(u_c)\mathcal{P}_\mathcal{I}(u - u_c))/2. \tag{2.11}$$

agree if

$$u = \mathcal{P}(u_c - \lambda \nabla f(u_c)) \tag{2.12}$$

for some $\lambda > 0$. To see this note that since $\epsilon > 0$, we must have $\mathcal{P}_\mathcal{A}(u - u_c) = 0$ if (2.12) holds.

Algorithm `boxtr` returns a trial point u_t by using Algorithm `trcg` for a the reduced quadratic model.

ALGORITHM 2.4. `boxtr`$(d, u_c, u_t, g, M, \epsilon, \eta, \Delta, kmax)$

1. *Compute* $\mathcal{I} = \mathcal{I}_\epsilon(u_c)$
2. *Find the direction* d *by calling* `trcg`$(d, u_c, \mathcal{P}_\mathcal{I}\nabla f(u_c), \mathcal{R}(u_c), M, \eta, \Delta, kmax)$
3. $u_t = \mathcal{P}(u_c + d)$

Having computed the trial point, one must decide whether to accept the new point or change the trust region radius. Both decisions are based on a comparison of the actual reduction

$$ared = f(u_t) - f(u_c) \tag{2.13}$$

to a predicted reduction based on the reduced quadratic model. Here

$$pred = ((u_t - u_c), \nabla f(u_c)) + ((u_t - u_c), \mathcal{R}(u_c)(u_t - u_c))/2. \tag{2.14}$$

In a typical trust region algorithm, the step is accepted if

$$\frac{ared}{pred} \geq \mu_1, \tag{2.15}$$

the trust region radius is reduced ($\Delta \to \omega_1\Delta, \omega_1 < 1$) if

$$\frac{ared}{pred} < \mu_2, \tag{2.16}$$

and increased ($\Delta \to \omega_2\Delta, \omega_2 > 1$) if

$$\frac{ared}{pred} \geq \mu_3. \tag{2.17}$$

Here $\mu_1 \leq \mu_2 < \mu_3 < 1$. For example, in [8], $\mu_1 = \mu_2 = .25$ and $\mu_3 = .75$. We must add other conditions to our trust region management scheme to account for the possibility that $ared > 0$, *i. e.* the quadratic model is not reduced, which may happen because of our particular choice of model if the trust region radius is too large. For technical reasons, we test for sufficient decrease in the function before accepting the trust region/step combination. We require that for some $\mu_0 \in (0, \mu_1)$

$$f(u_t) - f(u_c) \leq -\mu_0\sigma(u_c)\|u_c - \mathcal{P}(u_c - \hat{\lambda}_c\nabla f(u_c))\|_2, \tag{2.18}$$

where

$$\hat{\lambda}_c = \min\left(\frac{\Delta}{\|\nabla f(u_c)\|_2}, 1\right). \tag{2.19}$$

2.3. Termination. No algorithm that depends upon a measurement of decrease like $ared$ is reliable if the decreases in the function are smaller than the accuracy with which the function is computed. Once we have resolved a local minimum to that point, our view is that the iteration has succeeded.

Hence we terminate the algorithm if either $\sigma(u) < \tau_g$, or

$$|ared| < \tau_f. \tag{2.20}$$

Here τ_g and τ_f are small tolerances. We test for (2.20) every time the trust region radius is changed and if (2.20) holds at any point during an iteration, we terminate and accept the previous iteration. This stopping criterion is used only in the implementation.

2.4. The Complete Algorithm. In the interest of clarity, we do not make the trust region algorithmic parameters, μ_0, μ_1, μ_2, μ_3, μ_4, ω_1, ω_2, $kmax$, the preconditioner M, and the initial trust region radius Δ, formal arguments to the algorithm. The trust region radius is limited to a maximum value Δ_{max}.

The notation is that u_c is the current iteration. Upon exit from the trust region phase of the algorithm, we obtain $u_{1/2}$ and pass that intermediate iterate to the smoother to produce u_+. Choose a constant $\mu_4 \in (0, 1)$.

The inputs are $u \in \mathcal{U}$, the bounds, and the function f.

ALGORITHM 2.5. `trmin`(u, f, u_{max}, u_{min})

1. *Initialize* Δ, $k = 1$.
2. *Test for termination*
   if $\sigma(u) < \tau_g$ *or*
   if $k > 1$ *and* $|ared| < \tau_f$ *terminate successfully*
3. *Fix* η *and* ϵ *for this iterate. Set* $u_c = u$, $rflag = 0$.
4. *Find a new trial point* u_t *using Algorithm* `boxtr`.
   Terminate with failure if more than $kmax$ *iterations are taken.*
5. *Set* $\rho = ared/pred$.
   (a) *if* $\rho < \mu_1$ *or* (2.18) *does not hold, then* $\Delta = \omega_1\Delta$; $rflag = 1$, *go to step 4.*
   (b) *if* $\mu_1 \le \rho < \mu_2$, $\Delta = \omega_1\Delta$, $u_{1/2} = u_t$
   (c) *if* $\rho \ge \mu_2$ *and* $\Delta = \Delta_{max}$, $u_{1/2} = u_t$
   (d) *if* $\mu_2 \le \rho < \mu_3$, *or* $\rho \ge \mu_3$ *and* $rflag = 1$, $u_{1/2} = u_t$
   (e) *if* $\rho \ge \mu_3$ *and* $rflag = 0$, $\Delta = \min(\Delta_{max}, \omega_2\Delta)$, *go to step 4*
6. (a) *Find the smallest integer* $m \ge 0$ *such that*

$$f(\mathcal{P}(u_{1/2} - \beta^m \alpha^{-1} \nabla f(u_{1/2}))) - f(u_{1/2}) < -\mu_4 ared.$$

   then postsmooth, i. e. *set*

$$u_+ = \mathcal{P}(u_{1/2} - \beta^m \alpha^{-1} \nabla f(u_{1/2})).$$

   (b) $u = u_+$; $k = k + 1$; *go to step 2*

The flag $rflag$ is used to avoid an infinite loop of increasing and decreasing the trust region radius.

In the context of a globally convergent algorithm, attention must be paid to the postsmoothing step (6a). The line search prevents divergence in the early phase of the iteration when the approximate solutions are not accurate.

3. Convergence Results. In this section we derive global convergence results for Algorithm `trmin`. Recall that our notation is that u_c (resp. u_k) is the current (resp kth) iteration. Upon exit from the trust region phase of the algorithm, we obtain u_t (resp $u_{k+1/2}$) and pass that intermediate iterate to the smoother to produce u_+ (resp u_{k+1}).

3.1. Global Convergence. Given $u_c \in H$ and a the quadratic model function m_c from (2.10) we must first show that the trust region radius can be bounded from below.

Our assumptions are

ASSUMPTION 3.1.

1. *f is twice continuously differentiable in $\mathcal{U}$.*
2. *There is $r > 0$ such that for all $u \in \mathcal{U}$ and $z \in L^2[0,T]$*

$$\|z\|_2^2/r \leq |(z, \mathcal{R}(u)z)| \leq r\|z\|_2^2$$

and $\|\nabla^2 f(u)\|_2 \leq r$.

The second assumption is sort of a second order sufficiency condition as it is common in the convergence analysis of higher order methods. The finite difference approximation (2.9) satisfies this assumption for h sufficiently small, if the assumption holds for the original Hessian $\mathcal{R}$.

In this section we will use some notation from [2]. For $u \in \mathcal{U}$ define

$$u(\lambda) = \mathcal{P}(u - \lambda \nabla f(u)) \tag{3.1}$$

We begin with the lower bound for the gradient projection step from [2].

THEOREM 3.1. *Let Assumption 3.1 hold and let $\mu \in (0,1)$. Then there is λ_{max} such that for any $0 < \lambda \leq \lambda_{max}$ and $u \in \mathcal{U}$,*

$$f(u) - f(u(\lambda)) \geq \frac{\mu}{\lambda}\|u - u(\lambda)\|_2^2. \tag{3.2}$$

We will also use an lemma from convex analysis. We state this as a special case of a result from [19].

LEMMA 3.2. *Let $\lambda > 0$, $u \in \mathcal{U}$, and f differentiable. Then*

$$\|u - u(\lambda)\|_2 \text{ is an increasing function of } \lambda,$$

$$\lambda^{-1}\|u - u(\lambda)\|_2 \text{ is a decreasing function of } \lambda, \text{ and} \tag{3.3}$$

$$(\nabla f(u), u - u(\lambda)) \leq \|u - u(\lambda)\|_2^2/\lambda$$

From Theorem 3.1 we compute a lower bound on the trust region radius.

THEOREM 3.3. *Let Assumption 3.1 hold. Let $\mathcal{A}$ be computed from* (1.10) *with $\epsilon > 0$. Then there is a C such that on exit from step 5 of algorithm* `trmin`

$$\Delta \geq \Delta_{min} = C\|\nabla f(u_c)\|_2. \tag{3.4}$$

Proof. Since Algorithm `trmin` will, in the worst case of small trust region radius, take steps of the form

$$s = \mathcal{P}(u_c - \lambda \nabla f(u_c)) - u_c$$

the algorithm will therefore behave like the gradient projection algorithm [2], since by (1.11) $\mathcal{P}_{\mathcal{A}} s = 0$. In view of this, we can use known properties of the gradient projection algorithm to bound the trust region radius from below.

Algorithm `boxtr`, with no preconditioner, will return a trial point of the form $\mathcal{P}(u_c - \lambda \nabla f(u_c))$ if only one CG iteration is needed to satisfy the inexactness condition, if $\nabla f(u_c)$ is a direction of negative curvature for $\mathcal{P}_I \mathcal{R}(u_c)$, or if

$$\alpha_0 = \|\nabla f(u_c)\|_2^2 / (\nabla f(u_c), \mathcal{P}_{\mathcal{I}} \mathcal{R}(u_c) \mathcal{P}_{\mathcal{I}} \nabla f(u_c)) > 0$$

satisfies

$$\alpha_0 \|\nabla f(u_c)\|_2 \geq \Delta. \tag{3.5}$$

In this case

$$u_t = u_c(\lambda_t) = \mathcal{P}(u_c - \lambda_t \nabla f(u_c)) \tag{3.6}$$

where $\lambda_t \leq \alpha_0$ is such that

$$\lambda_t = \Delta / \|\nabla f(u_c)\|_2. \tag{3.7}$$

By Assumption 3.1, $|\alpha_0| \geq 1/r$. Hence, if $\Delta < \|\nabla f(u_c)\|_2 / r$ then $\Delta < \|\nabla f(u_c)\|_2 \alpha_0$. Since (3.5) holds, u_t will be given by (3.6), (3.7) holds, and $\lambda_t = \Delta / \|\nabla f(u_c)\|_2 < 1/r \leq 1$.

Now if we set $\mu = \mu_1$ in Theorem 3.1 and

$$\Delta < \|\nabla f(u_c)\|_2 \min(\lambda_{max}, 1/r), \tag{3.8}$$

then the above remarks, Theorem 3.1, and Lemma 3.2 imply that

$$ared \leq \frac{-\mu_1}{\lambda_t} \|u_c - u_c(\lambda_t)\|_2^2 \leq -\mu_1 \|u_c - u_c(\lambda_t)\|_2 \sigma(u_c). \tag{3.9}$$

Since, by (3.7) and (2.19)

$$\lambda_t = \Delta / \|\nabla f(u_c)\|_2 \geq \hat{\lambda}_c$$

we obtain from Lemma 3.2,

$$\|u_c - u_c(\lambda_t)\|_2 \geq \|u_c - u_c(\hat{\lambda}_c)\|_2.$$

Therefore, since $0 < \mu_0 < \mu_1$, from (3.9) we obtain

$$ared \leq -\mu_0 \sigma(u_c) \|u_c - u_c(\hat{\lambda}_c)\|_2$$

which is (2.18).

We summarize the analysis so far. We have shown that if (3.8) holds then $u_t = u_c(\lambda_t)$ for some λ_t and (2.18) holds.

We now describe how the right side of (3.8) must be augmented to imply that $ared/pred \geq \mu_3$. Assume that (3.8) holds. By Lemma 3.2 we have

$$(\nabla f(u_c), u_t - u_c) \leq -\|u_c - u_t\|_2^2 / \lambda_t$$

and hence by the definition of (2.14)

$$\begin{aligned} pred &\leq (\nabla f(u_c), u_t - u_c) + r\|u_c - u_t\|_2^2/2 \\ &\leq (r/2 - 1/\lambda_t)\|u_c - u_t\|_2^2. \end{aligned} \tag{3.10}$$

Note that

$$|ared - pred| = |f(u_t) - m_c(u_t)| \leq r\|u_c - u_t\|_2^2$$

and

$$|pred| \geq |r/2 - 1/\lambda_t|\|u_c - u_t\|_2^2$$

Since $\|u_t - u_c\|_2 \leq \Delta$ we have

$$\begin{aligned} \left|\frac{ared}{pred} - 1\right| &= \left|\frac{ared - pred}{pred}\right| \leq \frac{r\|u_c - u_t\|_2^2}{|r/2 - 1/\lambda_t|\|u_c - u_t\|_2^2} \\ &= \frac{r}{1/\lambda_t - r/2} \leq \frac{r}{\|\nabla f(u_c)\|_2/\Delta - r/2}. \end{aligned} \tag{3.11}$$

Hence, if

$$\Delta \leq \|\nabla f(u_c)\|_2/(r/2 + r/(1-\mu_3))$$

then

$$\frac{ared}{pred} \geq 1 - \left|\frac{ared}{pred} - 1\right| \geq 1 - \frac{r}{\|\nabla f(u_c)\|_2/\Delta - r/2} \geq \mu_3.$$

So, if the trust region radius ever satisfies

$$\Delta \leq \min\left(\|\nabla f(u_c)\|_2 \min(\lambda_{max}, 1/r), \|\nabla f(u_c)\|_2 (r/2 + r/(1-\mu_3))^{-1}\right)/\omega_2$$

then all acceptance tests will be passed and a further increase will be attempted. Hence, on exit from step 5 (3.4) will hold with

$$C = \min(\lambda_{max}, 1/r, (r/2 + r/(1-\mu_3))^{-1})/\omega_2$$

which completes the proof. □

Our main global convergence result is.

THEOREM 3.4. *Let Assumption 3.1 hold and let u_k be the sequence produced by Algorithm* `trmin` *Then*

$$lim_{k\to\infty}\sigma(u_k) = 0.$$

Proof. Step 6 of Algorithm `trmin`, (2.13) and (2.18) yield

$$\begin{aligned} f(u_{k+1}) - f(u_k) &= f(u_{k+1}) - f(u_{k+\frac{1}{2}}) + f(u_{k+\frac{1}{2}}) - f(u_k) \\ &\leq (1-\mu_4)(f(u_{k+\frac{1}{2}}) - f(u_k)) \\ &\leq -(1-\mu_4)\mu_0\sigma(u_k)\|u_k - u_k(\hat{\lambda}_k)\|_2 \end{aligned}$$

with $\hat{\lambda}_k$ defined by (2.19). The boundedness from below of f on $\mathcal{U}$ implies

$$\lim_{k\to\infty} \sigma(u_k)\|u_k - u_k(\hat{\lambda}_k)\|_2 = 0.$$

Since $\hat{\lambda}_k \leq 1$ by (2.19), Lemma 3.2 yields

$$\lim_{k\to\infty} \|u_k - u_k(\hat{\lambda}_k)\|_2 = 0.$$

Theorem 3.3 implies that either $\hat{\lambda}_k = 1$ or

$$\hat{\lambda}_k = \frac{\Delta_k}{\|\nabla f(u_k)\|_2} \geq C.$$

Therefore, with $C_1 = \min(1, C)$, Lemma 3.2 implies that

$$\|u_k - u_k(\hat{\lambda}_k)\|_2^2 \geq \|u_k - u_k(C_1)\|_2^2 \geq C_1^2 \sigma(u_k)^2,$$

completing the proof. □

3.2. Local Convergence. In [14] we gave conditions under which Algorithm `pnlocal` converges locally q-superlinearly to a solution u^*. In view of the lower bound on the trust region radius in (3.4), this will imply fast local convergence if a local minimum that satisfies the assumptions that we will outline in § 3.2.1 hold.

To begin with note that if u_t (in the language of Algorithm `trmin`) is sufficiently near a fixed point of $\mathcal{K}$ in the L^2 norm, then a full smoothing step (*i. e.* $m = 0$) in step 6a of Algorithm `trmin` will be taken and then u_+ will be near u^* in the L^∞ sense. Morevoer, if u^* satisfies the assumptions we outline in § 3.2.1, a full inexact projected Newton step will be taken from u_c to u_t.

Hence once u_k is L^2 near to u^*, u_{k+1} will be close in the L^∞ sense and then the active and inactive sets will be accurately approximated. This will be important for the local convergence result as it was in [11] and [14].

3.2.1. Assumptions for Superlinear Convergence. We begin by reviewing the assumptions required for superlinear convergence of inexact projected Newton methods from [14]. We will express those assumptions in the less general language of the present paper.

If u^* is a local minimizer that satisfies Assumptions 3.2 and 3.3 and the inexact projected Newton point is in the trust region, and the current iterate is sufficiently near u^* (in the L^2 sense) then the iteration will converge to u^* at a rate determined by the sequences $\{\epsilon_n\}$ and $\{\eta_n\}$.

The first assumption is the minimal regularity needed for any projected Newton method to converge rapidly.

ASSUMPTION 3.2. *There is a solution u^* to* (1.1)–(1.2) *subject to $u \in \mathcal{U}$ such that f is twice Lipschitz continuously Fréchet differentiable at u^* in any $L^q[0,T]$ for $1 \leq q \leq \infty$ and in $C[0,T]$ and $\mathcal{K}_0$ is Lipschitz continuous as a map from $L^2[0,T]$ to $C[0,T]$. There are $\epsilon_{max}, \rho_{max}, M > 0$ such that for all $\epsilon \in (0, \epsilon_{max})$ and all u such that $\|u - u^*\|_\infty < \rho_{max}$ the approximate reduced Hessian computed using* (2.6) *is nonsingular. Moreover*

$$\|\mathcal{K}_0'(u)\|, \|\mathcal{R}(u)\|, \|\mathcal{R}(u)^{-1}\| \leq M \tag{3.12}$$

where the norm in (3.12) *is any of* $\mathcal{L}(L^q, C[0,T])$, $1 \leq q \leq \infty$ *or* $\mathcal{L}(C[0,T])$.

The other assumption is related to a nondegeneracy assumption on the optimal control and a generalization of the fact that for finite dimensional problems the active set is identified in finitely many steps. We refer the reader to [14] for motivation and discussion.

We define sets

$$\mathcal{I}_+ = \{t \mid F(u)(t) = \nabla f(u)(t)\} \cap \mathcal{I} \text{ and } \mathcal{I}_- = \{t \mid F(u)(t) \neq \nabla f(u)(t)\} \cap \mathcal{I} \tag{3.13}$$

Clearly $[0,T] = \mathcal{I}_+ \cup \mathcal{I}_- \cup \mathcal{A}$.

ASSUMPTION 3.3. *There is* $\nu > 0$ *such* $u_{min}(t) + \nu \leq u_{max}(t)$ *for all* $t \in [0,T]$. *Let* $\mathcal{A}$ *be given by* (2.3) *and let* $p \in (0,1)$ *be given. Then there are* $\epsilon_{max}, \rho_{max}, M > 0$ *such that for all* $\epsilon \in (0, \epsilon_{max})$ *and all* $u = \mathcal{K}(v)$ *such that* $\|v - u^*\|_2 < \rho_{max}$

$$\|\mathcal{P}_{\mathcal{A}}(u - u^*)\|_2 \leq M\epsilon \|u - u^*\|_\infty, \tag{3.14}$$

$$m(\mathcal{I}_-) \leq M\epsilon, \tag{3.15}$$

and

$$\|u - \mathcal{P}(u - \nabla f(u))\|_\infty / M \leq \|u - u^*\|_\infty \leq M \|u - \mathcal{P}(u - \nabla f(u))\|_\infty. \tag{3.16}$$

In Assumption 3.3, m denotes Lebesgue measure.

On these assumptions we have the following local convergence result. The proof is, on the assumptions made above, a variation of those in [11] and [14].

THEOREM 3.5. *Let Assumptions 3.1, 3.2, and 3.3 hold. Let* $M > 0$ *and* $p \in (0,1)$ *be given. Then if* u_- *is sufficiently near* u^* *in* L^2, $u_c = \mathcal{K}(u_-)$,

$$\eta_c, \epsilon_c \leq M\sigma(u_c)^p, \tag{3.17}$$

and u_t *and* u_+ *are given by Algoritm* `trmin` *then* $s_t = u_t - u_c$ *satisfies* (2.7) (i. e. *a full inexact Newton step is taken*),

$$\|u_t - u^*\|_2 = O(\|u_c - u^*\|_\infty^{1+p}),$$

and

$$\|u_+ - u^*\|_\infty = O(\|u_c - u^*\|_\infty^{1+p}).$$

4. Numerical Example. All the results reported in this section were obtained on a SUN SPARC-20 running SUN fortran f77 version SC3.0.1 and SUN-OS 4.1.3.

Our numerical results are based on the problem posed by (1.1), (1.2), (1.3), (1.5), and (1.6). We set

$$g(y) = y, T = 1, z(x) = 6\cos(x(1-x)), \text{ and } \alpha = .01.$$

We report results for both the constrained and unconstrained problems. Our constraints are given by

$$u_{min}(t) = 2.75t \text{ and } u_{max}(t) = 4 + 10\sqrt{t}.$$

In our numerical examples we discretized in space with piecewise linear finite elements as we did for the multilevel results reported in [14], and integrated in time with the DAE solver DASPK (used in DASSL mode) [4], [5], [6], [15].

We used the trust region parameters

$$\mu_0 = \mu_1 = 10^{-4}, \mu_2 = .25, \mu_3 = .75, \omega_1 = .5, \omega_2 = 2. \text{ and } \Delta_{max} = 5.$$

The radius of the trust region was initialized to $\Delta_{max} = 5$. In the test for postsmoothing we used $\mu_5 = 10$.

The approximate active set was computed with

$$\epsilon = \min(\sigma(u)^{.5}, \delta_x/2)$$

Here $\delta_x = 1/(N+1)$ where N is the number of nodes. The forcing term was

$$\eta = \min(\sigma(u)^{.5}, .01).$$

For a mesh of size δ_x we use the tolerances

$$\tau_g = 10\delta_x^2, \tau_f = \tau_g/1000 = \delta_x^2/100,$$

and set the relative and absolute local truncation errors in DASPK to be

$$rtol = atol = \delta_x^2 * 1.d - 3.$$

Following [14] we limited the time step in DASPK to the spatial mesh width δ_x. Our difference increment h in (2.9) for the numerical reduced Hessian was

$$h = \delta_x/2.$$

The initial iterate was

$$u_0(x) = 3x$$

for both the constrained and unconstrained problems.

The solutions for the unconstrained (left) and constrained (right) problems, both with $\delta_x = 1/639$ are plotted in Figure 4.1. From the plot of the constrained minimizer on the right one can see that both the upper and lower bound constraints are attained on different parts of $[0, T]$.

For all computations we tabulate the iteration counter k, the value of the objective $f(u_k)$, the actual reduction (for $k \geq 1$), the norm of the projected gradient, $\sigma(u_k)$, the number of CG iterations required i_k (for $k \geq 1$), and the radius Δ of the trust region. For the constrained problem we also tabulate P_A, the fraction of points that are in the approximate active set. $i_k = 0$ means that the steepest descent or gradient projection step either went beyond the trust region boundary or satisfied the inexact Newton condition.

The iterations for the both the constrained and unconstrained problem, reported in Tables 4.1 and 4.2 terminated when $\sigma(u) < \tau_g$. For the unconstrained problem, full smoothing steps were taken at each iteration. For the constrained problem, a full smoothing step was taken at the final iteration only.

FIG. 4.1. *Minimizers,* $\delta_x = 1/639$

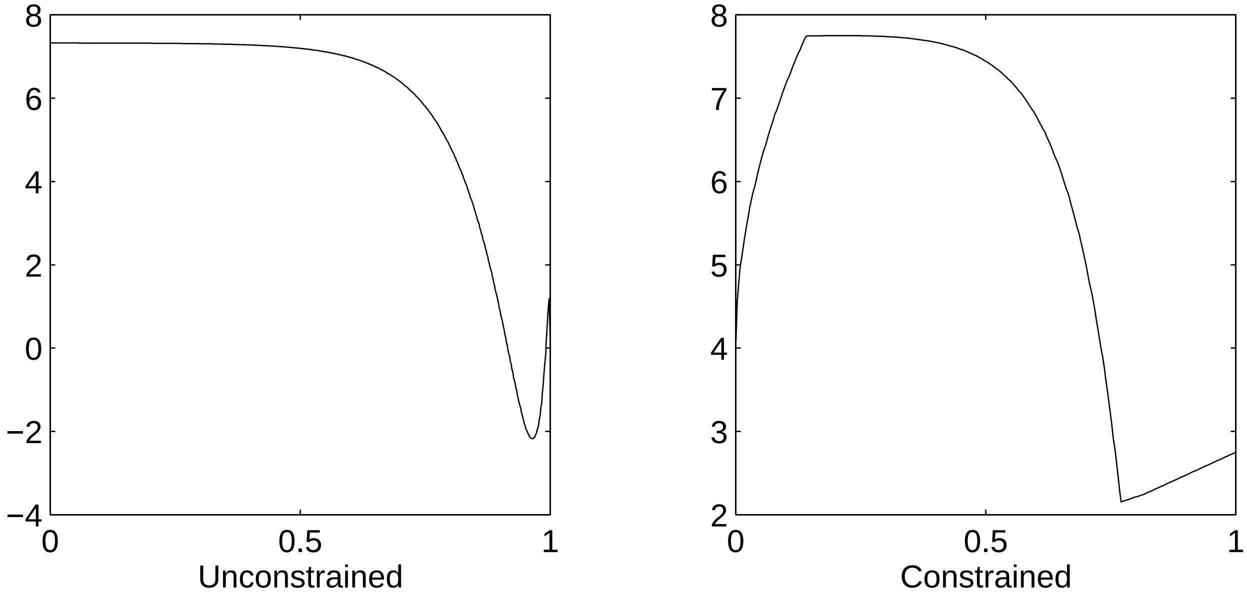

TABLE 4.1
Unconstrained Problem, h=1/639.

k	$f(u_k)$	$ared$	$\sigma(u_k)$	i_k	Δ
0	9.77e+00		4.33e+00		5.00
1	2.81e-01	-9.11e+00	2.45e-01	0	5.00
2	2.21e-01	-5.99e-02	3.30e-02	2	5.00
3	2.20e-01	-1.41e-03	1.32e-02	1	5.00
4	2.19e-01	-6.98e-04	2.83e-02	2	5.00
5	2.19e-01	-3.88e-04	1.00e-02	0	5.00
6	2.19e-01	-3.85e-04	8.75e-04	1	5.00
7	2.19e-01	-2.15e-06	5.93e-05	4	5.00
8	2.19e-01	-4.44e-08	1.34e-05	3	5.00

TABLE 4.2
Constrained Problem, h=1/639.

k	$f(u_k)$	$ared$	$\sigma(u_k)$	i_k	Δ	P_A
0	9.77e+00		4.33e+00		5.00	
1	2.60e+00	-9.11e+00	1.91e+00	0	5.00	0.209
2	1.53e+00	-2.30e+00	1.49e+00	0	5.00	0.025
3	7.90e-01	-1.23e+00	9.81e-01	1	5.00	0.000
4	2.81e-01	-4.90e-01	2.63e-02	1	5.00	0.141
5	2.78e-01	-1.88e-03	7.24e-03	1	0.61	0.223
6	2.78e-01	-4.32e-05	2.64e-03	1	0.03	0.322
7	2.78e-01	-1.06e-04	8.88e-03	1	0.26	0.350
8	2.78e-01	-6.04e-05	5.87e-03	0	0.26	0.331
9	2.78e-01	-2.16e-05	9.12e-04	1	0.02	0.375
10	2.78e-01	-6.21e-07	6.70e-04	1	0.02	0.380
11	2.78e-01	-3.19e-07	1.28e-05	0	0.02	0.380

REFERENCES

[1] L. ARMIJO, *Minimization of functions having Lipschitz-continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[2] D. B. BERTSEKAS, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Autom. Control, 21 (1976), pp. 174–184.

[3] ———, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[4] K. E. BRENAN, S. L. CAMPBELL, AND L. R. PETZOLD, *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, no. 14 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[5] P. N. BROWN, A. C. HINDMARSH, AND L. R. PETZOLD, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[6] ———, *Consistent initial condition calculation for differential-algebraic systems*, Tech. Report UCRL-JC-122175, Lawrence Livermore National Laboratory, August 1995.

[7] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Global convergence of a class of trust region algorithms for optimization problems with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[8] ———, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[9] J. W. DANIEL, *The Approximate Minimization of Functionals*, Prentice-Hall, Englewood Cliffs, 1971.

[10] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[11] C. T. KELLEY, *Identification of the support of nonsmoothness*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. Pardalos, eds., Boston, 1994, Kluwer Academic Publishers B.V., pp. 192–205.

[12] ———, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[13] C. T. KELLEY AND E. W. SACHS, *Fast algorithms for compact fixed point problems with inexact function evaluations*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 725–742.

[14] ———, *Multilevel algorithms for constrained compact fixed point problems*, SIAM J. on Sci. Comput., 15 (1994), pp. 645–667.

[15] L. R. PETZOLD, *A description of DASSL: a differential/algebraic system solver*, in Scientific Computing, R. S. Stepleman et al., ed., North Holland, Amsterdam, 1983, pp. 65–68.

[16] 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., Academic Press, New York, 1975, pp. 1–27.

[17] E. SACHS, *A parabolic control problem with a boundary condition of the Stefan-Boltzmann type*, ZAMM, 58 (1978), pp. 443–449.

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

[19] P. L. TOINT, *Global convergence of a class of trust–region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[20] R. WINTHER, *A numerical Galerkin method for a parabolic problem*, PhD thesis, Cornell University, Ithaca, New York, 1977.

[21] J. P. YVON, *Controle optimal d'un four industriel*, Tech. Report 22, INRIA, 1973.