SIAM J. NUMER. ANAL.
Vol. 37, No. 1, pp. 194–210

TRUST REGION ALGORITHMS AND TIMESTEP SELECTION*

DESMOND J. HIGHAM†

Abstract. Unconstrained optimization problems are closely related to systems of ordinary differential equations (ODEs) with gradient structure. In this work, we prove results that apply to both areas. We analyze the convergence properties of a trust region, or Levenberg–Marquardt, algorithm for optimization. The algorithm may also be regarded as a linearized implicit Euler method with adaptive timestep for gradient ODEs. From the optimization viewpoint, the algorithm is driven directly by the Levenberg–Marquardt parameter rather than the trust region radius. This approach is discussed, for example, in [R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, New York, 1987], but no convergence theory is developed. We give a rigorous error analysis for the algorithm, establishing global convergence and an unusual, extremely rapid, type of superlinear convergence. The precise form of superlinear convergence is exhibited—the ratio of successive displacements from the limit point is bounded above and below by geometrically decreasing sequences. We also show how an inexpensive change to the algorithm leads to quadratic convergence. From the ODE viewpoint, this work contributes to the theory of *gradient stability* by presenting an algorithm that reproduces the correct global dynamics and gives very rapid local convergence to a stable steady state.

Key words. global convergence, gradient system, Levenberg–Marquardt, quadratic convergence, steady state, superlinear convergence, unconstrained optimization

AMS subject classifications. 65L06, 65K10

PII. S0036142998335972

1. Introduction.

This work involves ideas from two areas of numerical analysis: optimization and the numerical solution of ODEs. We begin by pointing out a connection between the underlying mathematical problems.

Given a smooth function $f : \mathbb{R}^m \mapsto \mathbb{R}$, an algorithm for unconstrained optimization seeks a *local minimizer*, that is, a point $\mathbf{x}^\star$ such that $f(\mathbf{x}^\star) \leq f(\mathbf{x})$ for all $\mathbf{x}$ in some neighborhood of $\mathbf{x}^\star$. The following standard result gives necessary conditions and sufficient conditions for $\mathbf{x}^\star$ to be a local minimizer. Proofs may be found, for example, in [5, 6, 7].

THEOREM 1.1. *The conditions $\nabla f(\mathbf{x}^\star) = \mathbf{0}$ and $\nabla^2 f(\mathbf{x}^\star)$ positive semidefinite are necessary for $\mathbf{x}^\star$ to be a local minimizer, whilst the conditions $\nabla f(\mathbf{x}^\star) = \mathbf{0}$ and $\nabla^2 f(\mathbf{x}^\star)$ positive definite are sufficient.*

On the other hand, given a smooth function $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^m$ and $\mathbf{x}^{\text{init}} \in \mathbb{R}^m$, we may consider the ODE system

$$\mathbf{x}'(t) = \mathbf{F}(\mathbf{x}(t)), \quad t > 0, \qquad \mathbf{x}(0) = \mathbf{x}^{\text{init}}. \tag{1.1}$$

Now suppose that $\mathbf{F}$ in (1.1) has the form $\mathbf{F}(\mathbf{x}) \equiv -\nabla f(\mathbf{x})$. In this case the ODE (1.1) is said to have a *gradient* structure; see, for example, [21]. By the chain rule, we have

$$\frac{d}{dt} f(\mathbf{x}(t)) = \sum_{i=1}^{m} \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} = -\sum_{i=1}^{m} \left(\frac{\partial f}{\partial x_i} \right)^2 = -\|\nabla f(x(t))\|^2. \tag{1.2}$$

*Received by the editors March 20, 1998; accepted for publication (in revised form) February 9, 1999; published electronically November 17, 1999. This work was supported by the Engineering and Physical Sciences Research Council of the UK under grants GR/K80228 and GR/M42206.

http://www.siam.org/journals/sinum/37-1/33597.html

†Department of Mathematics, University of Strathclyde, Glasgow, G1 1XH, UK (djh@maths.strath.ac.uk).

From (1.2) we see that along any solution of the ODE, the quantity $f(\mathbf{x}(t))$ decreases in Euclidean norm as t increases. Moreover, it *strictly* decreases unless $\nabla f(x(t)) = \mathbf{0}$. Hence, solving the ODE up to a large value of t may be regarded as an attempt to compute a local minimum of f. The conditions given in Theorem 1.1 may now be interpreted as necessary conditions and sufficient conditions for $\mathbf{x}^\star$ to be a linearly stable fixed point of the ODE.

Several authors have noted the connection between optimization and gradient ODEs. Schropp [20] examined fixed timestep Runge–Kutta (RK) methods from a dynamical systems viewpoint, and found conditions under which the numerical solution of the gradient ODE converges to a stationary point of f. Schropp also gave numerical evidence to suggest that there are certain problem classes for which the ODE formulation is preferable to the optimization analogue. The book [12] shows that many problems expressible in optimization terms can also be written as ODEs, often with gradient structure. Chu has exploited this idea in order to obtain theoretical results and numerical methods for particular problems; see [3] for a review. In the optimization literature, the gradient ODE connection has also been mentioned; see, for example, the discussion on unconstrained optimization in [19]. Earlier work [1, 2] looked at the use of ODE methods to solve systems of nonlinear algebraic equations.

Related work by Kelley and Keyes [16] looked at implementations of the linearized implicit Euler method that are designed to give rapid convergence to steady state for general ODEs, with an emphasis on the case of semidiscretized partial differential equations. This approach has been widely used in the computational fluid dynamics community, and [16] developed a rigorous convergence theory. Since the class of problems considered in [16] is more general than the class of gradient systems, the results are necessarily weaker than those obtained here. A more detailed comparison of the results is given in subsection 4.2.

The algorithm analyzed in this work can be interpreted from the perspectives of optimization and timestepping. From the optimization viewpoint, the algorithm uses a trust region approach and is driven by the Levenberg–Marquardt parameter. This avoids the requirement of satisfying (exactly or approximately) a trust region radius constraint at each step. The algorithm is essentially that given by Fletcher [6, pp. 102–103], but to our knowledge, the convergence properties have not been analyzed before. We establish global convergence and provide sharp upper and lower bounds on the local convergence rate. An extremely fast type of superlinear convergence is identified—asymptotically k more bits of accuracy are obtained on the kth step. We also show that a minor modification leads to quadratic convergence.

From a timestepping viewpoint, this work adds to the literature on long-term dynamics for gradient systems [13, 15, 20, 22, 23]. The emphasis in this area has been placed on identifying methods that guarantee convergence to a fixed point (thus mimicking the global ODE dynamics). The gradient results in [15, 20] apply to fixed timestep Runge–Kutta methods and require the timestep to be sufficiently small. The analysis in [23] applies to very special classes of variable timestep methods and requires the gradient system to satisfy a one-sided Lipschitz condition. These results were extended to general adaptive Runge–Kutta pairs in [13], but the attractive feature that the local error tolerance could be chosen independently of the initial data was lost. None of the references [13, 15, 20, 22, 23] considers the rate at which convergence takes place. The adaptive algorithm analyzed in this work combines the desirable properties of (a) global convergence, that is, convergence to steady state independently of the initial data and the initial timestep, and (b) rapid local convergence.

The presentation is organized as follows. In the next section we introduce Newton's method and some simple numerical ODE methods. Section 3 is concerned with the trust region algorithm for unconstrained optimization. The algorithm is defined in section 3.1. A nonrigorous discussion of the convergence properties is given in section 3.2, and the main convergence theorems are proved in section 3.3. In section 4 we state the analogous results that hold when the algorithm is interpreted as an adaptive timestepping process for gradient ODEs. Subsection 4.2 discusses related behavior of general-purpose ODE methods.

2. Numerical methods. Most numerical methods for finding a local minimizer of f begin with an initial guess $\mathbf{x}_0$ and generate a sequence $\{\mathbf{x}_k\}$. Similarly, one-step methods for the ODE (1.1) produce a sequence $\{\mathbf{x}_k\}$ with $\mathbf{x}_k \approx \mathbf{x}(t_k)$. The time-levels $\{t_k\}$ are determined dynamically by means of the timestep $\Delta t_k := t_{k+1} - t_k$.

The steepest descent method for optimization has the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k), \tag{2.1}$$

where α_k is a scalar that may arise, for example, from a line search. This is equivalent to the explicit Euler method applied to the corresponding gradient ODE with timestep $\Delta t_k \equiv \alpha_k$. We note in passing that the poor performance of steepest descent in the presence of *steep-sided narrow valleys* is analogous to the poor performance of Euler's method on *stiff* problems. Indeed, Figure 4j in [7] and Figure 1.2 in [10] illustrate essentially the same behavior, viewed from these two different perspectives.

Newton's method for optimization is based on the local quadratic model

$$q_k(\boldsymbol{\delta}) := f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \boldsymbol{\delta} + \tfrac{1}{2} \boldsymbol{\delta}^T \nabla^2 f(\mathbf{x}_k) \boldsymbol{\delta}. \tag{2.2}$$

Note that $q_k(\boldsymbol{\delta})$ is the quadratic approximation to $f(\mathbf{x}_k + \boldsymbol{\delta})$ that arises from a Taylor series expansion about $\mathbf{x}_k$. If $\nabla^2 f(\mathbf{x}_k)$ is positive definite, then $q_k(\boldsymbol{\delta})$ has the unique minimizer $\boldsymbol{\delta}_k = -\left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \nabla f(\mathbf{x}_k)$. Thus we arrive at Newton's method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \nabla f(\mathbf{x}_k). \tag{2.3}$$

The following result concerning the local quadratic convergence of Newton's method may be found, for example, in [5, 6, 7].

THEOREM 2.1. *Suppose that $f \in C^2$ and that $\nabla^2 f$ satisfies a Lipschitz condition in a neighborhood of a local minimizer $\mathbf{x}^\star$. If $\mathbf{x}_0$ is sufficiently close to $\mathbf{x}^\star$ and if $\nabla^2 f(\mathbf{x}^\star)$ is positive definite, then Newton's method is well defined for all k and converges at second order.*

The implicit Euler method applied to (1.1) with $\mathbf{F}(\mathbf{x}) \equiv -\nabla f(\mathbf{x})$ using a timestep of Δt_k produces the equation

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \Delta t_k \nabla f(\mathbf{x}_{k+1}). \tag{2.4}$$

This is generally a nonlinear equation that must be solved for $\mathbf{x}_{k+1}$. Applying one iteration of Newton's method (that is, Newton's method for solving nonlinear equations) with initial guess $\mathbf{x}_{k+1} = \mathbf{x}_k$ gives

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \Delta t_k (I + \Delta t_k \nabla^2 f(\mathbf{x}_k))^{-1} \nabla f(\mathbf{x}_k). \tag{2.5}$$

This method is sometimes referred to as the linearized implicit Euler method; see, for example, [16, 24]. Note that for large values of Δt_k we have

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k - \left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \nabla f(\mathbf{x}_k), \tag{2.6}$$

and the ODE method looks like Newton's method (2.3). On the other hand, for small Δt_k we have

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k - \Delta t_k \nabla f(\mathbf{x}_k), \tag{2.7}$$

which corresponds to a small step in the direction of steepest descent (2.1). Hence, at the extremes of large and small Δt_k, the ODE method behaves like well-known optimization methods. However, we can show much more: for any value of Δt_k, the method (2.5) can be identified with a trust region process in optimization. This connection was pointed out by Goldfarb in the discussion on unconstrained optimization in [19]. The relevant optimization theory is developed in the next section.

3. A trust region algorithm.

3.1. The algorithm. We have seen that Newton's method is based on the idea of minimizing the local quadratic model $q_k(\boldsymbol{\delta})$ in (2.2) on each step. Since the model is valid only locally, it makes sense to restrict the increment, that is, to seek an increment $\boldsymbol{\delta}$ that minimizes $q_k(\boldsymbol{\delta})$ subject to some constraint $\|\boldsymbol{\delta}\| \le h_k$. Here h_k is a parameter that reflects how much trust we are prepared to place in the model.

Throughout this work we use $\|\cdot\|$ to denote the Euclidean vector norm and the corresponding induced matrix norm. In this case the locally constrained quadratic model problem is amenable to analysis. Lemma 3.1 below is one half of [6, Theorem 5.2.1]; a weaker version was proved in [8]. Lemma 3.2 is from [8]. For completeness, we give proofs of the lemmas here.

LEMMA 3.1. *Given symmetric $G \in \mathbb{R}^{m\times m}$ and $\mathbf{g} \in \mathbb{R}^m$, if, for some $\nu \ge 0$,*

$$(G + \nu I)\widehat{\boldsymbol{\delta}} = -\mathbf{g} \tag{3.1}$$

and $G + \nu I$ is positive semidefinite, then $\widehat{\boldsymbol{\delta}}$ is a solution of

$$\min_{\boldsymbol{\delta}} \mathbf{g}^T\boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T G\boldsymbol{\delta} \quad \textit{subject to} \quad \|\boldsymbol{\delta}\| \le \|\widehat{\boldsymbol{\delta}}\|. \tag{3.2}$$

Furthermore, if $G + \nu I$ is positive definite, then $\widehat{\boldsymbol{\delta}}$ is the unique solution of (3.2).

Proof. In the case where $G + \nu I$ is positive semidefinite, it is straightforward to show that $\widehat{\boldsymbol{\delta}}$ minimizes

$$\widehat{q}(\boldsymbol{\delta}) := g^T\boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T(G + \nu I)\boldsymbol{\delta}.$$

Hence, for all $\boldsymbol{\delta}$ we have $\widehat{q}(\boldsymbol{\delta}) \ge \widehat{q}(\widehat{\boldsymbol{\delta}})$; that is,

$$g^T\boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T G\boldsymbol{\delta} \ge g^T\widehat{\boldsymbol{\delta}} + \tfrac{1}{2}\widehat{\boldsymbol{\delta}}^T G\widehat{\boldsymbol{\delta}} + \tfrac{1}{2}\nu(\widehat{\boldsymbol{\delta}}^T\widehat{\boldsymbol{\delta}} - \boldsymbol{\delta}^T\boldsymbol{\delta}).$$

Thus $\widehat{\boldsymbol{\delta}}$ solves the problem (3.2). When $G + \nu I$ is positive definite, the inequality is strict for $\boldsymbol{\delta} \ne \widehat{\boldsymbol{\delta}}$, and hence the solution is unique. ∎

LEMMA 3.2. *Given symmetric $G \in \mathbb{R}^{m\times m}$ and $\mathbf{0} \ne \mathbf{g} \in \mathbb{R}^m$, suppose that $G + \nu I$ is positive definite for some $\nu \ge 0$. Then increasing ν in the linear system* (3.1) *causes $\|\widehat{\boldsymbol{\delta}}\|$ to decrease.*

Proof. Let the normalized eigenvectors of G form the columns of the orthogonal matrix Q and let $\{\lambda_i\}$ be the corresponding eigenvalues, so that $Q^T G Q = \operatorname{diag}(\lambda_i)$. From (3.1) we have

$$Q^T(G + \nu I)QQ^T\widehat{\boldsymbol{\delta}} = -Q^T\mathbf{g},$$

and hence

$$\|\widehat{\boldsymbol{\delta}}\| = \|\mathrm{diag}\left((\lambda_i + \nu)^{-1}\right) Q^T \mathbf{g}\|.$$

Since each $\lambda_i + \nu > 0$, the result follows. □

Note that Lemma 3.1 does not show how to compute an increment $\widehat{\boldsymbol{\delta}}$ given a trust region constraint $\|\boldsymbol{\delta}\| \leq h_k$. Such an increment may be computed or approximated using an iterative technique; see, for example, [6, pp. 103–107] or [5, pp. 131–143]. However, as mentioned in [6], it is reasonable to regard ν in (3.1) as a parameter that drives the algorithm—having chosen a value for ν and checked that $G + \nu I$ is positive definite, we may solve the linear system (3.1) and a posteriori obtain a trust region radius $h_k := \|\widehat{\boldsymbol{\delta}}\|$. Lemma 3.2 shows that $\|\widehat{\boldsymbol{\delta}}\|$ may be indirectly controlled through ν.

These remarks motivate Algorithm 3.3 below. We use $\lambda_{\min}(M)$ to denote the smallest eigenvalue of a symmetric matrix M and let $\epsilon > 0$ be a small constant. Given $\mathbf{x}_0$ and $\nu_0 > 0$ a general step of the trust region algorithm proceeds as follows.

ALGORITHM 3.3.

Compute $f_k := f(\mathbf{x}_k)$, $\mathbf{g}_k := \nabla f(\mathbf{x}_k)$ *and* $G_k := \nabla^2 f(\mathbf{x}_k)$
If $\lambda_{\min}(G_k + \nu_k I) \geq \epsilon$
    Solve $(G_k + \nu_k I)\boldsymbol{\delta}_k = -\mathbf{g}_k$
    Compute $\Delta f_k := f_k - f(\mathbf{x}_k + \boldsymbol{\delta}_k)$
    Compute $\Delta q_k := f_k - q_k(\boldsymbol{\delta}_k)$
    Compute $r_k := \Delta f_k / \Delta q_k$
    Set $\nu_{k+1} = V(r_k, \nu_k)$ *using* (3.3)
else
    set $r_k = -1$, $\nu_{k+1} = 2\nu_k$ *(and regard* $\boldsymbol{\delta}_k$ *as zero)*
end if
If $r_k \leq 0$
    set $\mathbf{x}_{k+1} = \mathbf{x}_k$
else
    set $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$
end if

The algorithm involves the function

$$V(r, \nu) = \begin{cases} 2\nu, & r < \frac{1}{4}, \\ \nu, & \frac{1}{4} \leq r \leq \frac{3}{4}, \\ \frac{1}{2}\nu, & \frac{3}{4} < r. \end{cases} \tag{3.3}$$

Note that r_k records the ratio of the reduction in f from $\mathbf{x}_k$ to $\mathbf{x}_k + \boldsymbol{\delta}_k$ and the reduction that is predicted by the local quadratic model. If r_k is significantly less than 1, then the model has been overoptimistic. This information is used in (3.3) to update the trust region parameter ν. In the case where the local quadratic model has performed poorly, we double the ν parameter, which corresponds to reducing the trust region radius on the next step. If the performance is reasonable, we retain the same value for ν. In the case of good performance we halve the value of ν, thereby indirectly increasing the trust region radius.

We emphasize that Algorithm 3.3 is a trust region algorithm in the sense that on each step $\boldsymbol{\delta}_k$ solves the local restricted problem

$$\min_{\boldsymbol{\delta}} {\mathbf{g}_k}^T \boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T G_k \boldsymbol{\delta} \quad \text{subject to} \quad \|\boldsymbol{\delta}\| \leq \|\boldsymbol{\delta}_k\|. \tag{3.4}$$

We also remark that the algorithm is essentially the same as that described in [6, pp. 102–103]. The underlying idea of adding a multiple of the identity matrix to ensure positive definiteness was first applied to the case where f has sum-of-squares form, leading to the Levenberg–Marquardt algorithm. Goldfeld, Quandt, and Trotter [8] extended the approach to a general objective function and gave some theoretical justification.

Theorems 5.1.1 and 5.1.2 of [6] provide a general convergence theory for a wide class of trust region methods. However, these results do not apply immediately to Algorithm 3.3, since the algorithm does not *directly* control the radius $h_k := \|\boldsymbol{\delta}_k\|$ but, rather, controls it indirectly via adaption of ν_k. In fact, we will see that the behavior established in Theorem 5.1.2 of [6], local quadratic convergence, does not hold for Algorithm 3.3. We are not aware of any existing convergence analysis that applies directly to Algorithm 3.3, except for general results of the form encapsulated in the Dennis–Moré characterization theorem for superlinear convergence [4, 5, 6] and the "strongly consistent approximation to the Hessian" theory given in [18]. These references are discussed further in the remarks that follow Theorem 3.5.

3.2. Motivation for the convergence analysis. The proofs in section 3.3 and the appendix are rather technical, and hence, to help orient the reader we give below a heuristic discussion of the key points.

Theorem 3.4 establishes global convergence, and the proof uses arguments that are standard in the optimization literature. Essentially, global convergence follows from the fact that when the local quadratic model is inaccurate the algorithm chooses a direction that is close to that of steepest descent. Perhaps of more interest is the rate of local convergence. Suppose that $\mathbf{x}_k \to \mathbf{x}^\infty$ as $k \to \infty$, with $\nabla f(\mathbf{x}^\infty) = 0$ and $\nabla^2 f(\mathbf{x}^\infty)$ positive definite, and suppose that for $k \geq \widehat{k}$ we have $r_k > 3/4$, and hence $\nu_{k+1} = \nu_k/2$. It follows that, for some constant C_1,

$$\nu_k = \frac{C_1}{2^k}, \quad k \geq \widehat{k}. \tag{3.5}$$

Note also that G_k and G_k^{-1} are bounded for large k.

Now, given a large k, let $\boldsymbol{\delta}_k^{\text{Newt}}$ denote the correction that would arise from Newton's method applied at $\mathbf{x}_k$, so that we have

$$(G_k + \nu_k I)\boldsymbol{\delta}_k = -\mathbf{g}_k, \tag{3.6}$$

$$G_k \boldsymbol{\delta}_k^{\text{Newt}} = -\mathbf{g}_k. \tag{3.7}$$

Expanding (3.6), using (3.7),

$$\boldsymbol{\delta}_k - \boldsymbol{\delta}_k^{\text{Newt}} = -\left[(G_k + \nu_k I)^{-1} - {G_k}^{-1}\right]\mathbf{g}_k = \nu_k {G_k}^{-2}\mathbf{g}_k + O(\|\mathbf{g}_k\|{\nu_k}^2). \tag{3.8}$$

Letting $\mathbf{d}_k := \mathbf{x}_k - \mathbf{x}^\infty$ and $e_k := \|\mathbf{d}_k\|$, we have $\mathbf{g}_k := \nabla f(\mathbf{x}_k) = \nabla f(\mathbf{x}^\infty + \mathbf{d}_k) = G_k\mathbf{d}_k + O(e_k^2)$. Hence, in (3.8)

$$\boldsymbol{\delta}_k - \boldsymbol{\delta}_k^{\text{Newt}} = \nu_k G_k^{-1}\mathbf{d}_k + O(\nu_k^2 e_k) + O(\nu_k e_k^2).$$

Using (3.5) we find that

$$\|\boldsymbol{\delta}_k - \boldsymbol{\delta}_k^{\text{Newt}}\| \leq \frac{C_2}{2^k}e_k + O(\nu_k^2 e_k) + O(\nu_k e_k^2) \tag{3.9}$$

for some constant C_2.

Now, since $\mathbf{x}_k + \boldsymbol{\delta}_k^{\text{Newt}}$ is the Newton step from $\mathbf{x}_k$, we have from Theorem 2.1

$$\|\mathbf{x}_k + \boldsymbol{\delta}_k^{\text{Newt}} - \mathbf{x}^\infty\| \leq C_3 e_k^2 \tag{3.10}$$

for some constant C_3. The triangle inequality gives

$$e_{k+1} \leq \|\mathbf{x}_k + \boldsymbol{\delta}_k - (\mathbf{x}_k + \boldsymbol{\delta}_k^{\text{Newt}})\| + \|\mathbf{x}_k + \boldsymbol{\delta}_k^{\text{Newt}} - \mathbf{x}^\infty\|,$$

and inserting (3.9) and (3.10) we arrive at the key inequality

$$e_{k+1} \leq \frac{C_4}{2^k} e_k + O(e_k^2) \tag{3.11}$$

for some constant C_4. The first term on the right-hand side of (3.11) distinguishes the algorithm from Newton's method and dominates the rate of convergence. To proceed, it is convenient to consider a shifted sequence; let $\widehat{e}_k := e_{k+N}$ for some fixed N to be determined. Then from (3.11),

$$\widehat{e}_{k+1} := e_{k+N+1} \leq \frac{C_4}{2^{k+N}} e_{k+N} + O(e_{k+N}^2) = \frac{C_4}{2^N} \frac{\widehat{e}_k}{2^k} + O(\widehat{e}_k^2). \tag{3.12}$$

Choosing N so that $2^N > C_4$, we have

$$\widehat{e}_{k+1} \leq \frac{1}{2^k} \widehat{e}_k + O(\widehat{e}_k^2). \tag{3.13}$$

Now, neglecting the $O(\widehat{e}_k^2)$ term in (3.13) leads to

$$\widehat{e}_j \leq \frac{\widehat{e}_0}{\prod_{i=0}^{j-1} 2^i} = \frac{\widehat{e}_0}{2^{j(j-1)/2}}. \tag{3.14}$$

If, in addition to ignoring the $O(\widehat{e}_k^2)$ term in (3.13), we also assume that equality holds, then we get equality in (3.14) and

$$\frac{\widehat{e}_{k+1}}{\widehat{e}_k^2} = \frac{\widehat{e}_0}{2^{(k+1)k/2}} \frac{2^{k(k-1)}}{\widehat{e}_0^2} = \frac{2^{k(k-3)/2}}{\widehat{e}_0} \to \infty \quad \text{as } k \to \infty, \tag{3.15}$$

but

$$\frac{\widehat{e}_{k+1}}{\widehat{e}_k} = \frac{\widehat{e}_0}{2^{(k+1)k/2}} \frac{2^{k(k-1)/2}}{\widehat{e}_0} = 2^{-k} \to 0 \quad \text{as } k \to \infty. \tag{3.16}$$

We see from (3.15) that the error sequence is not quadratically convergent. However, (3.16) corresponds to a very rapid form of superlinear convergence. Although this analysis used several simplifying assumptions, the main conclusions can be made rigorous, as we show in the next subsection. The type of superlinear convergence that we establish is likely to be as good as quadratic convergence in practice. This matter is discussed further after the proof of Theorem 3.5.

3.3. Convergence analysis of the trust region algorithm. The following theorem shows that Algorithm 3.3 satisfies a global convergence result. The structure of the proof is similar to that of [6, Theorem 5.1.1].

THEOREM 3.4. *Suppose that Algorithm* 3.3 *produces an infinite sequence such that* $\mathbf{x}_k \in B \subset \mathbb{R}^m$ *and* $\mathbf{g}_k \neq \mathbf{0}$ *for all* k*, where* B *is bounded and* $f \in C^2$ *on* B*. Then*

there is an accumulation point $\mathbf{x}^\infty$ *that satisfies the necessary conditions for a local minimizer in Theorem* 1.1.

Proof. Any sequence in B must have a convergent subsequence. Hence, we have $\mathbf{x}_k \to \mathbf{x}^\infty$ for $k \in \mathcal{S}$, where $\mathcal{S}$ collects the indices in the convergent subsequence. It is convenient to distinguish between two cases:

$$\text{(3.17)} \qquad \text{(i)} \ \sup_{k\in\mathcal{S}} \nu_k = \infty, \qquad \text{(ii)} \ \sup_{k\in\mathcal{S}} \nu_k \le W \quad \text{for some constant } W.$$

Case (i). From the form of $V(r,\nu)$ in (3.3), there must be an infinite subsequence whose indices form a set $\widehat{\mathcal{S}}$, where $\widehat{\mathcal{S}} \subseteq \mathcal{S}$, such that $r_k < \frac{1}{4}$. Also, using the boundedness of G_k and $\mathbf{g}_k$, we have

$$\|\boldsymbol{\delta}_k\| \le \|(G_k + \nu_k I)^{-1}\| \, \|\mathbf{g}_k\| = O(1/\nu_k),$$

and hence

$$\text{(3.18)} \qquad \|\boldsymbol{\delta}_k\| \to 0, \quad \text{as} \quad k \to \infty, \quad k \in \widehat{\mathcal{S}}.$$

Suppose that the gradient limit $\mathbf{g}^\infty := \nabla f(\mathbf{x}^\infty) \neq \mathbf{0}$. Then there exists a descent direction $\mathbf{s}$, normalized so that $\|\mathbf{s}\| = 1$, such that

$$\text{(3.19)} \qquad \mathbf{s}^T \mathbf{g}^\infty = -\alpha, \quad \alpha > 0.$$

Now, since $\boldsymbol{\delta}_k$ solves the local restricted subproblem (3.4), we have $q_k(\|\boldsymbol{\delta}_k\|\mathbf{s}) \ge q_k(\boldsymbol{\delta}_k)$ and so $\Delta q_k \ge q_k(\mathbf{0}) - q_k(\|\boldsymbol{\delta}_k\|\mathbf{s})$. Hence,

$$\text{(3.20)} \qquad \Delta q_k \ge -\|\boldsymbol{\delta}_k\|\mathbf{s}^T\mathbf{g}_k + o(\|\boldsymbol{\delta}_k\|) = \|\boldsymbol{\delta}_k\|\alpha + o(\|\boldsymbol{\delta}_k\|).$$

Also, a Taylor expansion of $f(\mathbf{x}_k + \boldsymbol{\delta}_k)$ about $\mathbf{x}_k$ gives

$$\text{(3.21)} \qquad \Delta f_k = \Delta q_k + o(\|\boldsymbol{\delta}_k\|^2).$$

We conclude from (3.18), (3.20), and (3.21) that $r_k = 1 + o(1)$ as $k \to \infty$ in $\widehat{\mathcal{S}}$, which contradicts $r_k < \frac{1}{4}$. Hence, $\mathbf{g}^\infty = \mathbf{0}$.

Now suppose that $G^\infty := G(\mathbf{x}^\infty)$ is not positive semidefinite; then there is a direction $\mathbf{v}$, with $\|\mathbf{v}\| = 1$, such that

$$\text{(3.22)} \qquad \mathbf{v}^T G^\infty \mathbf{v} = -\beta, \quad \beta > 0.$$

Pick $\widehat{k} \in \widehat{\mathcal{S}}$, and $\sigma = \pm 1$ so that $\sigma \mathbf{v}^T \mathbf{g}_k \le 0$ for all $k \in \widehat{\mathcal{S}}$ with $k \ge \widehat{k}$. Then, since $\boldsymbol{\delta}_k$ solves the local restricted subproblem (3.4), we have

$$\Delta q_k \ge q_k(\mathbf{0}) - q_k(\sigma\|\boldsymbol{\delta}_k\|\mathbf{v}) \ge -\tfrac{1}{2}\|\boldsymbol{\delta}_k\|^2 \mathbf{v}^T G_k \mathbf{v},$$

and hence

$$\text{(3.23)} \qquad \Delta q_k \ge \tfrac{1}{2}\|\boldsymbol{\delta}_k\|^2 \beta + o(\|\boldsymbol{\delta}_k\|^2).$$

It follows from (3.18), (3.21), and (3.23) that $r_k = 1 + o(1)$ as $k \to \infty$ in $\widehat{\mathcal{S}}$, which contradicts $r_k < \frac{1}{4}$. Hence, G^∞ is positive semidefinite.

Case (ii). From the form of $V(r,\nu)$ in (3.3), there must be an infinite subsequence whose indices form a set $\bar{\mathcal{S}}$, where $\bar{\mathcal{S}} \subseteq \mathcal{S}$, such that $r_k \ge \frac{1}{4}$ and $\lambda_{\min}(G_k + \nu_k I) \ge \epsilon$.

If $\mathbf{g}^\infty \neq \mathbf{0}$, then $\|\mathbf{g}_k\| \geq g_{\min}$ for some constant $g_{\min} > 0$ and for large $k \in \bar{\mathcal{S}}$, and hence

$$g_{\min} \leq \|\mathbf{g}_k\| \leq \|G_k + \nu_k I\| \, \|\boldsymbol{\delta}_k\| \leq (G_{\max} + W)\|\boldsymbol{\delta}_k\|,$$

where $G_{\max} := \sup_{\mathbf{x}\in B} \|\nabla^2 f(\mathbf{x})\|$. This gives

$$\|\boldsymbol{\delta}_k\| \geq \frac{g_{\min}}{G_{\max} + W} \quad \text{for large } k \in \bar{\mathcal{S}}.$$

Hence, removing the earlier indices from $\bar{\mathcal{S}}$ if necessary, we have with $h_k := \|\boldsymbol{\delta}_k\|$

$$\inf_{k\in\bar{\mathcal{S}}} h_k \geq h_{\min} := \frac{g_{\min}}{G_{\max} + W}. \tag{3.24}$$

Let $f^\infty := f(\mathbf{x}^\infty)$ and $\Delta f_k := f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq 0$. Since $f_1 - f^\infty \geq \sum_{k\in\bar{\mathcal{S}}} \Delta f_k$, we have $\Delta f_k \to 0$ as $k \to \infty$ in $\bar{\mathcal{S}}$. From $r_k \geq \frac{1}{4}$ it follows that $\Delta q_k \to 0$. Let $q^\infty(\boldsymbol{\delta}) := f^\infty + \boldsymbol{\delta}^T\mathbf{g}^\infty + \frac{1}{2}\boldsymbol{\delta}^T G^\infty \boldsymbol{\delta}$, choose $\bar{h} \in (0, h_{\min})$, let $\bar{\boldsymbol{\delta}}$ minimize $q^\infty(\boldsymbol{\delta})$ on $\|\boldsymbol{\delta}\| \leq \bar{h}$, and set $\bar{\mathbf{x}} := \mathbf{x}^\infty + \bar{\boldsymbol{\delta}}$. Then, for large $k \in \bar{\mathcal{S}}$,

$$\|\bar{\mathbf{x}} - \mathbf{x}_k\| \leq \|\bar{\boldsymbol{\delta}}\| + \|\mathbf{x}_k - \mathbf{x}^\infty\| = \|\bar{\boldsymbol{\delta}}\| + o(1) \leq \bar{h} + o(1) \leq h_k.$$

Hence, $\bar{\mathbf{x}} - \mathbf{x}_k$ is feasible on the subproblem that is solved by $\boldsymbol{\delta}_k$, and so

$$q_k(\bar{\mathbf{x}} - \mathbf{x}_k) \geq q_k(\boldsymbol{\delta}_k) = f_k - \Delta q_k. \tag{3.25}$$

Letting $k \to \infty$ in $\bar{\mathcal{S}}$, it follows from (3.25) that $q_k(\bar{\boldsymbol{\delta}}) \geq f^\infty = q^\infty(\mathbf{0})$. Thus $\boldsymbol{\delta} = \mathbf{0}$ also minimizes $q^\infty(\boldsymbol{\delta})$ on $\|\boldsymbol{\delta}\| \leq \bar{h}$, and since the constraint is inactive, the necessary conditions of Theorem 1.1 must be satisfied. Hence, $\mathbf{g}^\infty \neq \mathbf{0}$ is contradicted.

Now, with $\mathbf{g}^\infty = \mathbf{0}$ in Case (ii), we have

$$\|\boldsymbol{\delta}_k\| \leq \|(G_k + \nu_k I)^{-1}\| \, \|\mathbf{g}_k\| \leq \frac{1}{\epsilon}\|\mathbf{g}_k\| \to 0,$$

as $k \to \infty$ in $\bar{\mathcal{S}}$. Suppose G^∞ is not positive semidefinite. Then the arguments giving (3.22)–(3.23) may be applied and we conclude that $r_k = 1 + o(1)$ as $k \to \infty$ in $\bar{\mathcal{S}}$. It then follows from (3.3) that $\nu_k \to 0$, and since $\lambda_{\min}(G_k + \nu_k I) \geq \epsilon$ we must have G^∞ positive semidefinite. This gives the required contradiction. □

Note that, as mentioned in [6], since the algorithm computes a nonincreasing sequence f_k, the bounded region B required in this theorem will exist if any level set $\{\, \mathbf{x} : f(\mathbf{x}) \leq f_k \,\}$ is bounded.

In Theorem 3.4 we assume that $\mathbf{g}_k \neq \mathbf{0}$ for all k. If $\mathbf{g}_{\widehat{k}} = \mathbf{0}$ for some $\widehat{k}$, then the algorithm essentially terminates, giving $\mathbf{x}_k = \mathbf{x}_{\widehat{k}}$ and $\nabla f(\mathbf{x}_k) = \mathbf{0}$ for $k \geq \widehat{k}$. However, in this case we cannot conclude that $\nabla^2 f(\mathbf{x}_k)$ is positive semidefinite for $k \geq \widehat{k}$.

The next theorem quantifies the local convergence rate of Algorithm 3.3. The first part of the proof is based on that of [6, Theorem 5.1.2].

THEOREM 3.5. *If the accumulation point $\mathbf{x}^\infty$ of Theorem* 3.4 *also satisfies the sufficient conditions for a local minimizer in Theorem* 1.1, *then for the main sequence $\boldsymbol{\delta}_k \to 0$, $\nu_k \to 0$, and $r_k \to 1$. Further, the displacement error $e_k := \|\mathbf{x}_k - \mathbf{x}^\infty\|$ satisfies*

$$e_k \leq \frac{C}{2^{k^2/3}} \tag{3.26}$$

for some constant C, and if $e_k > 0$ for all k,

$$\frac{\widetilde{C}}{2^{k^2/2}} \leq \quad e_k \quad \leq \frac{C}{2^{k^2/3}}, \tag{3.27}$$

$$\frac{\bar{C}}{2^k} \leq \frac{e_{k+1}}{e_k} \leq \frac{\widehat{C}}{2^k} \tag{3.28}$$

for constants $\widetilde{C}, \bar{C} > 0$ and $\widehat{C}$, but the ratio e_{k+1}/e_k^2 is unbounded.

Proof. First, we show that Case (i) of (3.17) in the proof of Theorem 3.4 can be ruled out. Suppose that Case (i) arises. Then $r_k < \frac{1}{4}$, $\nu_k \to \infty$ and $\|\boldsymbol{\delta}_k\| \to 0$ as $k \to \infty$ in $\widehat{\mathcal{S}}$.

Since G^∞ is positive definite, the matrix G_k is also positive definite for large k in $\widehat{\mathcal{S}}$. In this case the Newton correction, $\boldsymbol{\delta}_k^{\text{Newt}}$, satisfying $G_k \boldsymbol{\delta}_k^{\text{Newt}} = -\mathbf{g}_k$, is well defined and gives a global minimum of the local quadratic model q_k. Define α by $\alpha\|\boldsymbol{\delta}_k^{\text{Newt}}\| = \|\boldsymbol{\delta}_k\|$ and note that since $\boldsymbol{\delta}_k$ solves the local restricted subproblem (3.4), we have $\alpha \leq 1$. Then

$$\begin{aligned} q_k(\alpha\boldsymbol{\delta}_k^{\text{Newt}}) &= f_k + \alpha{\boldsymbol{\delta}_k^{\text{Newt}}}^T \mathbf{g}_k + \tfrac{1}{2}\alpha^2 {\boldsymbol{\delta}_k^{\text{Newt}}}^T G_k \boldsymbol{\delta}_k^{\text{Newt}} \\ &= f_k + (\tfrac{1}{2}\alpha^2 - \alpha){\boldsymbol{\delta}_k^{\text{Newt}}}^T G_k \boldsymbol{\delta}_k^{\text{Newt}} \\ &\leq f_k - \tfrac{1}{2}\alpha^2 {\boldsymbol{\delta}_k^{\text{Newt}}}^T G_k \boldsymbol{\delta}_k^{\text{Newt}}. \end{aligned}$$

Hence, using $f_k = q_k(\mathbf{0})$,

$$\Delta q_k := q_k(\mathbf{0}) - q_k(\boldsymbol{\delta}_k) \geq q_k(\mathbf{0}) - q_k(\alpha\boldsymbol{\delta}_k^{\text{Newt}}) \geq \tfrac{1}{2}\alpha^2 {\boldsymbol{\delta}_k^{\text{Newt}}}^T G_k \boldsymbol{\delta}_k^{\text{Newt}} \geq \tfrac{1}{2}\alpha^2 \mu_{\min} \|\boldsymbol{\delta}_k^{\text{Newt}}\|^2,$$

where $\mu_{\min} > 0$ is a lower bound for the smallest eigenvalue of G_k for large k in $\widehat{\mathcal{S}}$. It follows that

$$\Delta q_k \geq \tfrac{1}{2}\mu_{\min}\|\boldsymbol{\delta}_k\|^2.$$

We may now conclude from (3.21) that $r_k \to 1$ as $k \to \infty$ in $\widehat{\mathcal{S}}$. Hence, Case (i) cannot arise.

For Case (ii), we have

$$\|\boldsymbol{\delta}_k\| \leq \|(G_k + \nu_k I)^{-1}\| \, \|\mathbf{g}_k\| \leq \frac{1}{\epsilon}\|\mathbf{g}_k\| \to 0, \tag{3.29}$$

as $k \to \infty$ with $k \in \bar{\mathcal{S}}$. Further, since $(G_k + \nu_k I)\boldsymbol{\delta}_k = -\mathbf{g}_k$,

$$\Delta q_k = -\boldsymbol{\delta}_k^T \mathbf{g}_k - \tfrac{1}{2}{\boldsymbol{\delta}_k}^T G_k \boldsymbol{\delta}_k = \tfrac{1}{2}{\boldsymbol{\delta}_k}^T G_k \boldsymbol{\delta}_k + \nu_k {\boldsymbol{\delta}_k}^T \boldsymbol{\delta}_k \geq \frac{\bar{\mu}_{\min}}{2}\|\boldsymbol{\delta}_k\|^2, \tag{3.30}$$

where $\bar{\mu}_{\min} > 0$ is a lower bound for the smallest eigenvalue of G_k for large k in $\widehat{\mathcal{S}}$. It follows from (3.21) that as $k \to \infty$ in $\bar{\mathcal{S}}$ we must have $r_k \to 1$, and hence $\nu_k \to 0$.

Having established that $\nu_k \to 0$, we now know that the correction used in the algorithm looks like the Newton correction $\boldsymbol{\delta}_k^{\text{Newt}}$, which satisfies $G_k \boldsymbol{\delta}_k^{\text{Newt}} = -\mathbf{g}_k$. Let $\mathbf{x}_{k+1}^{\text{Newt}} = \mathbf{x}_k + \boldsymbol{\delta}_k^{\text{Newt}}$. Also, let $\mathbf{d}_k := \mathbf{x}_k - \mathbf{x}^\infty$, so that $e_k = \|\mathbf{d}_k\|$, $e_k \to 0$ as $k \to \infty$ in $\bar{\mathcal{S}}$, and, by the triangle inequality,

$$e_{k+1} \leq \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{\text{Newt}}\| + \|\mathbf{x}_{k+1}^{\text{Newt}} - \mathbf{x}^\infty\|. \tag{3.31}$$

The quadratic convergence property of Newton's method given in Theorem 2.1 implies that for $\mathbf{x}_k$ sufficiently close to $\mathbf{x}^\infty$

$$\|\mathbf{x}_{k+1}^{\text{Newt}} - \mathbf{x}^\infty\| \leq A_1 e_k^2 \tag{3.32}$$

for some constant A_1.

Expanding the other term in (3.31), we find

$$\begin{aligned}\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{\text{Newt}} &= \boldsymbol{\delta}_k - \boldsymbol{\delta}_k^{\text{Newt}} \\ &= -\left[(G_k + \nu_k I)^{-1} - {G_k}^{-1}\right]\mathbf{g}_k \\ &= \nu_k {G_k}^{-2}\mathbf{g}_k + O(\|\mathbf{g}_k\|\nu_k^2).\end{aligned}$$

Since $\mathbf{g}_k = \nabla f(\mathbf{x}^\infty + \mathbf{d}_k) = \nabla^2 f(\mathbf{x}^\infty)\mathbf{d}_k + O(e_k^2) = G_k\mathbf{d}_k + O(e_k^2)$, we find that

$$\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{\text{Newt}} = \nu_k {G_k}^{-1}\mathbf{d}_k + O(\nu_k e_k^2) + O(\nu_k^2 e_k). \tag{3.33}$$

Using (3.32) and (3.33) in (3.31) gives, for large $k \in \bar{\mathcal{S}}$,

$$e_{k+1} \leq \frac{1}{\bar{\mu}_{\min}}\nu_k e_k + A_2 e_k^2 + O(\nu_k^2 e_k), \tag{3.34}$$

where A_2 is a constant.

Repeating the arguments that generated the inequalities (3.29) and (3.30), we can show that there is a neighborhood $\mathcal{N}$ around $\mathbf{x}^\infty$ such that if $\mathbf{x}_k, \mathbf{x}_{k+1} \in \mathcal{N}$ then $r_k \geq 3/4$, so that $\nu_{k+1} = \nu_k/2$. Hence, from (3.34), there is some $\bar{k} \in \bar{\mathcal{S}}$ for which $\mathbf{x}_{\bar{k}} \in \mathcal{N}$ and the main sequnce lies in $\mathcal{N}$ for $k \geq \bar{k}$. So in the main sequence we have $\mathbf{x}_k \to \mathbf{x}^\infty$, $\boldsymbol{\delta}_k \to 0$ and $r_k \to 1$ as $k \to \infty$, and $\nu_{k+1} = \nu_k/2$ for large k.

Hence (3.34) may be extended to the bound

$$e_{k+1} \leq A_3 \frac{1}{2^k} e_k + A_4 e_k^2 \quad \text{for all } k, \tag{3.35}$$

where A_3 and A_4 are constants. Lemma A.1 now gives (3.26).

To obtain a lower bound on e_{k+1} we use the triangle inequality in the form

$$e_{k+1} \geq \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{\text{Newt}}\| - \|\mathbf{x}_{k+1}^{\text{Newt}} - \mathbf{x}^\infty\|. \tag{3.36}$$

From (3.32) and (3.33) we have

$$e_{k+1} \geq \frac{A_5}{2^k} e_k - A_6 e_k^2, \tag{3.37}$$

for constants $A_5 > 0$ and A_6. Lemma A.1 gives the required result. □

We now list a number of remarks about Theorem 3.5.

1. The theorem shows that Algorithm 3.3 does not achieve a quadratic local convergence rate. This is caused by the fact that ν_k does not approach zero quickly enough. We have $\nu_k = O(2^{-k})$, which is reflected in the first term on the right-hand side of (3.35). A straightforward adaptation of the proof shows that by increasing the rate at which $\nu_k \to 0$, it is possible to make the second term on the right-hand side of (3.35) significant so that quadratic convergence is recovered. For example, this occurs if we alter the strategy for changing ν_k so that $\nu_{k+1} = \min(\nu_k/2, \nu_k^2)$ when $|r_k - 1| < .0001$ (and $\nu_{k+1} = V(r_k, \nu_k)$ in (3.3) otherwise). However, as explained in item 4 below, we would not expect this change to improve performance in practice. Quadratic convergence is also discussed in item 5 below.

2. The power $k^2/3$ appearing in (3.26) and (3.27) has been chosen partly on the basis of simplicity—it is clear from the proofs of Lemma A.1 and Theorem 3.5 that it can be replaced by ak^2 for any $a < 1/2$. (This will, of course, cause the constant C to change.)
3. It is also clear from the proof that the result is independent of the precise numerical values appearing in the algorithm. The values 1/4 and 3/4 in (3.3) can be replaced by any α and β, respectively, with $0 < \alpha < \beta < 1$ and the factor 2 in (3.3) can be replaced by any factor greater than unity. If the factor 1/2 in (3.3) is replaced by $1/K$, for $K > 1$, then the statement of the theorem remains true with powers of 2 replaced by powers of K. (The changes mentioned here will, of course, alter the constants C, $\widetilde{C}$, $\bar{C}$ and $\widehat{C}$.)
4. Theorem 3.5 shows that $e_{k+1}/e_k \to 0$, and hence the convergence rate is superlinear. However, the geometrically decreasing upper and lower bounds on e_{k+1}/e_k in (3.28) give us much more information. Asymptotically, while Newton's method gives twice as many bits of accuracy per step, the bound (3.28) corresponds to k more bits of accuracy on the kth step. In both cases, the asymptotic regime where e_k is small enough to make the convergence rate observable, but not so small that rounding errors are significant, is likely to consist of only a small number of steps.
5. Several authors have found conditions that are sufficient, or necessary and sufficient, for superlinear convergence of algorithms for optimization or rootfinding. The most comprehensive result of this form is the Dennis–Moré characterization theorem [4], [5, Theorem 8.2.4], and [6, Theorem 6.2.3]. Also, section 11.2 of [18] analyzes a class of rootfinding algorithms that employ "consistent approximations to the Hessian," and this approach may be used to establish superlinear convergence of Algorithm 3.3. However, these references, which cover general classes of algorithms, do not derive sharp upper and lower bounds on the *rate* of superlinear convergence of the type given in Theorem 3.5. In the terminology of [18, section 11.2], Algorithm 3.3 uses a strongly consistent approximation to the Hessian and superlinear convergence is implied by $\nu_k \to 0$. It also follows from [18, Result 11.2.7] that quadratic convergence arises if we ensure that $\nu_k \le C\|\mathbf{g}_k\|$ and convergence at R-order at least $(1+\sqrt{5})/2$ occurs if $\nu_k \le C\|\mathbf{x}_k - \mathbf{x}_{k-1}\|$ for some constant C.

Overall, Theorems 3.4 and 3.5 show that the algorithm has essentially the same basic properties as the trust region radius-driven alternative [6], without the requirement that an extra nonlinear equation be solved at each step.

4. Timestepping.

4.1. Gradient systems. If we identify the trust region parameter ν_k with the inverse of the timestep Δt_k, then the linearized implicit Euler method (2.5) is identical to the updating formula in Algorithm 3.3. Hence Algorithm 3.3 can be regarded as an adaptive linearized implicit Euler method for gradient ODEs, and the convergence analysis of section 3 applies. For completeness, we rewrite Algorithm 3.3 as a timestepping algorithm.

Given $\Delta t_0 > 0$ and $\mathbf{x}_0$ $(= \mathbf{x}^{\text{init}})$, a general step of the algorithm for the gradient system (1.1) with $\mathbf{F}(\mathbf{x}) \equiv -\nabla f(\mathbf{x})$ proceeds as follows.

ALGORITHM 4.1.

Compute $f_k := f(\mathbf{x}_k)$, $\mathbf{g}_k := \nabla f(\mathbf{x}_k)$ *and* $G_k := \nabla^2 f(\mathbf{x}_k)$
If $\lambda_{\min}(G_k + I/\Delta t_k) \geq \epsilon$
  Solve $(G_k + I/\Delta t_k)\boldsymbol{\delta}_k = -\mathbf{g}_k$
  Compute $\Delta f_k := f_k - f(\mathbf{x}_k + \boldsymbol{\delta}_k)$
  Compute $\Delta q_k := f_k - q_k(\boldsymbol{\delta}_k)$
  Compute $r_k := \Delta f_k / \Delta q_k$
  Set $\Delta t_{k+1} = W(r_k, \Delta t_k)$ *using* (4.1)
else
  set $r_k = -1$, $\Delta t_{k+1} = \Delta t_k/2$ *(and regard* $\boldsymbol{\delta}_k$ *as zero)*
end if
If $r_k \leq 0$
  set $\mathbf{x}_{k+1} = \mathbf{x}_k$
else
  set $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$
end if

The appropriate analogue of (3.3) is the function

$$W(r, \Delta t) = \begin{cases} \frac{1}{2}\Delta t, & r < \frac{1}{4}, \\ \Delta t, & \frac{1}{4} \leq r \leq \frac{3}{4}, \\ 2\Delta t, & \frac{3}{4} < r. \end{cases} \tag{4.1}$$

The following result is a restatement of Theorems 3.4 and 3.5 in this context.

THEOREM 4.2. *Suppose that Algorithm* 4.1 *for* (1.1) *with* $\mathbf{F}(\mathbf{x}) \equiv -\nabla f(\mathbf{x})$ *produces an infinite sequence such that* $\mathbf{x}_k \in B \subset \mathbb{R}^m$ *and* $\mathbf{g}_k \neq \mathbf{0}$ *for all* k, *where* B *is bounded and* $f \in C^2$ *on* B. *Then there is an accumulation point* $\mathbf{x}^\infty$ *that satisfies the necessary conditions for a local minimizer in Theorem* 1.1.

If the accumulation point $\mathbf{x}^\infty$ *also satisfies the sufficient conditions for a local minimizer in Theorem* 1.1, *then for the main sequence* $\boldsymbol{\delta}_k \to 0$, $\Delta t_k \to \infty$ *and* $r_k \to 1$. *Further, the displacement error* $e_k := \|\mathbf{x}_k - \mathbf{x}^\infty\|$ *satisfies*

$$e_k \leq \frac{C}{2^{k^2/3}} \tag{4.2}$$

for some constant C, *and if* $e_k > 0$ *for all* k,

$$\frac{\widetilde{C}}{2^{k^2/2}} \leq \ e_k \ \leq \frac{C}{2^{k^2/3}}, \tag{4.3}$$

$$\frac{\bar{C}}{2^k} \leq \frac{e_{k+1}}{e_k} \leq \frac{\widehat{C}}{2^k} \tag{4.4}$$

for constants $\widetilde{C}$, $\bar{C} > 0$, *and* $\widehat{C}$, *but the ratio* e_{k+1}/e_k^2 *is unbounded.*

4.2. General ODEs. We conclude with a discussion of Algorithm 4.1 in relation to general purpose adaptive timestepping algorithms.

The rule for changing timestep is different in spirit than the usual *local error control* philosophy for ODEs [9, 10]. This is to be expected since the aim of reaching equilibrium as quickly as possible is at odds with the aim of following a particular trajectory accurately in time. The timestep control policy in Algorithm 4.1 is based on a measurement of *closeness to linearity* of the ODE across the current timestep, rather than smallness of the local error. We also note that local error control algorithms

typically involve a user-supplied *tolerance* parameter, with the understanding that a smaller choice of tolerance produces a more accurate solution. Algorithm 4.1, on the other hand, involves fixed parameters.

The results in [13] showed that, for a special class of gradient systems, conventional local error control will eventually force the numerical solution to within $O(\tau)$ of a fixed point, where τ is the tolerance parameter. This is the best that can be expected in general, since Hall [11] showed that explicit Runge–Kutta pairs admit long term solutions that remain $O(\tau)$ away from a stable fixed point. Hence, timestep control based on the concept of local error is not the most efficient tool for capturing long term dynamics of gradient systems.

Kelley and Keyes [16] recently studied theoretical aspects of timestepping to steady state. They analyzed a family of adaptive algorithms based on linearized implicit Euler. As in Algorithm 4.1, the underlying idea in [16] is to increase the timestep where appropriate, in order to pick up the attractive convergence properties of Newton's method. Algorithms that are linearly, superlinearly, and quadratically convergent were identified in [16]. In the quadratically convergent case, the process switches to Newton's method when a preset timestep threshold is exceeded. Since [16] applies to general ODEs, the results are weaker than those for Algorithm 4.1, which is customized for gradient systems. In particular, the convergence result in [16] requires the initial timestep to be sufficiently small.

As a final point, we note that Algorithm 4.1 requires a check on the positive definiteness of the symmetric matrix $G_k + I/\Delta t_k$. This is an unusual requirement for a timestepping algorithm; however, an inexpensive and numerically stable test can be performed in the course of a Cholesky factorization [14, p. 225]. If this check is omitted from Algorithm 4.1, then the local convergence rate is unaffected, but the global convergence proof breaks down. Without an eigenvalue based test, there is a danger of convergence to an *unstable* fixed point. This can be regarded as a consequence of the fact that the implicit Euler method is overstable in the sense that the absolute stability region contains the infinite strip $\{z \in \mathbb{C} : \Re\{z\} > 1\}$ in the right-half of the complex plane; see, for example, [17, p. 229]. Another explanation is that Newton's method for optimizing f is identical to Newton's method for algebraic equations applied to $\nabla f = 0$; see, for example, [5, p. 100]. Hence, unless other measures are taken, there is no reason why stable fixed points should be preferred. In Algorithm 4.1 for gradient ODEs we check that $\lambda_{\min}(G_k + I/\Delta t_k) \geq \epsilon$ and $r_k > 0$, which helps to force the numerical solution to a stable fixed point. It is likely that traditional ODE error control would also direct the solution away from unstable fixed points, and hence the possibility of combining optimization and ODE ideas forms an attractive area for future work.

Appendix A. Convergence rate lemma.

LEMMA A.1. *Let $P, Q, T \geq 0$ and $R > 0$ be constants. Suppose $e_k \geq 0$ for all k, $e_k \to 0$ as $k \to \infty$, and*

$$e_{k+1} \leq \frac{P}{2^k} e_k + Q e_k^2 \quad \textit{for all } k. \tag{A.1}$$

Then

$$e_k \leq \frac{C}{2^{k^2/3}} \quad \textit{for some constant } C. \tag{A.2}$$

Further, if $e_k > 0$ for all k, then

$$\frac{e_{k+1}}{e_k} \leq \frac{\widehat{C}}{2^k} \quad \textit{for some constant } \widehat{C}, \tag{A.3}$$

and if, in addition,

$$e_{k+1} \geq \frac{R}{2^k} e_k - T e_k^2 \quad \textit{for all } k, \tag{A.4}$$

then

$$e_k \geq \frac{\widetilde{C}}{2^{k^2/2}} \quad \textit{and} \quad \frac{e_{k+1}}{e_k} \geq \frac{\bar{C}}{2^k} \quad \textit{for constants } \widetilde{C}, \bar{C} > 0, \tag{A.5}$$

but the ratio e_{k+1}/e_k^2 is unbounded.

Proof. Choose $C > 0$ such that

$$\tfrac{1}{2} + CQ \leq 1. \tag{A.6}$$

We first prove a result under restricted circumstances and then generalize to the full result. We assume that

$$P \leq \tfrac{1}{2} \quad \text{and} \quad e_i \leq \frac{C}{8}, \quad i = 0, 1, 2, 3. \tag{A.7}$$

Our induction hypothesis is

$$e_i \leq \frac{C}{2^{i(i-1)/2}}. \tag{A.8}$$

Note that, from (A.7), this holds for $i = 0, 1, 2, 3$. If (A.8) is true for $i = k \geq 3$, then, using (A.1),

$$e_{k+1} \leq \frac{P}{2^k} \frac{C}{2^{k(k-1)/2}} + Q \frac{C^2}{2^{k(k-1)}} \leq \frac{PC}{2^{k(k+1)/2}} + Q \frac{C^2}{2^{k(k+1)/2}}$$

(since $k(k+1)/2 \leq k(k-1)$ for $k \geq 3$). Hence, using (A.6) and (A.7),

$$e_{k+1} \leq \frac{C}{2^{k(k+1)/2}} (P + CQ) \leq \frac{C}{2^{k(k+1)/2}}.$$

Therefore, by induction, (A.8) is true for all k, if (A.7) holds.

Now, consider the shifted sequence $\widehat{e}_k := e_{k+N}$, for some fixed N. We have

$$\widehat{e}_{k+1} := e_{k+N+1} \leq \frac{P}{2^{k+N}} e_{k+N} + Q e_{k+N}^2 = \frac{P}{2^N} \frac{1}{2^k} \widehat{e}_k + Q \widehat{e}_k^2. \tag{A.9}$$

Since $e_k \to 0$ as $k \to \infty$, it is possible to choose N such that

$$\frac{P}{2^N} \leq \tfrac{1}{2} \quad \text{and} \quad \widehat{e}_i := e_{i+N} \leq \frac{C}{8}, \quad i = 0, 1, 2, 3. \tag{A.10}$$

From (A.9) and (A.10), the result (A.8) holds for this shifted sequence, so

$$\widehat{e}_k \leq \frac{C}{2^{k(k-1)/2}} \quad \text{for all } k. \tag{A.11}$$

Translating this into a result for the original sequence, we find that

$$e_k =: \widehat{e}_{k-N} \leq \frac{C}{2^{(k-N)(k-N-1)/2}} = \frac{C}{2^{N^2+N}} \frac{1}{2^{k^2/2-k(2N+1)/2}} \quad \text{for } k \geq N.$$

Relabelling C as $C/2^{N^2+N}$ and letting $\widehat{N} = (2N+1)/2$, we have

$$e_k \leq \frac{C}{2^{k^2/2-\widehat{N}k}} \quad \text{for } k \geq N.$$

Now $k^2/2 - \widehat{N}k \geq k^2/3$ for $k \geq 6\widehat{N}$. Hence,

$$e_k \leq \frac{C}{2^{k^2/3}} \quad \text{for } k \geq 6\widehat{N}.$$

Clearly, by increasing C, if necessary, the result will also hold for the finite sequence $e_0, e_2, \ldots, e_{6\widehat{N}}$. Hence, (A.2) is proved. The inequality (A.3) follows after dividing by e_k in (A.1) and using (A.2).

Now, (A.4) gives

$$e_{k+1} \geq \frac{e_k}{2^k}(R - 2^k T e_k).$$

From (A.2), for sufficiently large k we have $2^k T e_k \leq R/2$, so that

$$e_{k+1} \geq \frac{e_k}{2^k}\frac{R}{2} =: \frac{e_k}{2^k}\bar{C}. \tag{A.12}$$

Clearly, by reducing $\bar{C}$, if necessary, this result must hold for all k. Now, reduce $\bar{C}$, if necessary, so that $0 < \bar{C} < 1$. From (A.12),

$$e_k \geq \frac{1}{2^{k-1}} e_{k-1} \geq \frac{1}{2^{k-1}}\frac{1}{2^{k-2}} e_{k-2} \geq \cdots \geq \frac{1}{2^{k(k-1)/2}} e_0.$$

So letting $\widetilde{C} = e_0$ we have

$$e_k \geq \frac{\widetilde{C}}{2^{k(k-1)/2}} \geq \frac{\widetilde{C}}{2^{k^2/2}}. \tag{A.13}$$

Inequalities (A.12) and (A.13) give (A.5), as required.

Finally, using (A.2) and (A.5) we find that

$$\frac{e_{k+1}}{e_k^2} \geq \frac{\widetilde{C}}{C^2} 2^{(k^2-6k-3)/6} \to \infty \quad \text{as } k \to \infty. \qquad \square$$

Appendix B. This work has benefited from my conversations with a number of optimizers and timesteppers, most notably Roger Fletcher and David Griffiths. I also thank the editor and referee for useful feedback and the editor for pointing out reference [16].

REFERENCES

[1] J. P. Abbot and R. P. Brent, *Fast local convergence with single and multistep methods for nonlinear equations*, J. Austral. Math. Soc. Ser. B, 19 (1975), pp. 173–199.

[2] P. T. Boggs, *The solution of nonlinear systems of equations by A-stable integration techniques*, SIAM J. Numer. Anal., 8 (1971), pp. 767–785.
[3] M. T. Chu, *A list of matrix flows with applications*, in Hamiltonian and Gradient Flows, Algorithms and Control, A. Bloch, ed., Fields Insti. Commun. 3, AMS, Providence, RI, 1994, pp. 87–97.
[4] J. E. Dennis, Jr. and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.
[5] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.
[6] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, 1987.
[7] P. E. Gill, W. M. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.
[8] S. M. Goldfeld, R. E. Quandt, and H. F. Trotter, *Maximisation by quadratic hill-climbing*, Econometrica, 34 (1966), pp. 541–551.
[9] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I, *Non-stiff Problems*, 2nd ed., Springer-Verlag, Berlin, 1993.
[10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II, *Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin, 1996.
[11] G. Hall, *Equilibrium states of Runge–Kutta schemes*, ACM Trans. Math. Software, 11 (1985), pp. 289–301.
[12] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Springer-Verlag, Berlin, 1994.
[13] D. J. Higham and A. M. Stuart, *Analysis of the dynamics of local error control via a piecewise continuous residual*, BIT, 38 (1998), pp. 44–57.
[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
[15] A. R. Humphries and A. M. Stuart, *Runge–Kutta methods for dissipative and gradient dynamical systems*, SIAM J. Numer. Anal., 31 (1994), pp. 1452–1485.
[16] C. T. Kelley and D. E. Keyes, *Convergence analysis of pseudo-transient continuation*, SIAM J. Numer. Anal., 35 (1998), pp. 508–523.
[17] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley, Chichester, 1991.
[18] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[19] M. J. D. Powell, ed., *Nonlinear Optimization* 1981: *Proceedings of the NATO Advanced Research Institute*, Academic Press, New York, 1982.
[20] J. Schropp, *Using dynamical systems methods to solve minimization problems*, Appl. Numer. Math., 18 (1995), pp. 321–335.
[21] S. H. Strogatz, *Nonlinear Dynamics and Chaos*, Addison–Wesley, Reading, MA, 1994.
[22] A. M. Stuart and A. R. Humphries, *Model problems in numerical stability theory for initial value problems*, SIAM Rev., 36 (1994), pp. 226–257.
[23] 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.
[24] H. C. Yee and P. K. Sweby, *Global asymptotic behaviour of iterative implicit schemes*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 4 (1994), pp. 1579–1611.