SIAM J. NUMER. ANAL.
Vol. 46, No. 6, pp. 3071–3083

PROJECTED PSEUDOTRANSIENT CONTINUATION*

C. T. KELLEY†, LI-ZHI LIAO‡, LIQUN QI§, MOODY T. CHU†, J. P. REESE¶, AND C. WINTON†

We dedicate this paper to the memory of H. B. Keller

Abstract. We propose and analyze a pseudotransient continuation algorithm for dynamics on subsets of R^N. Examples include certain flows on manifolds and the dynamic formulation of bound-constrained optimization problems. The method gets its global convergence properties from the dynamics and inherits its local convergence properties from any fast locally convergent iteration.

Key words. pseudotransient continuation, constrained dynamics, gradient flow, bound-constrained optimization, quasi-Newton method

AMS subject classifications. 65H10, 65H20, 65K10, 65L05

DOI. 10.1137/07069866X

1. Introduction. In this paper we extend algorithms and convergence results [9, 12, 13, 15, 24] for the method of pseudotransient continuation (Ψtc) to a class of constrained problems in which projections onto the feasible set are easy to compute. Such constraints arise in inverse eigenvalue and singular value problems [7, 8], where Ψtc is an efficient alternative to a fully time-accurate geometric integration [14], for which theory is only available for fixed time-step methods. The objective of Ψtc is not to stay on the manifold with high precision but rather to move rapidly to the steady state.

The algorithms we propose use the projection and can hence be applied to other classes of problems. Bound-constrained optimization is one example [22] and we explore that in this paper as well. The results in this paper may also be applicable to more general problems on manifolds [1] if the relevant projections can be approximated efficiently, and this will be the subject of future work.

Ψtc was originally designed as a method for finding steady-state solutions to time-dependent differential equations. The idea is to mimic integration to the steady state while managing the "time step" to move the iteration as rapidly as possible to Newton's method. This is different from the standard approach in an algorithm for initial value problems [2], where the time step is controlled with stability and accuracy in mind. Ψtc also differs from traditional continuation methods in that

*Received by the editors July 30, 2007; accepted for publication (in revised form) April 30, 2008; published electronically September 4, 2008.
http://www.siam.org/journals/sinum/46-6/69866.html

†Center for Research in Scientific Computation and Department of Mathematics, North Carolina State University, Box 8205, Raleigh, NC 27695-8205 (Tim_Kelley@ncsu.edu, mtchu@ncsu.edu, cwwinton@ncsu.edu). The work of these authors was partially supported by National Science Foundation grants DMS-0404537 and DMS-0707220 and Army Research Office grants W911NF-04-1-0276, W911NF-06-1-0096, W911NF-06-1-0412, and W911NF-07-1-0112.

‡Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong, China (liliao@hkbu.edu.hk). The work of this author was partially supported by the Research Grant Council of Hong Kong.

§Department of Applied Mathematics, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China (maqilq@polyu.edu.hk). The work of this author was partially supported by the Research Grant Council of Hong Kong.

¶School of Computational Science, Florida State University, Dirac Science Library, Tallahassee, FL 32306-4120 (jreese@scs.fsu.edu).

the objective is to find a steady-state solution, not, as is the case for pseudoarclength continuation [20], to track that solution as a function of another parameter. Homotopy methods [31] also introduce an artificial parameter to solve nonlinear equations but not in a way that is intended to capture dynamic properties, such as stability, of the solution.

In the remainder of this section, we will briefly review Ψtc. We refer the reader to [9, 12, 13, 15, 24] for the existing convergence results and references to applications.

In section 2 we will describe an algorithm for constrained Ψtc and prove a convergence theorem which not only allows for constraints but has a weaker stability assumption than was used in [9, 13, 24] and includes more general assumptions on the iteration itself. We close section 2 with some remarks on applications of the theory. In section 3, we will show how the new form of Ψtc can be applied to bound-constrained optimization in a way that maintains superlinear convergence in the terminal phase of the iteration. In section 4 we apply the methods to two example problems.

1.1. Ψtc for nonlinear equations. The formulation for ODE dynamics is the easiest to understand and is sufficient for this paper. Suppose $F : R^N \to R^N$ is Lipschitz continuously differentiable and

$$u^* = \lim_{t\to\infty} u(t), \tag{1.1}$$

where u is the solution of the initial value problem

$$\frac{du}{dt} = -F(u), \qquad u(0) = u_0. \tag{1.2}$$

We will refer to (1.1) as a stability condition in what follows, and it is a property of both F and the initial point u_0. The objective of the algorithms we discuss in this paper is to find u^*.

One might try to find u^* by solving the nonlinear equation $F(u) = 0$ with a globalized version of Newton's method [21, 23]. The danger is that one may find a solution other than u^* and even a solution that is dynamically unstable. One could also use an ODE code to accurately integrate the initial value problem (1.2) to the steady state. The problem with this latter approach is that one is accurately computing transient behavior of u that is not necessarily needed to compute u^*.

The most common form of Ψtc is the iteration

$$u_+ = u_c - \left(\delta_c^{-1} I + F'(u_c)\right)^{-1} F(u_c), \tag{1.3}$$

where, as is standard, u_c is the current iteration, δ_c the current time step, and u_+ the new iteration. The "time step" δ is managed in a way that captures important transients early in the iteration but grows near u^* so that (1.3) becomes Newton's method. One common way to control δ is "switched evolution relaxation" (SER) [29]. In SER the new time step is

$$\delta_+ = \min(\delta_c \|F(u_c)\|/\|F(u_+)\|, \delta_{max}). \tag{1.4}$$

Using $\delta_{max} = \infty$ is common, in which case $\delta_+ = \delta_0 \|F(u_0)\|/\|F(u_+)\|$. We will refer to the updated formula (1.4) as SER-A, to distinguish it from (1.5).

The existing convergence theory applies only to SER-A, but there are other approaches, and we will discuss two. A variation of SER, which we call SER-B, was proposed in [19]. The formula for the time step is

$$\delta_+ = \max(\delta_c/\|u_+ - u_c\|, \delta_{max}). \tag{1.5}$$

In section 2.1.1, we show how SER-B can be modified so that the convergence theory applies in the case of certain gradient flows.

The temporal truncation error approach (TTE) [25] estimates the local truncation error of $(u)_i(t_n)$, the ith component of $u(t_n) \in R^N$, by $\tau_i \equiv \frac{\delta_n^2 (u)_i''(t_n)}{2}$, approximates $(u)_i''$ by

$$\frac{2}{\delta_{n-1}+\delta_{n-2}}\left[\frac{((u)_i)_n - ((u)_i)_{n-1}}{\delta_{n-1}} - \frac{((u)_i)_{n-1} - ((u)_i)_{n-2}}{\delta_{n-2}}\right], \tag{1.6}$$

and computes δ_n by setting $\tau_i = 3/4$ for all i. In section 4 we will compare the three methods for time-step management. The good performance of SER-B and TTE raises interesting research questions.

One way to see how Ψtc and temporal integration are related is to derive Ψtc from the implicit Euler method. The formula for an implicit Euler step is $u^{n+1} = u^n - \delta_n F(u^{n+1})$, where u^n is the approximation to u at t_n and $\delta_n = t_{n+1} - t_n$ is the nth time step. If we determine u^{n+1} by taking a single Newton step for the nonlinear equation $G(u) \equiv u - u^n + \delta_n F(u) = 0$, with u^n as the initial iterate, then we obtain (1.3).

2. Constrained Ψtc . Let F be Lipschitz continuous, and assume that $u(t) \in \Omega$ for all $t \geq 0$, where $\Omega \subset R^N$. Examples of such constrained dynamics are flows where F is the projected gradient onto the tangent space of Ω at u, and our two examples are such flows. One should not expect a general purpose integrator to keep the solutions in Ω, and we use a projection to correct after each step to keep the iterations in Ω.

Let $\mathcal{P}$ be a Lipschitz continuous projection onto Ω. Our assumptions on $\mathcal{P}$ are as follows.

Assumption 2.1.

1. $\mathcal{P}(u) = u$ for all $u \in \Omega$.
2. There are $M_{\mathcal{P}}$ and $\epsilon_{\mathcal{P}}$ such that for all $u \in \Omega$ and v such that $\|v - u\| \leq \epsilon_{\mathcal{P}}$

$$\|\mathcal{P}(v) - u\| \leq \|v - u\| + M_{\mathcal{P}}\|v - u\|^2. \tag{2.1}$$

Assumption 2.1 is trivially true if Ω is convex and $\mathcal{P}$ is the projection onto Ω, for then $\mathcal{P}$ is Lipschitz continuous with Lipschitz constant 1, so $M_{\mathcal{P}} = 0$. If Ω is a smooth manifold of the form $\Omega = \{u \,|\, \mathcal{F}(u) = 0\}$ where $\mathcal{F} : R^N \to R^M$ with $M < N$, and $\mathcal{P}$ is smooth, then $\|\mathcal{P}'(u)\| = 1$, which will imply (2.1). Note that we are making no explicit assumptions on Ω, only assumptions on the existence of $\mathcal{P}$ and its properties.

We will consider a Ψtc iteration of the form

$$u_+ = \mathcal{P}\left[u_c - \left(\delta_c^{-1} I + H(u_c)\right)^{-1} F(u_c)\right], \tag{2.2}$$

where H is an $N \times N$ matrix-valued function of u. We will assume that H is a sufficiently good approximation to F' (or, in the semismooth case, sufficiently close to ∂F) to make the iteration locally convergent. The theory we will develop applies equally well to the inexact formulation $u_+ = u_c + s$, where

$$\|(\delta_c^{-1} I + H(u_c))s + F(u_c)\| \leq \eta_c \|F(u_c)\|. \tag{2.3}$$

The "forcing term" η_c could be, for example, the termination tolerance in a linear iterative method for computing the step [10, 21]. The forcing term has a well-known effect on convergence analysis, which is reflected in (2.11).

Assumption 2.2.

1. There are $M_H, \epsilon_H > 0$ such that

$$\|H(u)\| \le M_H \text{ for all } u \in S(\epsilon_H) = \left\{ z \mid \inf_{t \ge 0} \|z - u(t)\| \le \epsilon_H \right\}. \tag{2.4}$$

   For all $\epsilon > 0$ there is $\bar{\epsilon} > 0$ such that if $u \in S(\epsilon_H)$ and $\|u - u^*\| > \epsilon$, then

$$\|F(u)\| > \bar{\epsilon}. \tag{2.5}$$

2. There is ϵ_L so that if $\|u_c - u^*\| \le \epsilon_L$, then $H(u_c)$ is nonsingular,

$$\|(I + \delta H(u_c))^{-1}\| \le (1 + \beta\delta)^{-1}, \text{ for some } \beta > 0 \text{ and all } \delta \ge 0, \tag{2.6}$$

   and the local Newton iteration

$$u_+^{NL} = u_c - H(u_c)^{-1} F(u_c) \tag{2.7}$$

   reduces the error by a factor $r \in [0, 1)$ for all $u_c \in \Omega$ sufficiently near u^*, i.e.,

$$\|e_+^{NL}\| \le r \|e_c\|. \tag{2.8}$$

One new feature in Assumption 2.2 is the local nonlinear iteration, which is general enough to allow for Gauss–Newton methods (and quasi-Newton methods, see section 2.1.3). Note that the convergence rate for the local iteration is expressed in terms of the underlying unprojected Newton-like method (2.7). The assumption we must make (2.8) on the unprojected iteration can be verified in the examples we consider in section 4.

Theorem 2.1 extends previous work in several ways. The smoothness assumptions on F are relaxed, the projection is introduced to handle constrained dynamics, H is constrained only by the local convergence behavior of the Newton-like iteration (2.7), so superlinear convergence is not required, and (2.6) need only hold in a neighborhood of u^*.

THEOREM 2.1. *Let F be locally Lipschitz continuous, and assume that*

$$\lim_{t \to \infty} u(t) = u^*$$

and that Assumptions 2.1 *and* 2.2 *hold. Let the sequence $\{\delta_n\}$ be updated with* (1.4). *Assume that there is $\delta^* > 0$ such that*

$$M_{\mathcal{P}} \epsilon_L / \beta < \delta^* \le \delta_n \tag{2.9}$$

for all n. Assume that the q-factor r in (2.8) *satisfies*

$$r < ((1 + M_{\mathcal{P}} \epsilon_L) - (1 + \beta\delta^*)^{-1})/2, \tag{2.10}$$

where β is the constant in (2.6). *Then if δ_0 and the sequence $\{\eta_n\}$ are sufficiently small, the inexact Ψtc iteration $u_{n+1} = \mathcal{P}(u_n + s_n)$, where $\|(\delta_n^{-1} I + H(u_n))s_n + F(u_n)\| \le \eta_n \|F(u_n)\|$ converges to u^*. Moreover, there is $K > 0$ such that for n sufficiently large*

$$\|e_{n+1}\| \le \|e_{n+1}^{NL}\| + K \|e_n\| \left(\eta_n + \delta_n^{-1}\right). \tag{2.11}$$

Proof. We will prove the result for the exact ($\eta_n = 0$) iteration with $\delta_{max} = \infty$ in (1.4). The complete proof is based on the same ideas but requires more bookkeeping. The outline of the proof follows those in [9, 13, 24].

We begin with the global phase. We wish to prove that, while u is out of the local convergence region for the iteration (2.7), the iteration remains close to the solution of the differential equation, i.e., in $S(\epsilon)$ for a sufficiently small ϵ. Only (1.1), the lower bound $\delta_n \geq \delta^*$, Lipschitz continuity of F, and Part 2.2 of Assumption 2.2 are needed for this stage of the analysis. Having done this, we address the local phase, where δ is small and u is near u^*. We will use the rest of Assumption 2.2 to prove the local convergence estimate (2.11).

Let $\epsilon < \min(\epsilon_L, \epsilon_H)$. We may reduce ϵ as the proof of the local convergence progresses. The first step is to show that if δ_0 is sufficiently small, then

$$\|u_n - u^*\| < \epsilon \tag{2.12}$$

for sufficiently large n. To do this we need only verify that, for δ sufficiently small,

$$\left(\delta^{-1} I + H(u_n)\right)^{-1} = \delta I + O\left(\delta^2\right), \tag{2.13}$$

and obtain upper and lower bounds on δ_n/δ_0 while (2.12) fails to hold.

The estimate (2.13) follows from (2.4) if $\delta < 1/(2M_H)$. To obtain bounds for δ_n, we can apply the update formula (1.4) and (2.5) to show that while $u_n \in S(\epsilon_H)$ and $\|u_n - u^*\| > \epsilon/2$, then

$$\begin{aligned} \delta^* &\equiv \delta_0 \|F(u_0)\| / \max_{u \in S(\epsilon_H)} \|F(u)\| \\ &\leq \delta_n = \delta_0 \|F(u_0)\| / \|F(u_n)\| \leq 2\delta_0 \|F(u_0)\| / C_F \epsilon. \end{aligned} \tag{2.14}$$

Equation (1.4) with $\delta_{max} = \infty$ and our lower bound on δ imply that

$$\delta^* \leq \delta_n \leq \frac{\|F(u_0)\|\delta_0}{\|F(u_n)\|} \leq \frac{2\delta_0 \|F(u_0)\|}{C_F \epsilon}. \tag{2.15}$$

Hence, for δ_0 sufficiently small, (2.13) holds if (2.12) does not.

With (2.13) in hand, we see that either (2.12) holds or

$$u_{n+1} = \mathcal{P}(u_n - \delta F(u_n)) + O\left(\delta^2\right), \tag{2.16}$$

where the constant in the O-term is independent of n. Now, if $u \in \Omega$, then Lipschitz continuity of $\mathcal{P}$ implies that

$$\mathcal{P}\left(u - \left(\delta^{-1} I + H(u)\right)^{-1}\right) F(u) = \mathcal{P}(u - \delta F(u)) + O\left(\delta^2\right).$$

Euler's method, $u_{n+1} = u_n - \delta F(u)$, has the same local truncation error as $u_{n+1} = \mathcal{P}(u_n - \delta F(u))$, because $u(t) \in \Omega$ for all t [26]. To see this, we note that

$$\mathcal{P}(u(t) - \delta F(u(t))) = \mathcal{P}\left(u(t+\delta) + O\left(\delta^2\right)\right) = u(t+\delta) + O\left(\delta^2\right).$$

Now let T be such that $\|u(t) - u^*\| < \epsilon/2$ for all $t \geq T$, and let N^* be the least integer $\geq T/\delta^*$. The standard analysis for the forward Euler method [2] implies that there is C_E such that

$$\|u_n - u(t_n)\| \leq C_E \max_{1 \leq k \leq n} \delta_k \leq \frac{2C_E \delta_0 \|F(u_0)\|}{C_F \epsilon}$$

for all $n \leq N^*$. In particular, $\|u_n - u^*\| \leq \epsilon$ for all $n \geq N^*$ if $\delta_0 \leq (C_F \epsilon^2)/(4C_E \|F(u_0)\|)$.

For the local phase, assume that $n \geq N^*$. Hence, (2.12) holds. We need to show that $\|e_{n+1}\| < \|e_n\|$ and that $\delta_{n+1} > \delta_n$. Once those things are done, we can complete the proof with a simple calculation.

Define $v_{n+1} = u_n - (\delta_n^{-1} I + H(u_n))^{-1} F(u_n)$, and note that $(H(u_n)^{-1} - (\delta_n^{-1} I + H(u_n))^{-1}) = (I + \delta_n H(u_n))^{-1} H(u_n)^{-1}$. Hence

$$\begin{aligned} v_{n+1} &= u_n - H(u_n)^{-1} F(u_n) + (H(u_n)^{-1} - (\delta_n^{-1} I + H(u_n))^{-1}) F(u_n) \\ &= u_{n+1}^{NL} + (I + \delta_n H(u_n))^{-1} H(u_n)^{-1} F(u_n) \\ &= u_{n+1}^{NL} - (I + \delta_n H(u_n))^{-1} \left(u_{n+1}^{NL} - u_n \right), \end{aligned}$$

and therefore $u_{n+1} = \mathcal{P}(v_{n+1}) = \mathcal{P}(u_{n+1}^{NL} - (I + \delta_n H(u_n))^{-1} (e_{n+1}^{NL} - e_n))$.

We use (2.1) to conclude that

$$\begin{aligned} e_{n+1} &= \mathcal{P}(v_{n+1}) - u^* = \mathcal{P}\left(u_{n+1}^{NL} - (I + \delta_n H(u_n))^{-1} \left(e_{n+1}^{NL} - e_n \right) \right) - u^* \\ &= \mathcal{P}\left(u_{n+1}^{NL} - (I + \delta_n H(u_n))^{-1} \left(e_{n+1}^{NL} - e_n \right) \right) - \mathcal{P}(u^*). \end{aligned}$$

So, since $\epsilon < \epsilon_L$,

$$\begin{aligned} \|e_{n+1}\| &\leq (1 + M_{\mathcal{P}} \epsilon_L) \left(\|e_{n+1}^N\| + \|(I + \delta_n H(u_n))^{-1}\| \left(\|e_{n+1}^{NL}\| + \|e_n\| \right) \right) \\ &\leq (1 + M_{\mathcal{P}} \epsilon_L) \left(2r + (1 + \beta \delta^*)^{-1} \right) \|e_n\|. \end{aligned}$$

This completes the proof since $(1 + M_{\mathcal{P}} \epsilon_L)(2r + (1 + \beta \delta^*)^{-1}) < 1$ by (2.9). Hence, the iteration converges at least locally q-linearly. Formula (1.4) then will imply (2.11). ∎

2.1. Remarks. Even in the unconstrained case (where $\mathcal{P}(u) = u$ for all $u \in R^N$) Theorem 2.1 extends the results from [9, 13, 24] by replacing the semismoothness and the inexact Newton condition with a general condition on the convergence of the local iteration. If the stability condition (1.1) holds, then Ψtc with SER is a convergent iteration for unconstrained optimization, even if one does not use exact Hessians.

2.1.1. Control of δ with f for gradient flows. The key parts to the proof of Theorem 2.1 are showing that the time step remains small until u_n is near the solution and that the step will grow at that time. The way this is done in the case of SER-A is to note that (1.4) implies that if u_n is not close to u^*, then δ_n is bounded from above and below by constant multiples of δ_0, and hence the method is an accurate temporal integration. Then, once near u^*, (2.6) drives u_n to u^* and δ_n to δ_{max}.

One can augment the time-step control method in several ways without affecting the theory. If the dynamics are a gradient flow, then one can reject the step if f is increased, reduce δ, and try again. As long as δ is bounded from above and below by constant multiples of δ_0 and the number of reductions is bounded, the global convergence assertion $u_n \to u^*$ will hold. If one accepts the SER-A formula only when f decreases, then the local theory will be unchanged and Theorem 2.1 will hold. One can also apply this idea to SER-B and TTE but must take care that increases in δ are not too large to maintain the resolution of the dynamics while $u(t)$ is far from u^*. One way to do this is to accept the SER-B or TTE step only if f decreases and limit the size of the increases in δ to factors of two at each iteration.

Another approach is the trust-region method from [15], and the proof of Theorem 2.1 extends to that method. Here $\delta_{max} = \infty$ and the increase in δ at each iteration is limited.

2.1.2. Semismooth nonlinearities. If F is semismooth, which is the case considered in [13], and $H(u_n) \in \partial F(u_n)$, then the local iteration (2.7) is superlinearly convergent, if all matrices in $\partial F(u^*)$ are nonsingular. Hence, we may recover the results from [13] from Theorem 2.1, the extension to DAE dynamics being the same as that in [13] and not relevant to this paper.

2.1.3. Quasi-Newton methods. The proof of Theorem 2.1 can be modified to allow a quasi-Newton [11, 22] model Hessian. This modification simply replaces the dependence of H on u_n with one on the history of the iteration and makes the assumptions of boundedness of H_n and convergence of the local iteration. The resulting modified proof would be somewhat longer, and, since our examples are not quasi-Newton iterations, we used the shorter and simpler proof in this paper.

Unlike Newton or Gauss–Newton iterations, one must assume (rather than verify) the convergence of the local iteration. The reason for this is that local convergence of quasi-Newton methods requires that both the initial iterate and the initial model Hessian be good approximations to the solution and the Hessian at the solution. The global theory [5] requires a line search and convex level sets, neither of which is a part of the Ψtc methods we propose.

2.1.4. Gauss–Newton iteration. If $F(u) = \nabla f(u) = R'(u)^T R(u)$ is the gradient of a nonlinear least squares functional $f(u) = R(u)^T R(u)/2$, where $R : R^N \to R^M$, with $M > N$, we may let H be the Gauss–Newton model Hessian $H(u) = R'(u)^T R'(u)$ and apply Theorem 2.1 if R' has full column rank at the minimizer. Moreover, many of the assumptions can be verified with ease in this case. If $u(t) \to u^*$, which we still must assume, then $H(u^*)$ is symmetric and positive definite, so (2.6) holds. Moreover $\delta^{-1} I + H(u)$ is nonsingular for all $\delta > 0$ and all u, because $H(u)$ is always nonnegative definite. For a zero-residual problem, the estimate (2.10) will hold because the local iteration converges q-quadratically if R is Lipschitz continuously differentiable. In this case, Ψtc is a version of the Levenberg–Marquardt method, where the parameter is selected based on the norm of the gradient rather than with a trust region scheme. Similar ideas for selection of the parameter have been made [22].

3. Bound-constrained optimization. The bound-constrained optimization problem is

$$\min_{\Omega} f(u), \tag{3.1}$$

where $\Omega = \{u \,|\, L \le u \le U\}$, and the inequalities are componentwise.

In order to describe necessary conditions and formulate the algorithms, we must recall some notation from [4, 22, 27].

The l^2 projection onto Ω is $\mathcal{P}$, where

$$\mathcal{P}(u)_i = \max(L_i, \min(U_i, (u)_i)). \tag{3.2}$$

Here $(u)_i$ denotes the ith component of the vector $u \in R^N$. $\mathcal{P}$ trivially satisfies Assumption 2.1 because Ω is convex.

We will assume that f is Lipschitz continuously differentiable. In that case, the first-order necessary conditions for optimality are [4, 22]

$$F(u) = u - \mathcal{P}(u - \nabla f(u)) = 0. \tag{3.3}$$

Equation (3.3) is a semismooth nonlinear equation. Fast locally convergent methods include the semismooth Newton, methods of [30], and the projected Newton or scaled gradient projection methods [4, 22, 27].

Consistent with the unconstrained case, the gradient flow equations are [26]

$$\frac{du}{dt} = -F(u), \qquad u(0) = u_0, \tag{3.4}$$

where in this case, F is defined by (3.3). If we let $u_0 \in \Omega$, then the solution of (3.4) satisfies

$$\lim_{t\to\infty} F(u(t)) = 0. \tag{3.5}$$

We will also assume that (1.1) holds. Since dynamics [26] force $u(t) \in \Omega$ for all t and Ω is bounded, then if there are only finitely many solutions of $F(u) = 0$ in Ω, (1.1) will hold for all $u_0 \in \Omega$ in this case.

We may apply Theorem 2.1 directly, once we describe the maps $H(u)$ and show that (2.8) holds. We use known results [4, 22, 27] from optimization to do this.

One choice of $H(u)$ is the reduced Hessian. Letting $u \in \Omega$ and $0 \le \sigma < \min(U_i - L_i)/2$, we define the set of σ-binding constraints as

$$\begin{aligned}\mathcal{B}^\sigma(u) \quad = \{i \,|\quad & U_i - (u)_i \le \sigma \text{ and } (\nabla f(u))_i < -\sqrt{\sigma} \text{ or} \\ & (u)_i - L_i \le \sigma \text{ and } (\nabla f(u))_i > \sqrt{\sigma}\}.\end{aligned}$$

For $\mathcal{N} \subset \{1, 2, \ldots N\}$ we define $D(\mathcal{N})$ as the diagonal matrix with entries

$$D(\mathcal{N})_{ii} = \begin{cases} 1 & i \in \mathcal{N}, \\ 0 & i \notin \mathcal{N} \end{cases}$$

and define the reduced Hessian as

$$\bar{\mathcal{R}} f(u) = I - D(\mathcal{B}^0(u))(I - \nabla^2 f(u))D(\mathcal{B}^0(u)). \tag{3.6}$$

The second-order sufficiency conditions for a point u^* to be a local minimizer are [27] that $F(u^*) = 0$ and $\bar{\mathcal{R}} f(u^*)$ is positive definite. We will assume that u^* satisfies the second-order sufficiency conditions in what follows.

One must approximate the bounding constraints carefully in order to obtain a superlinearly convergent iteration. One way to do this [22, 27] is to let

$$H(u) = I - D(\mathcal{B}^\sigma(u)) \left(I - \nabla^2 f(u)\right) D(\mathcal{B}^\sigma(u)), \tag{3.7}$$

where $\sigma(u) = \|u - \mathcal{P}(u - \nabla f(u))\|$. With this choice the local iteration satisfies (2.8). In the case of a small residual bound-constrained nonlinear least squares problem, we may replace $\nabla^2 f(u)$ with the Gauss–Newton model Hessian.

4. Examples. In this section we present two examples. The first is a nonlinear equation on a manifold, which is a gradient flow of an inverse singular value problem. The projection is more subtle in this case, and we describe it in detail in section 4.1.1.

The second is a nonlinear bound-constrained least squares problem for which we compare the three variants of Ψtc (SER-A, SER-B, and TTE) with the trust-region method (`lmtr`) from [15], which in the nonlinear least squares case is a classic trust-region method [11, 22], and the damped Levenberg–Marquardt algorithm (`lmls`) from [22]. The projection in this case is trivial to compute, and we use the reduced Gauss–Newton model Hessian instead of $\nabla^2 f$ in (3.7). This example is a small artificial problem, which enables us to consider the cases where the minimizer is in the interior

of the feasible set, on the boundary (and hence degenerate in the sense that the binding constraints are a proper subset of the active constraints), and outside of the feasible set (and therefore a nonzero residual problem). In this example we do not increase δ unless f decreases, and we manage δ with the approach in section 2.1.1. All of the Ψtc methods, and especially SER-B and TTE, work much better if we do that.

In all of the examples, SER-B performs very well.

4.1. Inverse singular value problem. The inverse singular value problem [7] is to find $c \in R^N$ so that the $M \times N$ matrix

$$B(c) = B_0 + \sum_{k=1}^{N} c_k B_k$$

has prescribed singular values $\{\sigma_i\}_{i=1}^N$. This is one example of a wide class of inverse eigenvalue and singular value problems for which a dynamic formulation is useful [8].

One can assume without loss of generality that the matrices $\{B_i\}_{i=1}^N$ are orthonormal with respect to the Frobenius inner product and then formulate the problem as a constrained nonlinear least squares problem

$$\min \Psi(U,V) \equiv \|R(U,V)\|_F^2 \tag{4.1}$$

for $M \times M$ and $N \times N$ matrices U and V, subject to the constraint that U and V be orthogonal. If one finds a solution with a zero residual, then one has solved the original problem. This is not always possible, as the original problem may not have a solution [6]. In (4.1) the residual is

$$R(U,V) = U\Sigma V^T - B_0 - \sum_{k=1}^{N} \langle U\Sigma V^T, B_k\rangle B_k,$$

where $\langle\cdot,\cdot\rangle$ is the Frobenius inner product.

If we let Ω denote the manifold of pairs of $M\times M$ and $N\times N$ orthogonal matrices, then the projection of $\nabla\Psi$ onto the tangent space of Ω at $(U,V) \in \Omega$ is

$$g(U,V) = \frac{1}{2}\begin{pmatrix} \left(R(U,V)V\Sigma^T U^T - U\Sigma V^T R(U,V)^T\right) U \\ \left(R(U,V)^T U\Sigma V^T - V\Sigma^T U^T R(U,V)\right) V \end{pmatrix}.$$

The gradient flow equations for the problem are of the form (1.2) with

$$F(u) = g(U,V), \text{ where } u = \begin{pmatrix} U \\ V \end{pmatrix}. \tag{4.2}$$

Since $F(u)$ is in the tangent space of Ω at u [7], the solution of (1.2) is in Ω if $u_0 \in \Omega$. Since g is analytic in u, the results of [28] will apply, and so (1.1) holds for all initial vectors $u_0 \in \Omega$.

4.1.1. The projection onto Ω. The projection of an $N \times N$ matrix A onto the manifold of orthogonal matrices [17, 18] $A \to U_P$. Here $A = U_P H_P$, with U_P orthogonal and H_P symmetric positive semidefinite, is a polar decomposition of A. H_P is unique. U_P is unique if A is nonsingular. In this case (2.1) will hold. Since $S(\epsilon)$ is near a curve of orthogonal matrices, which have full rank, the possible singularity

of A is not an issue for us. One can compute U_P directly from the singular value decomposition $A = U\Sigma V^T$ of A as $U_P = UV^T$. This is efficient when A is small. For large A, there are several efficient iterative methods [16, 18].

So, in the context of this paper, given a pair of $N \times N$ matrices $w = (A, B)^T$,

$$\mathcal{P}(w) = \begin{pmatrix} U_P^A \\ U_P^B \end{pmatrix},$$

where U_P^A and U_P^B are the orthogonal parts of the polar decompositions of A and B.

4.1.2. Convergence of the local method. In this section we will verify that (2.8) holds. The local method uses the reduced Gauss–Newton model Hessian, which requires the projection $P_T(u)$ onto the tangent space at a point $u \in \Omega$. One can compute that projection by noting that if $w(t)$ is a differentiable orthogonal matrix-valued function, then differentiating $w^T(t)w(t) = I$,

$$\frac{dw(t)^T w(t)}{dt} = \dot{w}(t)^T w(t) + w(t)^T \dot{w}(t) = 0,$$

and hence $w^T\dot{w}$ is skew-symmetric. This implies that the tangent space for the manifold of orthogonal matrices at a point U is the space of matrices W for which U^TW is skew-symmetric. The projection onto the tangent space can then be computed as follows. If $\{S_i\}$ is a Frobenius-orthonormal basis for the skew-symmetric matrices, then a Frobenius-orthonormal basis for the tangent space is $\{US_i\}$, which we can use to compute the projection. If we do this for each component of $u = (U, V)^T$, we obtain $P_T(u)$. Alternatively we could use

$$P_T(u) = \mathcal{P}'(u) \text{ for all } u \in \Omega\ , \tag{4.3}$$

which follows from the fact that $\mathcal{P}$ is a map-to-nearest.

The local method for $F(u) = 0$ uses $H(u) = (I - P_T(u)) + P_T(u)F'(u)P_T(u)$, and we will show that $u_+^{NL} = u_c - H(u_c)^{-1}F(u_c)$ satisfies $\|e_+^{NL}\| = O(\|e_c\|^2)$, which implies (2.8) and will allow us to apply Theorem 2.1. We do this by noting that for all $u \in \Omega$

$$\begin{aligned} F(u) &= P_T(u)F(u) = P_T(u)F'(u)e + O\left(\|e\|^2\right) \\ &= H(u)e - (I - P_T(u))e + P_T(u)F'(u)(I - P_T(u))e + O\left(\|e\|^2\right) \\ &= H(u)e + O\left(\|(I - P_T(u))e\| + \|e\|^2\right). \end{aligned} \tag{4.4}$$

If $u \in \Omega$ is near u^*, then we can use (4.3) and the Lipschitz continuity of $\mathcal{P}$ to conclude that

$$u = \mathcal{P}(u) = \mathcal{P}(u^*) + \mathcal{P}'(u)e + O\left(\|e\|^2\right) = P_T(u)e + O\left(\|e\|^2\right),$$

and so $(I - P_T(u))e = O(\|e\|^2)$. Hence $u_+^{NL} - u^* = u_c - u^* - H(u_c)^{-1}F(u_c) = O(\|e_c\|^2)$.

4.1.3. Computations. We take the example from [7], having orthonormalized the matrices $\{B_j\}_{j=0}^4$ with classical Gram–Schmidt and the Frobenius inner product. In Figure 4.1 we compare the relative performance of the three time-step management strategies SER-A, SER-B, and TTE. While TTE does poorly, it is interesting to see that both versions of SER do well.

We did not reduce δ to respond to increases in $\|\Psi\|$ in this example, and the SER-A and SER-B iterations still converged well.

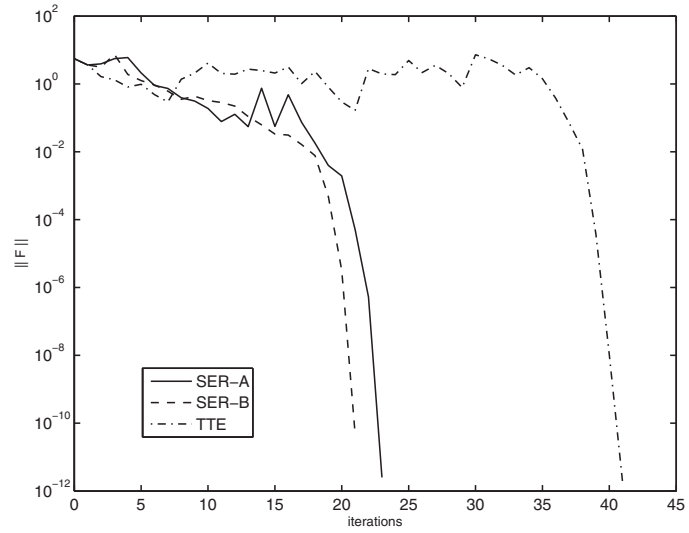

FIG. 4.1. *Inverse singular value problem.*

4.2. Inverse problem. This small ($N = 2$) example is taken from [3, 22]. We seek to identify the damping coefficient c and spring constant k for a simple harmonic oscillator. The governing differential equation is

$$w'' + cw' + kw = 0; \qquad w(0) = w_0, w'(0) = 0 \tag{4.5}$$

on the interval $[0, 1]$. We let $u = (c, k)^T$ and fit samples of the exact solution at 100 equally spaced points. We let $c = k = 1$ be the parameter values for the true solution and use `ode15s` from MATLAB to integrate (4.5) with the approximate parameters. The relative and absolute error tolerances were 10^{-6}.

The function to be minimized is

$$f(u) = \frac{1}{2} R(u)^T R(u) = \frac{1}{2} \sum_{i=1}^{100} (w^{exact}(t_i) - w_i(u))^2,$$

where $t_i = i/100$, $w_i(u)$ is the solution returned by `ode15s` with $u = (c, k)^T$, and w^{exact} is the solution of (4.5) with $(c, k) = (1, 1)$. The upper bounds are $[10, 10]$, and we consider three cases for the lower bounds: $[0, 0]$, placing the global minimizer in the interior of the feasible region, $[1, 0]$, placing the global minimizer on the boundary, and $[2, 0]$, placing the global minimizer outside. In the last case the solution of the unconstrained zero-residual problem does not satisfy the bound constraints. The residual at the optimal point for the constrained problem is 21.5.

The initial iterate in all cases was $(c, k) = (10, 10)$. In this computation we set $\delta_0 = 1/100$ and terminate the continuation when either the norm of the projected gradient $\|F\|$ has been reduced by a factor of 10^3 or $f < 10^{-6}$.

In Figure 4.2 we plot the values of f and $\|F\|$ as functions of the iteration count for several variations of Ψtc : SER-A (`ser-a`), SER-B (`ser-b`), TTE (`tte`), and the trust-region Levenberg–Marquardt method `lmtr` from [15]. We also compare them with the Levenberg–Marquardt line search method (`lmls`) from [22]. For SER-A, SER-B, and TTE, we rejected any step that increased the residual and decreased the time step by factors of 2 until either the residual decreased or $dt = 10^{-4}$. In the latter case we terminated the iteration.

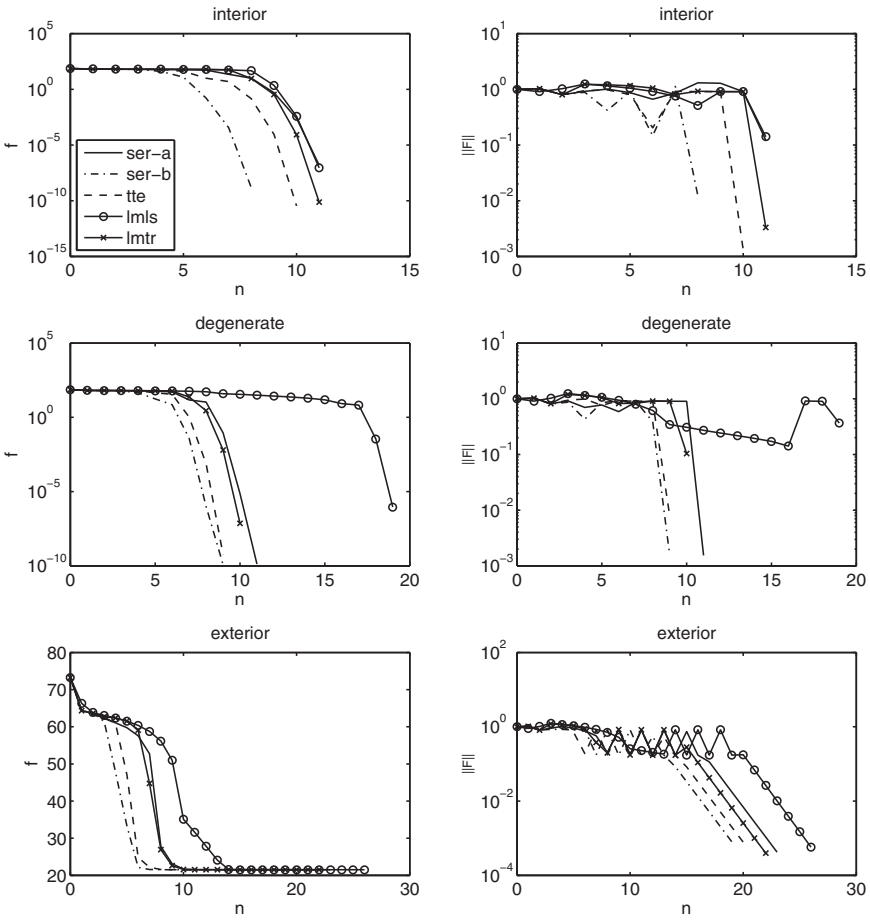

FIG. 4.2. *Parameter ID example.*

The damped Levenberg–Marquardt method from [22] is not as effective as the other four approaches, and SER-B is consistently better than the others.

5. Conclusions. We have described and analyzed a generalization of the pseudo-transient continuation algorithm which can be applied to a class of constrained nonlinear equations. The new approach can be applied to bound-constrained problems and to certain inverse eigenvalue and singular value problems.

We have reported on numerical testing which illustrates the performance of the method.

REFERENCES

[1] P.-A. ABSIL, C. G. BAKER, AND K. A. GALLIVAN, *Trust-region methods on Riemannian manifolds*, Found. Comput. Math., 7 (2007), pp. 303–330.

[2] U. M. ASCHER AND L. R. PETZOLD, *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM, Philadelphia, 1998.

[3] H. T. BANKS AND H. T. TRAN, *Mathematical and Experimental Modeling of Physical Processes*, Department of Mathematics, North Carolina State University, unpublished lecture notes for Mathematics, 1997, pp. 573–574.

[4] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.
[5] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.
[6] D. Chu and M. T. Chu, *Low rank update of singular values*, Math. Comp., 75 (2006), pp. 1351–1366.
[7] M. T. Chu, *Numerical methods for inverse singular value problems*, SIAM J. Numer. Anal., 29 (1992), pp. 885–903.
[8] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford Science, New York, 2005.
[9] T. Coffey, C. T. Kelley, and D. E. Keyes, *Pseudo-transient continuation and differential-algebraic equations*, SIAM J. Sci. Comput., 25 (2003), pp. 553–569.
[10] R. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[11] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics Appl. Math. 16, SIAM, Philadelphia, 1996.
[12] P. Deuflhard, *Adaptive Pseudo-Transient Continuation for Nonlinear Steady State Problems*, Technical report 02-14, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2002.
[13] K. R. Fowler and C. T. Kelley, *Pseudo-transient continuation for nonsmooth nonlinear equations*, SIAM J. Numer. Anal., 43 (2005), pp. 1385–1406.
[14] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration*, 2nd ed., Springer Ser. Comput. Math. 31, Springer-Verlag, Berlin, 2006.
[15] D. J. Higham, *Trust region algorithms and time step selection*, SIAM J. Numer. Anal., 37 (1999), pp. 194–210.
[16] N. J. Higham, *Computing the polar decomposition – with applications*, SIAM J. Sci. Comput., 7 (1986), pp. 1160–1174.
[17] N. J. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J. C. Glover and S. Barnett, eds., Oxford University Press, London, 1989, pp. 1–27.
[18] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1178–1192.
[19] H. Jiang and P. A. Forsyth, *Robust linear and nonlinear strategies for solution of the transonic Euler equations*, Comput & Fluids, 24 (1995), pp. 753–770.
[20] H. B. Keller, *Lectures on Numerical Methods in Bifurcation Theory*, Tata Institute of Fundamental Research, Lectures on Mathematics and Physics, Springer-Verlag, New York, 1987.
[21] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1987.
[22] C. T. Kelley, *Iterative Methods for Optimization*, Frontiers Appl. Math. 18, SIAM, Philadelphia, 1987.
[23] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Fundam. Algorithms 1, SIAM, Philadelphia, 1987.
[24] C. T. Kelley and D. E. Keyes, *Convergence analysis of pseudo-transient continuation*, SIAM J. Numer. Anal., 35 (1998), pp. 508–523.
[25] D. E. Keyes and M. D. Smooke, *A parallelized elliptic solver for reacting flows*, in Parallel Computations and Their Impact on Mechanics, A. K. Noor, ed., American Society of Mechanical Engineers, New York, 1987, pp. 375–402.
[26] L.-Z. Liao, H. Qi, and L. Qi, *Neurodynamical optimization*, J. Global Optim., 28 (2004), pp. 175–195.
[27] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.
[28] S. Łojasiewicz and M. A. Zurro, *On the gradient inequality*, Bull. Pol. Acad. Sci. Math., 47 (1999), pp. 143–145.
[29] W. Mulder and B. V. Leer, *Experiments with implicit upwind methods for the Euler equations*, J. Comput. Phys., 59 (1985), pp. 232–246.
[30] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Program., 58 (1993), pp. 353–367.
[31] L. T. Watson, S. C. Billups, and A. P. Morgan, *Algorithm 652: HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 13 (1987), pp. 281–310.