PROJECTED PSEUDO-TRANSIENT CONTINUATION *

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

Abstract. We propose and analyze a pseudo-transient 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. Pseudo-transient continuation, constrained dynamics, gradient flow, bound-constrained optimization, quasi-Newton method

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

1. Introduction. In this paper we extend algorithms and convergence results [15, 20, 26, 38, 48], for the method of pseudo-transient continuation (Ψtc) to a class of constrained problems in which projections onto the tangent space of the constraints are easy to compute. Such constraints arise in bound-constrained optimization [46] and inverse eigenvalue and singular value problems [13, 14]. 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 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 [4, 27, 70] where the time step is controlled with stability and accuracy in mind. Ψtc also differs from traditional continuation methods in that the objective is to find a steady-state solution, not, as is the case for pseudo-arclength continuation [21, 44, 67, 68], to track that solution as a function of another parameter. Homotopy methods [68, 77] 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.

Ψtc can resolve both ordinary differential equation (ODE) [48] and differential algebraic equation (DAE) [15, 26] dynamics, for both smooth and semismooth nonlinearities [26]. The method has been successfully applied to problems in computational fluid dynamics [16, 49, 62, 64, 65, 76], combustion [50, 74], plasma dynamics [51], radiation transport [72], and hydrology [23].

In the remainder of this section, we will review Ψtc for nonlinear equations and state a convergence result from [26] as an example of the theory. We will show how that result can be applied to unconstrained optimization in a way different from the trust-region approach in [38]. Our results for the constrained case, as we will show, also apply if the time step is managed as it was in [38].

In § 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 [26, 48], and includes more general assumptions on the iteration itself. We close section § 2 with some remarks on applications of the theory. In § 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 § 4 we apply the methods to two example problems.

---

*Version of July 29, 2007.

† North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA (Tim_Kelley@ncsu.edu,mtchu@ncsu.edu, jpreese@unity.ncsu.edu,cwwinton@ncsu.edu), The work of these authors has been partially supported by National Science Foundation Grants DMS-0404537 and DMS-0707220, and Army Research Office Grants DAAD19-01-1-0592, 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 has been 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 has been partially supported by the Research Grant Council of Hong Kong.

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

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

$$F(u) = 0 \tag{1.3}$$

with a globalized version of Newton's method [45, 47]. 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 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

$$u_+ = u_c - (\delta_c^{-1} I + F'(u_c))^{-1} F(u_c), \tag{1.4}$$

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 (1.4) becomes Newton's method. One common way to control δ is "Switched Evolution Relaxation" (SER) [62]. In SER the new time step is

$$\delta_+ = \min(\delta_c \|F(u_c)\|/\|F(u_+)\|, \delta_{max}) = \min(\delta_0 \|F(u_0)\|/\|F(u_+)\|, \delta_{max}). \tag{1.5}$$

Using $\delta_{max} = \infty$ is common. We will refer to update formula (1.5) as SER-A, to distinguish it from (1.6).

Our general 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 [43]. The formula for the time step is

$$\delta_+ = \max(\delta_c / \|u_+ - u_c\|, \delta_{max}). \tag{1.6}$$

In § 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) [50] estimates the local truncation error of $(u)_i(t_n)$ by

$$\tau \equiv \frac{\delta_n^2 (u)_i''(t_n)}{2},$$

approximates $(u)_i''$ by

$$\frac{2}{\delta_{n-1} + \delta_{n-2}} \left[\frac{((u)_i)_n - ((u)_i)_{n-1}}{\delta_{n-1}} - \frac{((u)_i)_{n-1} - ((u)_i)_{n-2}}{\delta_{n-2}} \right], \tag{1.7}$$

and computes δ_n by setting $\tau = 3/4$. In § 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

$$u^{n+1} = u^n - \delta_n F(u^{n+1}), \tag{1.8}$$

where u^n is the approximation to u at t_n and $\delta_n = t_{n+1} - t_n$ is the nth time step. (1.8) is a nonlinear equation for u^{n+1}, and software for (1.2) will use a Newton-like method to solve (1.8) to a tolerance fine enough for both stability and accuracy constraints to hold. If we express (1.8) as

$$G(u) = u - u^n + \delta_n F(u) = 0, \tag{1.9}$$

then an ODE code would solve (1.9) to high precision and accept the output of the nonlinear solver as u^{n+1}. If, on the other hand, we take a single Newton iteration for (1.9), with u^n as the initial iterate, we get

$$u_+ = u^n - G'(u^n)^{-1}G(u^n). \tag{1.10}$$

Since

$$G(u^n) = \delta_n F(u^n) \text{ and } G'(u^n) = I + \delta_n F'(u^n)$$

we get

$$u_+ = u^n - (I + \delta_n F'(u^n))^{-1}\delta_n F(u^n) = u^n - (\delta_n^{-1} I + F'(u^n))^{-1}F(u^n), \tag{1.11}$$

which is (1.4). Formula (1.4) is one of the Rosenbrock methods [27].

The convergence results in [15, 48] require that F be Lipschitz continuously differentiable, while those in [26] require semismoothness [61, 66]. In either case, one assumes that F is smooth enough for some form of Newton's method to have good local convergence properties.

We will state a special case of the result from [26] for smooth F to illustrate the ideas. We will use an inexact [17, 45, 47] formulation of Ψtc , where

$$u_+ = u_c + s, \tag{1.12}$$

and the step s satisfies the inexact Newton condition

$$\|(\delta_c^{-1} I + F'(u_c))s + F(u_c)\| \le \eta_c \|F(u_c)\|. \tag{1.13}$$

A common way to realize (1.13) is to use an iterative method to solve the equation

$$(\delta_c^{-1} I + F'(u_c))s = -F(u_c)$$

for the Ψtc step, and terminate the linear iteration when the relative residual is small.

The assumptions from [26] were

Assumption 1.1.

1. *F is Lipschitz continuously differentiable.*
2. *$F'(u^*)$ is nonsingular.*
3. *Equation* (1.1) *holds.*
4. *There are $\epsilon_H, \beta > 0$ such that $I + \delta F'(u)$ is nonsingular and*

$$\|(I + \delta F'(u))^{-1}\| \le (1 + \beta\delta)^{-1} \tag{1.14}$$

for all $\delta \ge 0$ and

$$u \in S(\epsilon_H) = \{z \mid \inf_{t \ge 0} \|z - u(t)\| \le \epsilon_H\}. \tag{1.15}$$

In the statement of Theorem 1.1, which is a special case of the main theorem in [26], we use the standard notation

$$e = u - u^*.$$

Theorem 1.1. *Let Assumption 1.1 hold. Assume that the sequence $\{\delta_n\}$ is updated with SER* (1.5)*. Then if δ_0 and the elements of the sequence $\{\eta_n\}$ are sufficiently small, the Ψtc sequence*

$$u_{n+1} = u_n - (\delta_n^{-1} I + F'(u_n))^{-1}F(u_n) \tag{1.16}$$

converges to u^*. Moreover there is $K > 0$ such that

$$\|e_{n+1}\| \le K\|e_n\|(\eta_n + \delta_n^{-1} + \|e_n\|). \tag{1.17}$$

Part 4 of Assumption 1.1 says that $F'(u)$ has no unstable eigenvalues near the trajectory $u(t)$. That is too strong for our purposes. We will significantly weaken Part 4 of Assumption 1.1 in § 2.

1.2. Ψtc for Optimization. In the simple case of unconstrained optimization,

$$\min_{u \in R^N} f(u), \tag{1.18}$$

one could consider applying Ψtc to the gradient flow equations

$$u' = -\nabla f(u), u(0) = u_0, \tag{1.19}$$

where $u' = du/dt$.

Assuming that f is smooth, then f decreases along the trajectory defined by (1.19), so numerical integration of (1.19) with sufficiently small time steps could be expected to drive ∇f to zero, which is what one expects, at least theoretically, of globally convergent gradient-based optimization methods [19, 25, 29, 46, 63]. This idea has been explored before, for example in [31, 32, 42, 53, 69, 75], with integration done by a variety of methods. One might expect that a variant of Ψtc would be more efficient than integrating (1.19) accurately with a stiff integrator, and that has been demonstrated in [59] for unconstrained problems.

Another approach to unconstrained optimization with Ψtc is the trust-region variant from [38], with the "time" variable playing the role of a Levenberg-Marquardt [19, 46, 52, 60] parameter. The assumption that (1.1) holds is not needed in the results of [38], instead one must adjust δ to reflect the quality of the quadratic model. The price paid for this weaker assumption is that one can only show that $\nabla f(u_n) \to 0$ unless a limit point satisfies the second order sufficiency conditions, which means (1.14) holds and one then obtains superlinear convergence.

Ψtc has also recently been used successfully for problems in optimal control [28, 34–37], and the results for DAE dynamics from [15, 26] apply to that work.

In the unconstrained case, where one wishes to find a local minimizer of a smooth $f : R^N \to R$, one could integrate (1.19). If $u(t)$ is a solution of (1.19), then

$$df(u(t))/dt = \nabla f(u(t))^T u'(t) = -\|\nabla f(u(t))\|^2 \le 0$$

and the inequality is strict unless $\nabla f(u(t)) = 0$ for some finite t. While this does not imply that $\lim_{t\to\infty} u(t)$ will exist, it does imply that any limit point of the set

$$\mathcal{U} = \{u(t) \,|\, t \ge 0\}$$

will satisfy the first order necessary conditions. If, in fact, a limit point u^* satisfies the second order sufficient conditions, then (1.1) will hold, and (1.14) will hold if u is sufficiently near u^*. Equation (1.1) can hold under weaker assumptions. For example it is easy to show [33, 53] that if there are only finitely many critical points and f has bounded level sets, then (1.1) holds for all choices of u_0, and the only stable equilibria are local minimizers.

If there are infinitely many critical points, the situation is more subtle. One result in this direction is that if $\mathcal{U}$ is bounded and f is real analytic, then (1.1) holds [11, 55, 56, 58, 73] for all u_0. More generally if the Lojasiewicz inequality holds at all critical points then (1.1) holds. The Lojasiewicz inequality [55] holds at a critical point u^* if there is a neighborhood U of u^* and a constant c such that

$$\|\nabla f(u)\| \ge c|f(u) - f(u^*)| \tag{1.20}$$

for all $u \in U$. Extensions to certain nonsmooth functions have also been made [8, 57]. One can also construct counterexamples of non-analytic C^∞ functions f for which (1.1) does not hold for certain values of u_0 [2, 3].

The Ψtc iteration for (1.19) is at first look the same as that for equations, with F replaced by ∇f.

$$u_+ = u_c - (\delta_c^{-1} I + H_c)^{-1} \nabla f(u_c), \tag{1.21}$$

where H_c is a model Hessian; [38] uses the exact Hessian $\nabla^2 f(u)$. The theory in [48] also requires positivity of the Hessian, whereas the new results in § 2 only require that the Hessian be positive definite near the solution. In particular, one could use $H_c = \nabla^2 f(u_c)$ even if it has negative eigenvalues, and there is no need to make δ small enough to force positivity of $\delta_c^{-1} I + H_c$, provided H_c is positive definite near the solution.

The theory gives little guidance into the choice of δ_0. In the case of optimization problems, one way to remedy a time step which is too large consistently with the theory is to reduce δ when the new point would increase f. As long as the number of such decreases is finite, the proofs of the results in [26] and this paper would continue to hold. Another such approach is the trust region method in [38], where δ is reduced if the reduction in the quadratic model of f is a poor predictor of the reduction in f itself. We discuss this issue in § 2.1.1.

2. Constrained Ψtc . Let F be Lipschitz continuous and assume that

$$u(t) \in \Omega \text{ for all } t \geq 0, \tag{2.1}$$

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

Assumption 2.1.

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

$$\|\mathcal{P}(v) - u\| \leq \|v - u\| + M_{\mathcal{P}} \|v - u\|^2. \tag{2.2}$$

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 (2.2).

We will consider a Ψtc iteration of the form

$$u_+ = \mathcal{P}\left[u_c - (\delta_c^{-1} I + H(u_c))^{-1} F(u_c)\right], \tag{2.3}$$

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

$$u_+ = u_c + s, \tag{2.4}$$

where

$$\|(\delta_c^{-1} I + H(u_c))s + F(u_c)\| \leq \eta_c \|F(u_c)\|. \tag{2.5}$$

Assumption 2.2, which replaces Part 4 in Assumption 1.1, weakens Assumption 1.1 in several respects. One new feature is the local nonlinear iteration, which is general enough to allow for quasi-Newton or Gauss-Newton methods. Note that the convergence rate for the local iteration is expressed in terms of the unprojected method

$$u_+^N = u_c - H(u_c)^{-1} F(u_c)$$

rather than the projected method

$$u_+^N = \mathcal{P}(u_c - H(u_c)^{-1}F(u_c)).$$

The reasons for this are that in many cases, bound-constrained optimization being one, the projected iteration is analyzed in terms of the unprojected one and the statement of the theorem and its proof are much simpler. The assumption on the unprojected iteration can be verified in the examples we consider in § 4.

ASSUMPTION 2.2.

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

$$\|H(u)\| \le M_H \text{ for all } u \in S(\epsilon_H). \tag{2.6}$$

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

$$\|F(u)\| > \bar{\epsilon}. \tag{2.7}$$

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

$$\|(I + \delta H(u_c))^{-1}\| \le (1 + \beta\delta)^{-1}, \text{ for some } \beta > 0 \text{ and all } \delta \ge 0, \tag{2.8}$$

and the Newton iteration

$$u_+^N = u_c - H(u_c)^{-1}F(u_c) \tag{2.9}$$

reduces the error by a (small) factor $r \in [0, 1)$ for all $u_c \in \Omega$, i. e.

$$\|e_+^N\| \le r\|e_c\|. \tag{2.10}$$

Theorem 2.1 extends Theorem 1.1 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 (2.9), so superlinear convergence is not required, and (2.8) need only hold in a neighborhood of u^*.

THEOREM 2.1. *Let F be locally Lipschitz continuous, 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 (1.5)*. Assume that there is $\delta^* > 0$ such that*

$$M_{\mathcal{P}}\epsilon_L/\beta < \delta^* \le \delta_n \tag{2.11}$$

for all n. Assume that the q-factor r in (2.10) *satisfies*

$$r < ((1 + M_{\mathcal{P}}\epsilon_L) - (1 + \beta\delta^*)^{-1})/2, \tag{2.12}$$

where β is the constant in (2.8)*. 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

$$\|e_{n+1}\| \le \|e_{n+1}^N\| + K\|e_n\|(\eta_n + \delta_n^{-1}), \tag{2.13}$$

where

$$\|e^N_{n+1}\| = \|(u_n - H(u_n)^{-1}F(u_n)) - u^*\|.$$

Proof.

We will prove the result for the exact ($\eta_n = 0$) iteration with $\delta_{max} = \infty$ in (1.5). The complete proof is based on the same ideas, but requires more bookkeeping. The outline of the proof follows those in [15,26,48].

We begin with the global phase. We wish to prove that while u is out of the local convergence region for the iteration (2.9), 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 1 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 (2.13).

Let $\epsilon < \epsilon_L$. 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

$$\|u_n - u^*\| < \epsilon \tag{2.14}$$

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

$$(\delta^{-1}I + H(u_n))^{-1} = \delta I + O(\delta^2), \tag{2.15}$$

and obtain an upper and lower bounds on δ_n/δ_0 while (2.14) fails to hold.

The estimate (2.15) follows from (2.6) if $\delta < 1/(2M_H)$. To obtain bounds for δ_n, we can apply the update formula (1.5) and (2.7) to show that while $u_n \in S(\epsilon_H)$ and $\|u_n - u^*\| > \epsilon/2$, then

$$\begin{aligned} \delta^* &\equiv \delta_0\|F(u_0)\|/\max_{u\in S(\epsilon_H)}\|F(u)\| \\ &\leq \delta_n = \delta_0\|F(u_0)\|/\|F(u_n)\| \leq 2\delta_0\|F(u_0)\|/C_F\epsilon. \end{aligned} \tag{2.16}$$

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

$$\delta^* \leq \delta_n \leq \frac{\|F(u_0)\|\delta_0}{\|F(u_n)\|} \leq \frac{2\delta_0\|F(u_0)\|}{C_F\epsilon}. \tag{2.17}$$

Hence for δ_0 sufficiently small, (2.15) holds if (2.14) does not.

With (2.15) in hand, we see that either (2.14) holds or

$$u_{n+1} = \mathcal{P}(u_n - \delta F(u_n)) + O(\delta^2), \tag{2.18}$$

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

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

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 [53]. To see this, we note that

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

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 [4, 27] 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

$$\|u_N - u^*\| \leq \epsilon$$

if

$$\delta_0 \leq \frac{C_F\epsilon^2}{4C_E\|F(u_0)\|}.$$

For the local phase, assume that (2.14) 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

$$\begin{aligned} v_{n+1} &= u_n - H(u_n)^{-1}F(u_n) + (H(u_n)^{-1} - (\delta_n^{-1}I + H(u_n))^{-1})F(u_n) \\ &= u_{n+1}^N + (I + \delta_n H(u_n))^{-1}H(u_n)^{-1}F(u_n) \\ &= u_{n+1}^N - (I + \delta_n H(u_n))^{-1}(u_{n+1}^N - u_n). \end{aligned}$$

So,

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

We use (2.2) to conclude that

$$\begin{aligned} e_{n+1} &= \mathcal{P}(v_{n+1}) - u^* \\ &= \mathcal{P}(u_{n+1}^N - (I + \delta_n H(u_n))^{-1}(e_{n+1}^N - e_n)) - u^* \\ &= \mathcal{P}(u_{n+1}^N - (I + \delta_n H(u_n))^{-1}(e_{n+1}^N - e_n)) - \mathcal{P}(u^*). \end{aligned} \tag{2.19}$$

So, since $\epsilon < \epsilon_L$,

$$\begin{aligned} \|e_{n+1}\| &\leq (1 + M_{\mathcal{P}}\epsilon_L)\left(\|e_{n+1}^N\| + \|(I + \delta_n H(u_n))^{-1}\|(\|e_{n+1}^N\| + \|e_n\|)\right) \\ &\leq (1 + M_{\mathcal{P}}\epsilon_L)(2r + (1 + \beta\delta^*)^{-1})\|e_n\|. \end{aligned} \tag{2.20}$$

This completes the proof since

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

by (2.11). Hence, the iteration converges at least locally q-linearly. Formula (1.5) then will imply (2.13). □

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 [15,26,48] 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 trajectory and that the step will grow at that time. The way this is done in the case of SER-A is to note that (1.5) 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^*, (2.8) drives u_n to u^* and δ_n to ∞.

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 proportional to 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 improve SER-B with this approach. The SER-B approach will work fine once u_n is near u^*, because then the local step will be small, driving δ_n to ∞. The proof of Theorem 2.1 breaks down in the global phase of the iteration, because then a small step can lead to a large value of δ too early in the iteration, resulting in inaccurate resolution of the dynamics.

However, if one requires a decrease in f before accepting the SER-B step (reducing δ, say by factors of 2) until a decrease is obtained, and only then updates δ with SER-B, then $\{f(u_n)\}$ will decrease and $F(u_n) = \nabla f(u_n) \to 0$. If u^* is the unique critical point of f in the set $\{u \,|\, f(u) \le f(u_0)\}$, then managing δ in this way will lead to convergence and (2.13) will hold. If u^* is not a unique critical point, then one must find another way to limit the increase in δ. One way to do this is to enforce a limit such as $\delta_n \le 2\delta_{n-1}$. Doing this allows one to select δ_0 to make u_n remain close to the trajectory.

Another approach is the trust-region method from [38], and the proof of Theorem 2.1 extends to that method. In the trust-region approach, δ is increased, decreased, or left unchanged as a function of the agreement between the reduction in the local quadratic model and the reduction (if any) in f. The step is rejected if there is an increase in f, and δ is reduced. If the model reduction and the actual reduction agree well, as they will in the case of an exact Hessian when near the solution, then δ is increased by a factor of two with each iteration, and superlinear convergence follows from (2.13), with $\eta_n = 0$ and $\delta_n = O(2^{-n})$. The limit on the increase in δ_n is key to keeping u_n near the solution arc.

Any of these approaches may allow one to use a larger value of δ_0 than a purely dynamic approach in which δ is only controlled by SER or TTE alone.

2.1.2. Semismooth Nonlinearities. If F is semismooth, which is the case considered in [26], and $H(u_n) \in \partial F(u_n)$, then the local iteration (2.9) is superlinearly convergent, if all matrices in $\partial F(u^*)$ are nonsingular. Hence we may recover the results from [26] from Theorem 2.1, the extension to DAE dynamics being the same as that in [26], and not relevant to this paper.

2.1.3. Modifications to the Jacobian. One could also use a continuous scaling matrix and solve the modified system

$$u' = -P(u)\nabla f(u),$$

where $P(u)$ is a symmetric positive definite matrix-valued function of u. This would, however, require an estimate for a Jacobian of $P(u)\nabla f(u)$ in (1.21).

Theorem 2.1 allows one to use a quasi-Newton model Hessian. One possibility for gradient flows is a Ψtc -BFGS method, where

$$H_{n+1} = H_n + \frac{y_n y_n^T}{y_n^T s_n} - \frac{(H_n s_n)(H_n s_n)^T}{s_n^T H_n s_n}, \tag{2.21}$$

where $s_n = u_{n+1} - u_n$ and $y_n = \nabla f(u_{n+1}) - \nabla f(u_n)$. The update (2.21) is the BFGS update [9, 24, 30, 71]. In the case of bound constraints, the update formula must be modified to account for active, binding, and inactive constraints (see § 3).

The theory in this paper does not completely include local convergence of Ψtc -quasi-Newton methods, because the local theory of such 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 [10] requires a line search and convex level sets, neither of which is a part of the Ψtc methods we propose.

A quasi-Newton approach that is covered by Theorem 2.1 would be to recompute the Hessian either periodically or when convergence becomes slow, and to either update it with the BFGS formula or leave it fixed if convergence is satisfactory. The nonlinear solver `nsold` from [47] uses a similar approach to blend Newton's method and the chord method. Skipping the update when $y^T s < 0$, for example, is also covered by Theorem 2.1.

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

$$H(u) = R'(u)^T R'(u) \tag{2.22}$$

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 (2.8) 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 (2.12) 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 [46]. One could also incorporate the structured quasi-Newton update for nonlinear least squares from [18, 19].

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

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

where

$$\Omega = \{u \,|\, L \le u \le U\}, \tag{3.2}$$

and the inequalities in (3.2) are component wise inequalities.

In order to describe necessary conditions and formulate the algorithms, we must recall some notation from [6, 7, 46].

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

$$\mathcal{P}(u)_i = \begin{cases} L_i & \text{if } (u)_i \le L_i \\ (u)_i & \text{if } L_i < (u)_i < U_i \\ U_i & \text{if } (u)_i \ge U_i \end{cases}. \tag{3.3}$$

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 [6, 46]

$$F(u) = u - \mathcal{P}(u - \nabla f(u)) = 0. \tag{3.4}$$

(3.4) is a semismooth nonlinear equation. Fast locally convergent methods include the semismooth Newton with the methods of [66] and the projected Newton or scaled gradient projection methods [6, 46].

Consistent with the unconstrained case, the gradient flow equations are [53],

$$\frac{du}{dt} = -F(u) \qquad u(0) = u_0, \tag{3.5}$$

where in this case, F is defined by (3.4). If we let $u_0 \in \Omega$, then the solution of (3.5) satisfies

$$\lim_{t\to\infty} F(u(t)) = 0. \tag{3.6}$$

We will also assume that (1.1) holds. Since dynamics [53] 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 (2.10) holds. For bound-constrained optimization, that is a bit subtle, and we describe one approach in § 3.1.

3.1. Reduced Hessian. One choice of $H(u)$ is the reduced Hessian, which we define in this section. We begin with the sufficient conditions for optimality. Our approach follows that of [6, 7, 46, 54]. Let $u \in \Omega$, we define the set of binding constraints as

$$\mathcal{B}(u) = \{i \,|\, (u)_i = L_i \text{ and } (\nabla f(u))_i < 0\} \cup \{i \,|\, (u)_i = U_i \text{ and } (\nabla f(u))_i > 0\}.$$

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

$$\bar{\mathcal{R}} f(u) = I - D(\mathcal{B}(u))(I - \nabla^2 f(u))D(\mathcal{B}(u)). \tag{3.7}$$

The second order sufficiency conditions for a point u^* to be a local minimizer are [54]

$$F(u^*) = 0, \tag{3.8}$$

and

$$\bar{\mathcal{R}} f(u^*) \text{ is positive definite.} \tag{3.9}$$

We will assume that u^* satisfies the second order sufficiency conditions in what follows.

A simple iteration of the form

$$u_+ = \mathcal{P}(u_c - \bar{\mathcal{R}} f(u_c)^{-1} \nabla f(u_c))$$

will not converge rapidly, because it can fail to identify the binding constraints. To remedy this, one must overestimate the binding set [7, 54]. For

$$0 \le \sigma < \min(U_i - L_i)/2,$$

define the σ-binding set

$$\begin{aligned} \mathcal{B}^\sigma(u) \quad = \{i \,|\, & \quad U_i - (u)_i \le \sigma \text{ and } (\nabla f(u))_i < -\sqrt{\sigma} \text{ or} \\ & \quad (u)_i - L_i \le \sigma \text{ and } (\nabla f(u))_i > \sqrt{\sigma}\}. \end{aligned} \tag{3.10}$$

Given a model Hessian $\mathcal{H}$ we define the σ-reduced Hessian at u by

$$H_\sigma(u) = I - D(\mathcal{B}^\sigma(u))(I - \mathcal{H})D(\mathcal{B}^\sigma(u)). \tag{3.11}$$

The scaled gradient projection iteration is

$$u_+ = \mathcal{P}(u_c - H_\sigma(u_c)^{-1} \nabla f(u_c)).$$

If σ is chosen carefully, then the scaled gradient projection iteration has the same global convergence properties as the original gradient projection method and there are superlinearly convergent implementations [7,46]. One way to do this is to make σ depend on the current iteration u_c as

$$\sigma(u_c) = \|u_c - \mathcal{P}(u_c - \nabla f(u_c))\|. \tag{3.12}$$

Theorem 3.1 is a local convergence result for the case $H_c = \nabla^2 f(u_c)$.

THEOREM 3.1. *Let u^* satisfy the second order sufficiency conditions and let $\{u_n\}$ be the projected Newton iterations with*

$$\mathcal{H} = \nabla^2 f$$

and $\sigma = \sigma(u_n)$ given by (3.12)*. Then if u_0 is sufficiently near to u^*, then $\mathcal{B}(u_n) = \mathcal{B}(u^*)$ for n sufficiently large and the iteration converges q-quadratically to u^*.*

Theorem 3.1 is simply the well-known local convergence result for Newton's method with the added feature that the binding constraints will be identified in finitely many steps. This latter feature is important, as the analysis is the same as for the unconstrained case after the active set has been identified (and hence (2.9) holds), and convergence proofs use this fact [7, 22, 46, 54].

Based on Theorem 3.1 we can use the reduced Hessian,

$$H(u) = I - D(\mathcal{B}^\sigma(u))(I - \nabla^2 f(u))D(\mathcal{B}^\sigma(u)). \tag{3.13}$$

In the case of a small residual bound-constrained nonlinear least squares problem, we can let $\mathcal{H}$ be the Gauss-Newton model Hessian.

One can also use quasi-Newton updates for the model Hessian. The implementation of the BFGS method in this context is similar to the approach from [46]. We modify y_n by using the approximate inactive set before applying the update. So we let

$$y_n^\# = (I - D(\mathcal{B}^\sigma(u_c)))y_n, \tag{3.14}$$

and the update is

$$H_{n+1} = H_n + \frac{y_n^\#(y_n^\#)^T}{(y_n^\#)^T s_n} - \frac{(H_n s_n)(H_n s_n)^T}{s_n^T H_n s_n}. \tag{3.15}$$

4. Examples. In this section we present two examples. The first 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 [38], which in the nonlinear least squares case is a classic trust region method [19, 46], and the damped Levenberg-Marquardt algorithm (`lmls`) from [46]. The projection in this case is trivial to compute, and we use the reduced Gauss-Newton model Hessian ((2.22) and (3.11)).

The first 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 non-zero residual problem). In this example we do not increase dt unless f decreases, and we manage dt with the approach in § 2.1.1. All of the PTC methods, and especially SER-B and TTE, work much better if we do that.

The second example 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 § 4.2.1.

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

4.1. Inverse Problem. This small ($N = 2$) example is taken from [5, 46]. We seek to identify the damping coefficient c and spring constant k for a simple harmonic oscillator. The governing differential equation is

$$w'' + cw' + kw = 0; w(0) = w_0, w'(0) = 0, \tag{4.1}$$

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 (4.1) with the approximate parameters. The relative and absolute error tolerances were 10^{-6}.

The function to be minimized is

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

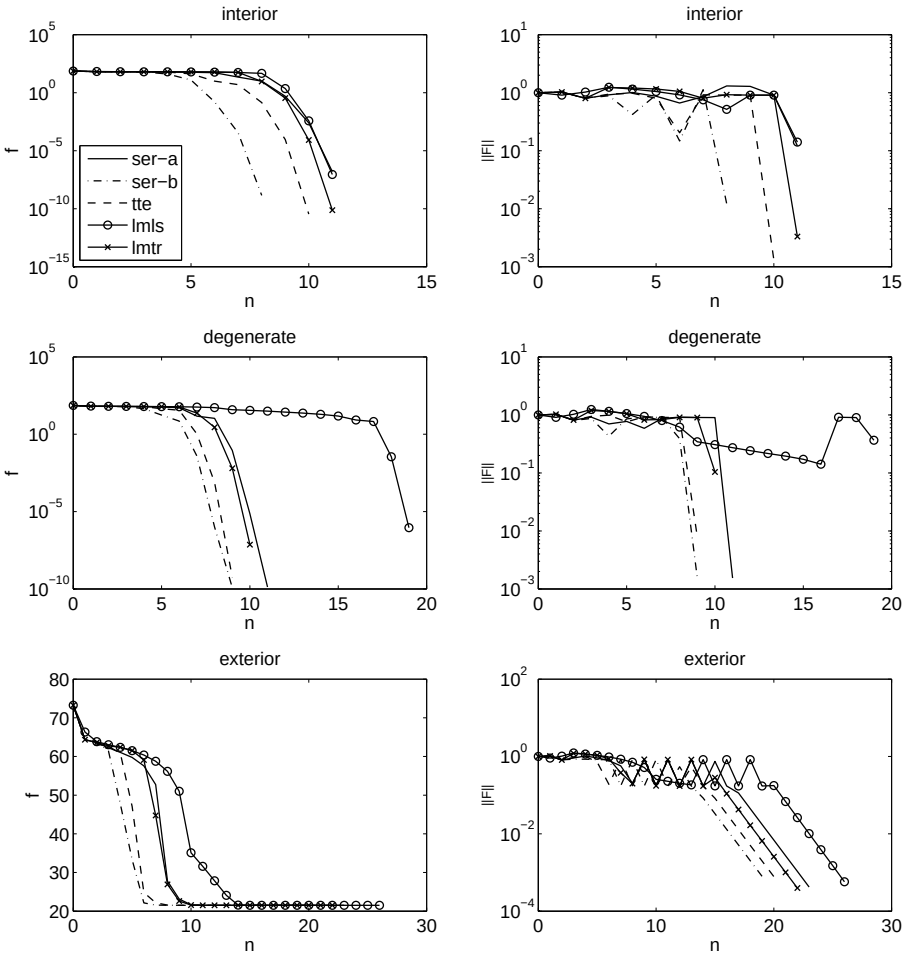
where $t_i = i/100$, $w_i(u)$ is the solution returned by `ode15s` with $u = (c, k)^T$, and w^{exact} is the solution of (4.1) with $(c, k) = (1, 1)$. The upper bounds are $[10, 10]$, and we consider three cases for the lower bounds: $[0, 0]$, placing the global minimizer in the interior of the feasible region, $[1, 0]$, placing the global minimizer on the boundary, and $[2, 0]$, with the global minimizer outside. In the latter 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 4.1 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 [38]. We also compare them with the Levenberg-Marquardt line search method (`lmls`) from [46]. 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.

The damped Levenberg-Marquardt method from [46] is not as effective as the other four approaches, and SER-B is consistently better than the others.

FIG. 4.1. *Parameter ID Example*



4.2. Inverse Singular Value Problem. The inverse singular value problem [13] 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 [14].

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,

$$\min \Psi(U,V) \equiv \|R(U,V)\|_F^2, \tag{4.2}$$

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 [12]. In (4.2) the residual is

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

where $< \cdot, \cdot >$ 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

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

Let

$$u = \begin{pmatrix} U \\ V \end{pmatrix}.$$

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

$$F(u) = g(U,V). \tag{4.3}$$

Since $F(u)$ is in the tangent space of Ω at u [13], the solution of (1.2) is in Ω if $u_0 \in \Omega$. Since g is analytic in u, the results of [11, 58, 73] will apply, and so (1.1) holds for all initial vectors $u_0 \in \Omega$.

4.2.1. The projection onto Ω. The projection of an $N \times N$ matrix A onto the manifold of orthogonal matrices [40, 41] is the map

$$A \to U_P.$$

Here $A = U_P H_P$, with U_P orthogonal and H_P symmetric positive semi-definite, is a polar decomposition of A. H_P is unique. U_P is unique if A is nonsingular. In this case (2.2) will hold. Since $S(\epsilon)$ is near to 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 [39, 41].

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

$$\mathcal{P}(w) = \begin{pmatrix} U_P^A \\ U_P^B \end{pmatrix}, \tag{4.4}$$

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

4.2.2. Convergence of the Local Method. In this section we will verify that (2.10) 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 value function, then differentiating

$$w^T(t)w(t) = I$$

implies

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

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

$$P_T(u) = \mathcal{P}'(u) \text{ for all } u \in \Omega \ , \tag{4.5}$$

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

$$u_+ = u_c - H(u_c)^{-1}F(u_c)$$

satisfies

$$\|e_+\| = O(\|e_c\|^2), \tag{4.6}$$

which will allow us to apply Theorem 2.1.

We verify (4.6) by noting that for all $u \in \Omega$

$$\begin{aligned} F(u) &= P_T(u)F(u) = P_T(u)F'(u)e + O(\|e\|^2) \\ &= H(u)e - (I - P_T(u))e + P_T(u)F'(u)(I - P_T(u))e + O(\|e\|^2) \qquad (4.7) \\ &= H(u)e + O(\|(I - P_T(u))e\| + \|e\|^2). \end{aligned}$$

If $u \in \Omega$ is near u^*, then we can use (4.5) and and Lipschitz continuity of $\mathcal{P}$ to conclude,

$$u = \mathcal{P}(u) = \mathcal{P}(u^*) + \mathcal{P}'(u)e + O(\|e\|^2) = P_T(u)e + O(\|e\|^2) \tag{4.8}$$

and so,

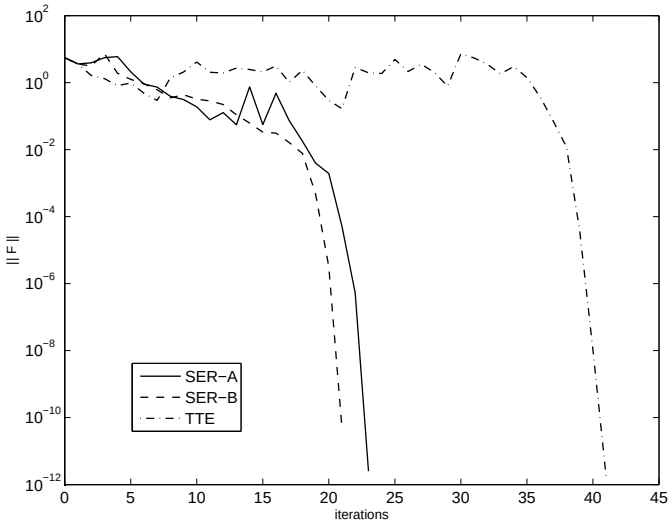
$$(I - P_T(u))e = O(\|e\|^2).$$

Hence

$$u_+ - u^* = u_c - u^* - H(u_c)^{-1}F(u_c) = O(\|e_c\|^2),$$

as asserted.

FIG. 4.2. *Inverse Singular Value Problem*



4.2.3. Computations. We take the example from [13], having orthonormalized the matrices $\{B_j\}_{j=0}^4$ with classical Gram-Schmidt and the Frobenius inner product. In Figure 4.2 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 dt to respond to increases in $\|\Psi\|$ in this example, and the SER-A and SER-B iterations still converged well.

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. Galivan, *Trust-region methods on Riemannian manifolds*, 2006. To appear in Found. Comp. Math.
[2] P.-A. Absil and K. Kurdyak, *On the stable equilibrium points of gradient systems*, 2006. To appear in Systems and Control Letters.
[3] P.-A. Absil, R. Mahony, and B. Andrews, *Convergence of the iterates of descent methods for analytic cost functions*, 2005.
[4] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM, Philadelphia, 1998.
[5] 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 573-4, 1997.
[6] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Autom. Control, 21 (1976), pp. 174–184.
[7] ———, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.
[8] J. Bolte, A. Daniilidis, and A. S. Lewis, *The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems*, 2007.
[9] C. G. Broyden, *A new double-rank minimization algorithm*, AMS Notices, 16 (1969), p. 670.
[10] 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.
[11] R. Chill, *On the łojasiewicz gradient inequality*, J. Funct. Anal., 201 (2003), pp. 572–601.
[12] D. Chu and M. T. Chu, *Low rank update of singular values*. To appear in Math. Comp., 2006.
[13] M. T. Chu, *Numerical methods for inverse singular value problems*, SIAM J. Numer. Anal., 29 (1992), pp. 885–903.
[14] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford Science Publishers, New York, 2005.

[15] T. Coffey, C. T. Kelley, and D. E. Keyes, *Pseudo-transient continuation and differential-algebraic equations*, SIAM J. Sci. Comp., 25 (2003), pp. 553–569.
[16] T. S. Coffey, R. J. McMullan, C. T. Kelley, and D. S. McRae, *Globally convergent algorithms for nonsmooth nonlinear equations in computational fluid dynamics*, J. Comp. Appl. Math., 152 (2003), pp. 69–81.
[17] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[18] J. E. Dennis, D. M. Gay, and R. E.Welsch, *An adaptive nonlinear least-squares algorithm*, ACM Trans. Math. Software, 7 (1981), pp. 348–368.
[19] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.
[20] P. Deuflhard, *Adaptive pseudo-transient continuation for nonlinear steady state problems*, Tech. Rep. 02-14, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, March 2002.
[21] E. J. Doedel and J. P. Kernévez, *AUTO: Software for continuation and bifurcation problems in ordinary differential equations*, tech. rep., California Institute of Technology, 1986.
[22] J. C. Dunn, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, SIAM J. Control Optim., 19 (1981), pp. 368–400.
[23] M. W. Farthing, C. E. Kees, T. Coffey, C. T. Kelley, and C. T. Miller, *Efficient steady-state solution techniques for variably saturated groundwater flow*, Advances in Water Resources, 26 (2003), pp. 833–849.
[24] R. Fletcher, *A new approach to variable metric methods*, Comput. J., 13 (1970), pp. 317–322.
[25] ———, *Practical methods of optimization*, John Wiley and Sons, New York, 1987.
[26] K. R. Fowler and C. T. Kelley, *Pseudo-transient continuation for nonsmooth nonlinear equations*, SIAM J. Numer. Anal., 43 (2005), pp. 1385–1406.
[27] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, 1971.
[28] I. Gherman and V. Schulz, *Preconditioning of one-shot pseudo-timestepping methods for shape optimization*, PAMM, 5 (2005), pp. 741–742.
[29] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.
[30] D. Goldfarb, *A family of variable metric methods derived by variational means*, Math. Comp., 24 (1970), pp. 23–26.
[31] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer Verlag, New York, 1987.
[32] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*, Springer Verlag, New York, 1991.
[33] Q. Han, L.-Z. Liao, H. Qi, and L. Qi, *Stability analysis of gradient-based neural networks for optimization problems*, J. Global Opt., 19 (2001), pp. 363–381.
[34] S. B. Hazra and V. Schulz, *Simultaneous pseudo-timestepping for PDE-model based optimization problems*, BIT, 44 (2004), pp. 457–472.
[35] ———, *Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with constraints*, SIAM J. Sci. Comput., 28 (2006), pp. 1078–1099.
[36] S. B. Hazra, V. Schulz, and J. Brezillon, *Simultaneous pseudo-time stepping for 3D aerodynamic shape optimization*, 2005. Preprint.
[37] S. B. Hazra, V. Schulz, J. Brezillon, and N. Gauger, *Aerodynamic shape optimization using simultaneous pseudo-timestepping*, J. Comp. Phys., 204 (2005), pp. 46–64.
[38] D. J. Higham, *Trust region algorithms and time step selection*, SIAM J. Numer. Anal., 37 (1999), pp. 194–210.
[39] N. J. Higham, *Computing the polar decomposition – with applications*, SIAM J. Sci. Comput., 7 (1986), pp. 1160–1174.
[40] ———, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J. C. Glover and S. Barnett, eds., Oxford University Press, 1989, pp. 1–27.
[41] 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.
[42] A. R. Humphries and A. M. Stuart, *Runge-Kutta methods for dissipative and gradient dynamical systems*, SIAM J. Numer. Anal., 31 (94), pp. 1452–1485.
[43] H. Jiang and P. A. Forsyth, *Robust linear and nonlinear strategies for solution of the transonic Euler equations*, Computers and Fluids, 24 (1995), pp. 753–770.
[44] 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.
[45] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.
[46] ———, *Iterative Methods for Optimization*, no. 18 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1999.
[47] ———, *Solving Nonlinear Equations with Newton's Method*, no. 1 in Fundamentals of Algorithms, SIAM, Philadelphia, 2003.
[48] C. T. Kelley and D. E. Keyes, *Convergence analysis of pseudo-transient continuation*, SIAM J. Numer. Anal., 35 (1998), pp. 508–523.
[49] D. E. Keyes, *Aerodynamic applications of Newton-Krylov-Schwarz solvers*, in Proc. of 14th International Conference on Num. Meths. in Fluid Dynamics, R. N. et al. eds, ed., New York, 1995, Springer, pp. 1–20.
[50] 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., ASME, 1987, pp. 375–402.
[51] D. Knoll and P. McHugh, *Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow*, SIAM J. Sci. Comput., 19 (1998), pp. 291–301.
[52] K. Levenberg, *A method for the solution of certain nonlinear problems in least squares*, Quart. Appl. Math., 4 (1944),

pp. 164–168.
[53] L.-Z. Liao, H. Qi, and L. Qi, *Neurodynamical optimization*, J. Global Opt., 28 (2004), pp. 175–195.
[54] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.
[55] S. Łojasiewicz, *Une properiété topologique des sous-ensembles analytiques réels*, in Les 'Equations aux Dérivées Partielles, Paris, 1963, Éditions du Centre Natinoal de la Recherche Scientifique, pp. 87–89.
[56] ———, *Sur les trajectories de gradient d'une function analytique*, in Seminari di Geometria 1982-1983 (lecture notes), Dipartemento di Matematica, Universita di Bologna, 1984, pp. 115–117.
[57] ———, *Sur la géométrie semi- et sous-analytique*, Ann. Inst. Fourier, 43 (1993), pp. 1575–1595.
[58] S. Łojasiewicz and M. A. Zurro, *On the gradient inequality*, Bull. Polish Acad. Sci. Math., 47 (1999), pp. 143–145.
[59] X.-L. Luo, C. T. Kelley, L.-Z. Liao, and H.-W. Tam, *Combining trust region techniques and Rosenbrock methods for gradient systems.* Submitted for publication, 2006.
[60] D. W. Marquardt, *An algorithm for least squares estimation of nonlinear parameters*, SIAM J., 11 (1963), pp. 431–441.
[61] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.
[62] W. Mulder and B. V. Leer, *Experiments with implicit upwind methods for the Euler equations*, J. Comp. Phys., 59 (1985), pp. 232–246.
[63] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
[64] P. D. Orkwis and D. S. McRae, *Newton's method solver for high-speed separated flowfields*, AIAA Journal, 30 (1992), pp. 78–85.
[65] ———, *Newton's method solver for the axisymmetric Navier-Stokes equations*, AIAA Journal, 30 (1992), pp. 1507–1514.
[66] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Mathematical Programming, 58 (1993), pp. 353–367.
[67] W. C. Rheinboldt, *Numerical Analysis of Parametrized Nonlinear Equations*, John Wiley and Sons, New York, 1986.
[68] A. G. Salinger, N. M. Bou-Rabee, R. P. Pawlowski, E. D. Wilkes, E. A. Burroughs, R. B. Lehoucq, and L. A. Romero, *LOCA 1.0 Library of Continuation Algorithms: Theory and Implementation Manual*, Technical Report SAND2002-0396, Sandia National Laboratory, March 2002.
[69] J. Schropp, *Using dynamical systems methods to solving minimization problems*, Appl Num Math, 18 (1995), pp. 93–113.
[70] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, New York, 1994.
[71] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comp., 24 (1970), pp. 647–657.
[72] A. I. Shestakov, J. L. Milovich, and A. Noy, *Solution of the nonlinear Poisson-Boltzmann equation using pseudo-transient continuation and the finite element method*, J. Colloid Interface Sci, 247 (2002), pp. 62–79.
[73] L. Simon, *Asymptotics for a class of nonlinear evolution equations, with applications to geometric problems*, Ann. Math., 118 (1983), pp. 525–571.
[74] M. D. Smooke, R. Mitchell, and D. Keyes, *Numerical solution of two-dimensional axisymmetric laminar diffusion flames*, Combust. Sci. and Tech., 67 (1989), pp. 85–122.
[75] A. M. Stuart and A. R. Humphries, *The essential stability of local error control for dynamical systems*, SIAM J. Numer. Anal., 32 (1995), pp. 1940–1971.
[76] V. Venkatakrishnan, *Newton solution of inviscid and viscous problems*, AIAA Journal, 27 (1989), pp. 885–891.
[77] 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.