SIAM J. NUMER. ANAL.
Vol. 46, No. 4, pp. 2112–2132

INEXACT NEWTON DOGLEG METHODS[*]

ROGER P. PAWLOWSKI[†], JOSEPH P. SIMONIS[‡], HOMER F. WALKER[‡], AND JOHN N. SHADID[†]

Abstract. The dogleg method is a classical trust-region technique for globalizing Newton's method. While it is widely used in optimization, including large-scale optimization via truncated-Newton approaches, its implementation in general inexact Newton methods for systems of nonlinear equations can be problematic. In this paper, we first outline a very general dogleg method suitable for the general inexact Newton context and provide a global convergence analysis for it. We then discuss certain issues that may arise with the standard dogleg implementational strategy and propose modified strategies that address them. Newton–Krylov methods have provided important motivation for this work, and we conclude with a report on numerical experiments involving a Newton–GMRES dogleg method applied to benchmark CFD problems.

Key words. dogleg methods, trust-region methods, inexact Newton methods, Newton's method, Newton iterative methods, Newton–Krylov methods, Newton–GMRES methods, truncated-Newton methods, globalized Newton methods, fully-coupled solution methods, Navier–Stokes equations

AMS subject classifications. 65H10, 65F10

DOI. 10.1137/050632166

1. Introduction.

The problem of interest is a system of nonlinear equations

$$F(x) = 0, \tag{1.1}$$

where $F : I\!R^n \to I\!R^n$ is continuously differentiable. A classical algorithm for solving (1.1) is *Newton's method*, which generates a sequence of iterates $\{x_k\}$ through steps $s_k = x_{k+1} - x_k$ that satisfy the linear *Newton equation*

$$F'(x_k)s_k = -F(x_k). \tag{1.2}$$

An *inexact Newton method*, defined in [9], is an extension of Newton's method having the following basic form:

Algorithm IN. Inexact newton method [9]

```
Let $x_0$ be given.
For $k = 0, 1, \ldots$ (until convergence) do:
    Choose $\eta_k \in [0, 1)$ and $s_k^{\rm IN}$ such that
(1.3)        $\|F(x_k) + F'(x_k)\, s_k^{\rm IN}\| \le \eta_k \|F(x_k)\|$.
    Set $x_{k+1} = x_k + s_k^{\rm IN}$.
```

[*] Received by the editors May 24, 2005; accepted for publication (in revised form) August 2, 2007; published electronically May 2, 2008.
http://www.siam.org/journals/sinum/46-4/63216.html

[†] Department of Computational Sciences, Sandia National Laboratories, MS 0316, P.O. Box 5800, Albuquerque, NM 87185-0316 (rppawlo@sandia.gov, jnshadi@sandia.gov). The work of these authors was partially supported by the US DOE ASC program and the US DOE Office of Science MICS program at Sandia National Laboratories under contract DE-AC04-94AL85000.

[‡] Department of Mathematical Sciences, Worcester Polytechnic Institute, Worcester, MA 01609-2280 (jpsimoni@wpi.edu, walker@wpi.edu). The work of these authors was supported in part by Sandia National Laboratories under the ASC program and in part by the Sandia National Laboratories Computer Science Research Institute (contract 16099 with WPI). The third author's work was also supported in part by NSF grant DMS-0540684 with Worcester Polytechnic Institute and by the Center for Simulation of Accidental Fires and Explosions funded at the University of Utah by the U.S. Department of Energy under contracts LLNL B341493 and B524196.

The *inexact Newton condition* (1.3) is a relaxation of (1.2) that expresses the defining property of an inexact Newton method. The usual point of view is that, for each k, the *inexact Newton step* s_k^{IN} is chosen to reduce the norm of $F(x_k) + F'(x_k)s$, the *local linear model* of F at x_k, to an extent specified by η_k. Viewed in this way, η_k is often called a *forcing term* (cf. [13]).

While there is no restriction in Algorithm IN on how s_k^{IN} satisfying (1.3) is determined, inexact Newton methods are often implemented as *Newton iterative methods*, in which s_k^{IN} is found by applying an iterative linear algebra method to (1.2) until (1.3) holds for a specified η_k. For most large-scale applications, the most robust and efficient such methods are *Newton–Krylov methods*, in which the iterative linear algebra methods are *Krylov subspace methods.*[1] Newton–Krylov methods, which have provided considerable motivation for this paper, can enable the efficient solution of (1.2) on large-scale distributed architectures and may also allow "matrix-free" implementations (see [25]) that do not require the evaluation or storage of F'. The term "Newton–Krylov" appears to have originated in [3], but these methods date back at least to the *truncated Newton method* for optimization introduced in [10] (see also [28] and [46], which anticipated some algorithmic features), in which the Krylov subspace method is the preconditioned conjugate-gradient (PCG) method or its Lanczos variant. A Newton–Krylov method that uses a specific Krylov subspace method is often designated by appending the name of the Krylov solver to "Newton," as in the *Newton–GMRES method*, which uses the generalized minimal residual (GMRES) method [38]. For recent general references on Newton–Krylov methods, see [24], [25], and [30].

Inexact Newton methods, like all Newton-like methods, must usually be *globalized*, i.e., augmented with certain auxiliary procedures (*globalizations*) that increase the likelihood of convergence to a solution when good initial approximate solutions are not available. Globalizations are typically constructed to test whether a trial step gives satisfactory progress toward a solution and, if necessary, to modify it in some way to obtain a step that does give satisfactory progress. There are two major categories of globalizations: *backtracking* (*linesearch*, *damping*) methods, in which step lengths are adjusted (usually shortened) to obtain satisfactory steps; and *trust-region* methods, in which a step is ideally chosen to minimize the norm of the local linear model of F within a specified "trust region."[2] (More specifically, the trust-region step from an approximate solution x_k is ideally $\arg\min_{\|s\|\leq\delta} \|F(x_k) + F'(x_k)s\|$, where $\delta > 0$ is a given *trust-region radius* and $\|\cdot\|$ is a norm of interest.) Both backtracking and trust-region methods have strong theoretical support; see, e.g., [11], [7], and [12] and the references therein. Backtracking methods are relatively easy to implement; however, each step direction is restricted to be that of the initial trial step. While this step is normally constructed to be a descent direction for $\|F\|$, and is always a descent direction for $\|F\|$ if the step satisfies (1.3) [4, Prop. 3.3], [12, Lem. 7.1], it may be only a weak descent direction, especially if the Jacobian is ill-conditioned [42]. Since trust-region steps are increasingly nearly in the steepest-descent direction as the trust-region radius decreases, trust-region methods have the potential advantage of producing modified steps that may be stronger descent directions than the initial trial step; however, their implementation in practical methods may be problematic.

[1] An introduction to Krylov subspace methods is beyond the scope of this paper; we refer the reader to the surveys [16] and [20] and the books [19], [37], and [48].

[2] See [11, Ch. 6] for a general discussion of classical globalizations and [7] for a more recent and very extensive treatment of trust-region methods.

A fundamental issue is that it is often not feasible to compute the ideal trust-region step with much accuracy (see, e.g., [11, Ch. 6] and [7, Ch. 7]), and various ways of approximating it have been devised.

The *dogleg method* [33], [32] provides perhaps the most popular way of approximating the trust-region step. In the traditional dogleg method for solving (1.1), given an approximate solution x_k, one defines the *dogleg curve* Γ_k^{DL} to be the piecewise-linear curve joining 0, the *Cauchy point* s_k^{CP} (defined to be the minimizer of $\|F(x_k)+F'(x_k)s\|$ in the steepest-descent direction, provided $\|\cdot\|$ is an inner-product norm),[3] and the Newton step $s_k^{\mathrm{N}} \equiv -F'(x_k)^{-1}F(x_k)$. Then, given a trust-region radius $\delta > 0$, the dogleg step s_k is defined as follows: If $\|s_k^{\mathrm{N}}\| \le \delta$, then $s_k = s_k^{\mathrm{N}}$; otherwise, s_k is chosen on Γ_k^{DL} such that $\|s_k\| = \delta$. It can be shown (see, e.g., [11, Ch. 6]) that, as a point s traverses Γ_k^{DL} from 0 to s_k^{N}, $\|s\|$ is monotone increasing and $\|F(x_k)+F'(x_k)s\|$ is monotone decreasing. Thus, the dogleg step s_k is uniquely defined, and $s_k = \arg\min_{s\in\Gamma_k^{\mathrm{DL}},\ \|s\|\le\delta}\|F(x_k)+F'(x_k)s\|$. Moreover, if δ is sufficiently small (specifically, if $\delta \le \|s_k^{\mathrm{CP}}\|$), then s_k is a short step in the steepest-descent direction.

One can extend the dogleg method to the inexact Newton context in a straightforward way by substituting s_k^{IN} for s_k^{N} in the definition of Γ_k^{DL}. The resulting dogleg curve retains the appealing property of offering a continuum of steps ranging from s_k^{IN} to short steps in the steepest-descent direction. However, some desirable properties are lost. First, for any $\eta_k > 0$, no matter how small, the local linear model norm may not decrease monotonically along Γ_k^{DL} between s_k^{CP} and s_k^{IN}. For example, this will be the case if s_k^{N} is nearly on and interior to the line segment joining s_k^{CP} and s_k^{IN}, as illustrated on the left in Figure 1.1. Note that, in this illustration, the point on Γ_k^{DL} having norm δ does not minimize the local linear model norm along Γ_k^{DL} within the trust region. Second, unless η_k is sufficiently small, $\|s\|$ may not increase monotonically as s traverses Γ_k^{DL} from s_k^{CP} to s_k^{IN}; as a consequence, Γ_k^{DL} may have more than one point of intersection with the trust-region boundary, as illustrated on the right in Figure 1.1. (If, in this illustration, s_k^{IN} were moved to the right along the same contour to a point slightly outside the trust-region boundary, then Γ_k^{DL} would have *three* points of intersection with the trust-region boundary.) Thus, with this straightforward extension, the dogleg step cannot be reliably characterized as the unique point on Γ_k^{DL} having norm δ; neither can we always expect a point on Γ_k^{DL} having norm δ to minimize the local linear model norm along Γ_k^{DL} within the trust region.

Remark. These issues do not arise in the case of a truncated Newton method for minimizing $f : \mathbb{R}^n \to \mathbb{R}^1$. In this case, PCG is applied to (1.2) with $F(x_k) = \nabla f(x_k)$ and $F'(x_k) = \nabla^2 f(x_k)$ until either a specified inexact-Newton condition holds, a PCG iterate falls outside of the trust region, or a nonpositive eigenvalue of $\nabla^2 f(x_k)$ is detected. In this context, the model of interest is the local quadratic model of f, rather than the local linear model of $F = \nabla f$. One can show [44, Th. 2.1], [7, section 7.5] that, in an appropriate norm $\|\cdot\|$, the first PCG iterate is the Cauchy point for the local quadratic model and, with the minimizing properties of the PCG iterates, that the model is monotone decreasing along the dogleg curve joining 0, the Cauchy point, and any subsequent PCG iterate. Moreover, the PCG iterate norms are monotone

[3] Our definition of the Cauchy point follows that in [11, Ch. 6]. In [7], the Cauchy point is defined to be the minimizer in the steepest-descent direction *within the trust region.* In section 2 below, we recall the definition of the steepest-descent direction with respect to an inner-product norm of interest and discuss potential difficulties in evaluating it.

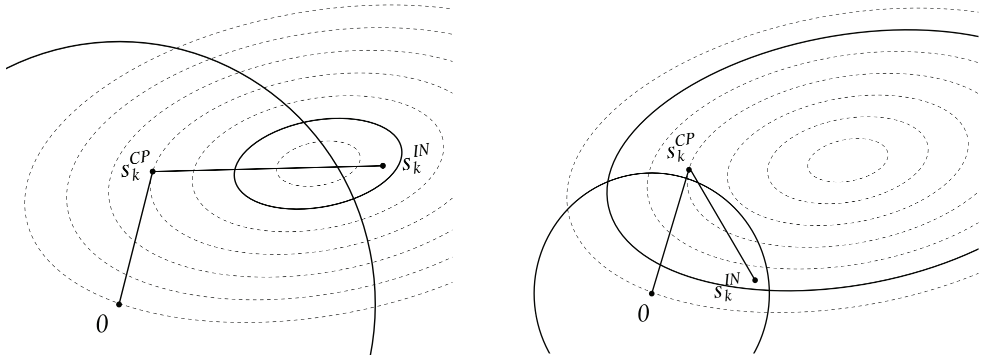

FIG. 1.1. *Illustrative inexact Newton dogleg curves in* $\mathbf{R}^2$. *The ellipses represent level curves of the local linear model norm. The solid ellipses represent* $\{s \in \mathbf{R}^n : \|F(x_k) + F'(x_k)s\| = \eta_k \|F(x_k)\|\}$. *The circular arcs represent trust-region boundaries.*

increasing [44, Th. 2.1], [7, Th. 7.5.1], and it follows that $\|s\|$ is monotone increasing as s traverses the dogleg curve from 0 to the final PCG iterate.

With these issues in mind, our goal in this paper is to develop an understanding of dogleg methods in the general inexact Newton context that will place them on a firm footing, both theoretically and practically. In section 2, we outline a very general inexact Newton dogleg method and provide a convergence analysis for it. Since computing exact Cauchy points may be undesirably expensive or infeasible, as discussed further in section 2, the method allows approximations of these points in addition to inexact Newton steps in defining the dogleg curves.[4] The method also allows a great deal of flexibility in determining steps along the dogleg curves in order to accommodate a variety of strategies for selecting steps. In section 3, we address specific step-selection strategies, noting possible shortcomings of the traditional strategy and suggesting certain alternatives and enhancements. In section 4, we discuss implementational details and report on numerical experiments with Newton–GMRES dogleg methods applied to benchmark problems involving the steady-state Navier–Stokes equations.[5] We conclude with summary observations in section 5.

2. The general method. We assume throughout that $\langle \cdot, \cdot \rangle$ is an inner product of interest and that $\| \cdot \|$ is the associated norm. We use $\langle \cdot, \cdot \rangle_2$ and $\| \cdot \|_2$ to denote the Euclidean inner product and norm, and recall that there is a unique symmetric positive-definite $D \in I\!\!R^{n \times n}$ such that $\langle u, v \rangle = \langle u, Dv \rangle_2$ for all $u, v \in I\!\!R^n$. (In practice, D is likely to be explicitly available, but this is not assumed here.) We define the merit function

$$f(x) \equiv \frac{1}{2} \|F(x)\|^2$$

[4] The use of approximate Cauchy points is particularly relevant in the context of interest here. In order to keep the focus on the issues of major interest, we do not consider inaccuracy from other sources. See [6] for a general discussion of inaccuracy in inexact Newton methods and [2] and [5] for a treatment of the effects of particular sources of inaccuracy in Jacobian-vector products.

[5] Globalization is usually less critical for time-dependent problems, since time-steps are usually sufficiently small to maintain convergence of inexact Newton iterates in implicit time-stepping methods. However, globalization has been used to advantage in some cases; for example, it has been observed in groundwater flow simulations that globalization appears necessary (and a linesearch globalization is sufficient) to achieve desirably long time steps [49].

and note that f is continuously differentiable since F is, with $\nabla f(x) = F'(x)^T DF(x)$. One has that $\nabla f(x)^T s = \langle \nabla f(x), s\rangle_2 = \langle D^{-1}\nabla f(x), s\rangle$; hence, the steepest-descent direction for f with respect to $\langle \cdot, \cdot \rangle$ is

$$d(x) \equiv -D^{-1}\nabla f(x) = -D^{-1}F'(x)^T DF(x). \tag{2.1}$$

Accurately evaluating $d(x)$ may be problematic in practice for at least two reasons: first, solving systems with D may be undesirable, even if D is available; second, multiplication by $F'(x)^T$ may be infeasible, e.g., in applications of "matrix-free" Newton–Krylov methods in which $F'(x)$ is unavailable. However, useful approximations of $d(x)$ may be available in many circumstances. If $F'(x)^T$-products can be evaluated, then $-\nabla f(x)$ may be an acceptable substitute for $d(x)$; see Remark 2 after Theorem 2.1. If accurate $F'(x)^T$-products are not available, then one may be able to approximate them using approximations of $F'(x)$ that can be readily computed, e.g., by omitting terms in $F'(x)$ that cannot be easily evaluated or that lie outside a desirable sparsity pattern. Additionally, in the Newton–Krylov context, certain approximations of $d(x)$ may be available at little cost, as explained further at the end of this section. Thus, in outlining our general inexact Newton dogleg method, we allow at each iterate x_k an approximate steepest-descent direction $\hat{d}_k \approx d(x_k)$ and prescribe in Theorem 2.1 how good this approximation must be in order to ensure desirable convergence properties of the method.

With each $\hat{d}_k$, we have an approximate Cauchy point

$$\hat{s}_k^{\mathrm{CP}} \equiv \arg\min\{\|F(x_k) + F'(x_k)s\| : \ s = \lambda \hat{d}_k, -\infty < \lambda < \infty\}.$$

(Here, and in the following, we use "ˆ" to designate approximations derived from $\hat{d}_k$.) Then, given an inexact Newton step s_k^{IN}, we define an approximate dogleg curve $\hat{\Gamma}_k^{\mathrm{DL}}$ to be the piecewise-linear curve connecting 0, $\hat{s}_k^{\mathrm{CP}}$, and s_k^{IN}. Our general method allows great latitude in selecting trial steps on this approximate dogleg curve within the trust region, imposing only very mild minimum-length requirements.

Our test for accepting a step is based on a comparison of the actual reduction of $\|F\|$ and the reduction "predicted" by the local linear model, defined at the kth iteration by, respectively,

$$\begin{aligned} \mathit{ared}_k &\equiv \|F(x_k)\| - \|F(x_k + s_k)\|, \\ \mathit{pred}_k &\equiv \|F(x_k)\| - \|F(x_k) + F'(x_k)\, s_k\|. \end{aligned} \tag{2.2}$$

Specifically, our test requires $\mathit{ared}_k \geq t \cdot \mathit{pred}_k > 0$ for each k, where $t \in (0,1)$ is independent of k. Tests of this type have been used by a number of others; see, e.g., [14], [15], [34], and [12] for such tests applied to systems of nonlinear equations and [27], [43], and [7] for analogous tests applied to unconstrained minimization.

The following is our general inexact Newton dogleg method. See section 4 for the specific parameter values and ancillary procedures used in our test implementation.

Algorithm INDL. Inexact newton dogleg method

```
Let $x_0$, $\eta_{\max} \in [0,1)$, $t \in (0,1)$, $\theta_{\max} \in (0,1)$, $\delta_{\min} > 0$ and
  $\delta \geq \delta_{\min}$ be given.
For $k = 0, 1, \ldots$ (until convergence) do:
    Choose $\eta_k \in [0, \eta_{\max}]$ and $s_k^{\mathrm{IN}}$ such that
            $\|F(x_k) + F'(x_k)\, s_k^{\mathrm{IN}}\| \leq \eta_k \|F(x_k)\|$.
```

```
Determine $\hat{d}_k$ and evaluate $\hat{s}_k^{\mathrm{CP}}$.
Determine $s_k \in \hat{\Gamma}_k^{\mathrm{DL}}$ with $\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\} \le \|s_k\| \le \delta$.
While $ared_k < t \cdot pred_k$ do:
    If $\delta = \delta_{\min}$, stop; else choose $\theta \in (0, \theta_{\max}]$.
    Update $\delta \leftarrow \max\{\theta\delta, \delta_{\min}\}$.
    Redetermine $s_k \in \hat{\Gamma}_k^{\mathrm{DL}}$ with $\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\} \le \|s_k\| \le \delta$.
Set $x_{k+1} = x_k + s_k$ and update $\delta$ in $[\delta_{\min}, \infty)$.
```

As it is stated, Algorithm INDL requires determining both s_k^{IN} and $\hat{s}_k^{\mathrm{CP}}$ before determining the initial $s_k \in \hat{\Gamma}_k^{\mathrm{DL}}$. In a practical implementation, only one of these may be needed; see section 3.1.

The algorithm employs $\delta_{\min} > 0$ that serves as a lower bound on allowable trust-region radii and is also used in determining a lower bound on admissible dogleg step lengths. This $\delta_{\min}$ is used in the proof of Theorem 2.1. Such a $\delta_{\min}$ is likely to be a part of any practical algorithm and, in practice, can be related to a step-length stopping tolerance or taken so small that it does not undesirably restrict choices of the trust-region radii or the dogleg steps.

It is easily verified that the while-loop cannot continue indefinitely; that is, once an initial s_k has been obtained, either an acceptable s_k is determined or the algorithm terminates with $\delta = \delta_{\min}$ after at most a finite number of iterations of the while-loop. For the purposes of Theorem 2.1, there is complete freedom in choosing $\theta \in (0, \theta_{\max}]$. In particular, because of the constrained update $\delta \leftarrow \max\{\theta\delta, \delta_{\min}\}$, it is not necessary to impose a positive lower bound on θ, although one may well do this in practice. (There are a number of practical possibilities for choosing θ; see, in particular, [11] for choices based on minimizing interpolating polynomials over a fixed subinterval $[\theta_{\min}, \theta_{\max}] \subset (0, 1)$.) Similarly, for the purposes of Theorem 2.1, there is complete freedom in updating δ in $[\delta_{\min}, \infty)$ following the while-loop. Of course, one would want to do this judiciously in practice; see, e.g., [11] and [7] for practically effective procedures.

Our global convergence result for Algorithm INDL is given in Theorem 2.1 below. For this, we recall that $x \in \mathbb{R}^n$ is a *stationary point* of $\|F\|$ if $\|F(x)\| \le \|F(x) + F'(x)\, s\|$ for every $s \in \mathbb{R}^n$. In the present context, x is a stationary point if and only if $\nabla f(x) = 0$.

THEOREM 2.1. *Assume that F is continuously differentiable. Suppose that $\{x_k\}$ is produced by Algorithm* INDL *and that, for d defined in* (2.1) *and some $\epsilon > 0$,*

$$\frac{\langle \hat{d}_k, d(x_k)\rangle}{\|\hat{d}_k\| \|d(x_k)\|} \ge \epsilon \tag{2.3}$$

for every k. If x_* is a limit point of $\{x_k\}$, then x_* is a stationary point of $\|F\|$. If, additionally, $F'(x_*)$ is nonsingular, then $F(x_*) = 0$ and $x_k \to x_*$. Moreover, for all sufficiently large k, the initial s_k is accepted without modification in the while-loop, and $s_k = s_k^{\mathrm{IN}}$ is an admissible step.

Remark 1. In Theorem 2.1, we implicitly assume that $F(x_k)$, $\hat{d}_k$, and $d(x_k)$ are nonzero for every k.

Remark 2. Inequality (2.3) states that the vectors $\hat{d}_k$ are uniformly bounded away from orthogonality with the vectors $d(x_k)$. Note that

$$\frac{\langle -\nabla f(x_k), d(x_k)\rangle}{\| - \nabla f(x_k)\| \|d(x_k)\|} = \frac{\|\nabla f(x_k)\|_2^2}{\langle \nabla f(x_k), D\nabla f(x_k)\rangle_2^{1/2} \langle \nabla f(x_k), D^{-1}\nabla f(x_k)\rangle_2^{1/2}} \ge \epsilon$$

for $0 < \epsilon \le \kappa_2(D)^{-1/2}$, where $\kappa_2(D) = \|D\|_2\|D^{-1}\|_2$. Thus (2.3) holds for some $\epsilon > 0$ if $\hat{d}_k$ is $-\nabla f(x_k)$ or a sufficiently accurate approximation of it.

Remark 3. If $x_k \to x_*$ with $F(x_*) = 0$ and $F'(x_*)$ nonsingular and if $s_k = s_k^{\rm IN}$ for all sufficiently large k, then the convergence is ultimately governed by the choices of the forcing terms η_k as in the local convergence theory of [9] and [13].

Proof. Let $\{x_k\}$ be produced by Algorithm INDL and suppose that (2.3) holds for every k. Since $\langle F'(x_k)\hat{d}_k, F(x_k)\rangle = -\langle \hat{d}_k, d(x_k)\rangle \neq 0$ by (2.3), we have that $F'(x_k)\hat{d}_k \neq 0$ for every k. Then a straightforward calculation yields

$$(2.4) \qquad \|\hat{s}_k^{\rm CP}\| = \frac{|\langle F(x_k), F'(x_k)\hat{d}_k\rangle|}{\|F'(x_k)\hat{d}_k\|^2}\|\hat{d}_k\|,$$

$$(2.5) \qquad \hat{\eta}_k^{\rm CP} \equiv \frac{\|F(x_k) + F'(x_k)\hat{s}_k^{\rm CP}\|}{\|F(x_k)\|} = \sqrt{1 - \frac{\langle F(x_k), F'(x_k)\hat{d}_k\rangle^2}{\|F(x_k)\|^2\|F'(x_k)\hat{d}_k\|^2}}.$$

Our plan is to build on (2.4)–(2.5) to obtain upper bounds on $\|\hat{s}_k^{\rm CP}\|$ and $\hat{\eta}_k^{\rm CP}$ at points of interest that will allow us to use results from [12], which are stated as Theorem 2.2 following the proof. Equation (2.4) immediately gives

$$(2.6) \qquad \|\hat{s}_k^{\rm CP}\| \le \frac{\|F(x_k)\|\,\|\hat{d}_k\|}{\|F'(x_k)\hat{d}_k\|}\,.$$

Also, from (2.3), we obtain

$$\begin{aligned}\epsilon &\le \frac{\langle \hat{d}_k, d(x_k)\rangle}{\|\hat{d}_k\|\,\|d(x_k)\|} = -\frac{\langle \hat{d}_k, F'(x_k)^T D F(x_k)\rangle_2}{\|\hat{d}_k\|\,\|d(x_k)\|}\\ &= -\frac{\langle F'(x_k)\hat{d}_k, F(x_k)\rangle}{\|\hat{d}_k\|\,\|d(x_k)\|} \le \frac{\|F'(x_k)\hat{d}_k\|}{\|\hat{d}_k\|}\cdot\frac{\|F(x_k)\|}{\|d(x_k)\|}\,,\end{aligned}$$

whence

$$\frac{\|\hat{d}_k\|}{\|F'(x_k)\hat{d}_k\|} \le \frac{\|F(x_k)\|}{\epsilon\|d(x_k)\|}\,.$$

Then (2.6) yields

$$(2.7) \qquad \|\hat{s}_k^{\rm CP}\| \le \frac{\|F(x_k)\|^2}{\epsilon\|d(x_k)\|}\,.$$

From (2.3), we also have

$$\begin{aligned}-\frac{\langle F(x_k), F'(x_k)\hat{d}_k\rangle}{\|F(x_k)\|\,\|F'(x_k)\hat{d}_k\|} &= \frac{\langle \hat{d}_k, d(x_k)\rangle}{\|\hat{d}_k\|\,\|d(x_k)\|}\cdot\frac{\|\hat{d}_k\|\,\|d(x_k)\|}{\|F(x_k)\|\,\|F'(x_k)\hat{d}_k\|}\\ &\ge \epsilon\cdot\frac{\|\hat{d}_k\|}{\|F'(x_k)\hat{d}_k\|}\cdot\frac{\|d(x_k)\|}{\|F(x_k)\|} \ge \frac{\epsilon}{\|F'(x_k)\|}\cdot\frac{\|d(x_k)\|}{\|F(x_k)\|}\,,\end{aligned}$$

and it follows from (2.5) that

$$(2.8) \qquad \hat{\eta}_k^{\rm CP} \le \sqrt{1 - \frac{\epsilon^2}{\|F'(x_k)\|^2}\cdot\frac{\|d(x_k)\|^2}{\|F(x_k)\|^2}}\,.$$

Let x_* be a limit point of $\{x_k\}$, and suppose that x_* is not a stationary point of $\|F\|$. Then $F(x_*) \neq 0$. Additionally, $d(x_*) = -D^{-1}\nabla f(x_*) \neq 0$ since x_* is not a stationary point of $\|F\|$. Then, by continuity, there is a neighborhood N_* of x_* such that $0 < \inf_{x\in N_*} \|F(x)\| \leq \sup_{x\in N_*} \|F(x)\| < \infty$, $0 < \inf_{x\in N_*} \|d(x)\|$, and $0 < \inf_{x\in N_*} \|F'(x)\| \leq \sup_{x\in N_*} \|F'(x)\| < \infty$. Then one sees from (2.7)–(2.8) that there are bounds M and $\eta_{\max}^{\mathrm{CP}} < 1$ such that $\|\hat{s}_k^{\mathrm{CP}}\| \leq M$ and $\hat{\eta}_k^{\mathrm{CP}} \leq \eta_{\max}^{\mathrm{CP}}$ whenever $x_k \in N_*$.

Suppose that $x_k \in N_*$ and s_k is determined by Algorithm INDL. If s_k lies on $\hat{\Gamma}_k^{\mathrm{DL}}$ between $\hat{s}_k^{\mathrm{CP}}$ and s_k^{IN}, then it follows from norm convexity that

$$(2.9)\qquad \|F(x_k) + F'(x_k)\, s_k\| \leq \max\{\hat{\eta}_k^{\mathrm{CP}}, \eta_k\}\|F(x_k)\| \leq \max\{\eta_{\max}^{\mathrm{CP}}, \eta_{\max}\}\|F(x_k)\|.$$

If s_k lies on $\hat{\Gamma}_k^{\mathrm{DL}}$ between 0 and $\hat{s}_k^{\mathrm{CP}}$, then $\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\} \leq \|s_k\| \leq \|\hat{s}_k^{\mathrm{CP}}\|$. We have

$$\|F(x_k)\| - \|F'(x_k)s_k^{\mathrm{IN}}\| \leq \|F(x_k) + F'(x_k)s_k^{\mathrm{IN}}\| \leq \eta_k\|F(x_k)\| \leq \eta_{\max}\|F(x_k)\|,$$

which implies

$$(2.10)\qquad \|s_k^{\mathrm{IN}}\| \geq \frac{(1-\eta_{\max})}{\|F'(x_k)\|}\|F(x_k)\|.$$

The right-hand side of (2.10) is bounded away from zero for $x_k \in N_*$; thus, there is a $\bar{\delta} > 0$ such that $\bar{\delta} \leq \min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\}$ whenever $x_k \in N_*$. Since the local linear model norm is monotone decreasing along the segment of $\hat{\Gamma}_k^{\mathrm{DL}}$ between 0 and $\hat{s}_k^{\mathrm{CP}}$, we have, again using norm convexity, that

$$(2.11)\qquad \begin{aligned} \|F(x_k) + F'(x_k)\, s_k\| &\leq \left\|F(x_k) + F'(x_k)\left(\frac{\bar{\delta}}{\|\hat{s}_k^{\mathrm{CP}}\|}\hat{s}_k^{\mathrm{CP}}\right)\right\| \\ &\leq \left(1 - \frac{\bar{\delta}}{\|\hat{s}_k^{\mathrm{CP}}\|}\right)\|F(x_k)\| + \frac{\bar{\delta}}{\|\hat{s}_k^{\mathrm{CP}}\|}\|F(x_k) + F'(x_k)\,\hat{s}_k^{\mathrm{CP}}\| \\ &\leq \left[1 - \frac{\bar{\delta}}{\|\hat{s}_k^{\mathrm{CP}}\|}(1-\hat{\eta}_k^{\mathrm{CP}})\right]\|F(x_k)\| \\ &\leq \left[1 - \frac{\bar{\delta}}{M}(1-\eta_{\max}^{\mathrm{CP}})\right]\|F(x_k)\|. \end{aligned}$$

From (2.9) and (2.11), one concludes that whenever $x_k \in N_*$, we have $\|F(x_k) + F'(x_k)\, s_k\| \leq \bar{\eta}\|F(x_k)\|$, where

$$\bar{\eta} \equiv \max\left\{\eta_{\max}^{\mathrm{CP}}, \eta_{\max}, 1 - \frac{\bar{\delta}}{M}(1-\eta_{\max}^{\mathrm{CP}})\right\} < 1;$$

therefore, with pred_k defined as in (2.2),

$$(2.12)\qquad \frac{\mathit{pred}_k}{\|F(x_k)\|} = \frac{\|F(x_k)\| - \|F(x_k) + F'(x_k)\, s_k\|}{\|F(x_k)\|} \geq (1-\bar{\eta}) > 0.$$

Since $x_k \in N_*$ for infinitely many values of k, (2.12) implies that $\sum_{k=0}^{\infty} \mathit{pred}_k/\|F(x_k)\|$ diverges, and it follows from Theorem 2.2 that $F(x_*) = 0$. This is a contradiction; therefore, x_* must be a stationary point of $\|F\|$.

Suppose now that x_* is a limit point of $\{x_k\}$ and that $F'(x_*)$ is nonsingular. Then x_* must be a stationary point of $\|F\|$, and, since $F'(x_*)$ is nonsingular, we must have

$F(x_*) = 0$. Additionally, we can let N_* be a neighborhood of x_* such that $F'(x)$ is invertible for $x \in N_*$ and $\sup_{x\in N_*}\{\|F(x)\|, \|F'(x)\|, \|F'(x)^{-1}\|\} < \infty$. Noting that

$$\frac{\|F(x_k)\|}{\|d(x_k)\|} \le \| \left[D^{-1}F'(x_k)^T D\right]^{-1} \|, \tag{2.13}$$

we have from (2.7) that, for $x_k \in N_*$,

$$\|\hat{s}_k^{\mathrm{CP}}\| \le \frac{1}{\epsilon} \| \left[D^{-1}F'(x_k)^T D\right]^{-1} \| \, \|F(x_k)\|. \tag{2.14}$$

One concludes that there is again a bound M such that $\|\hat{s}_k^{\mathrm{CP}}\| \le M$ whenever $x_k \in N_*$. Moreover, it follows from (2.13) and (2.8) that, for $x_k \in N_*$,

$$\hat{\eta}_k^{\mathrm{CP}} \le \sqrt{1 - \frac{\epsilon^2}{\|F'(x_k)\|^2 \, \| \left[D^{-1}F'(x_k)^T D\right]^{-1} \|^2}} \, .$$

Consequently, there is again a bound $\eta_{\max}^{\mathrm{CP}} < 1$ such that $\hat{\eta}_k^{\mathrm{CP}} \le \eta_{\max}^{\mathrm{CP}}$ whenever $x_k \in N_*$.

Suppose that $x_k \in N_*$ and s_k is determined by Algorithm INDL. If s_k lies on $\hat{\Gamma}_k^{\mathrm{DL}}$ between $\hat{s}_k^{\mathrm{CP}}$ and s_k^{IN}, then (2.9) holds as before. If s_k lies on $\hat{\Gamma}_k^{\mathrm{DL}}$ between 0 and $\hat{s}_k^{\mathrm{CP}}$, then we again have $\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\} \le \|s_k\| \le \|\hat{s}_k^{\mathrm{CP}}\|$. Reasoning as before, we have that

$$\begin{aligned}
\|F(x_k) + F'(x_k)\, s_k\| \;&\le\; \left\| F(x_k) + F'(x_k) \left(\frac{\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\}}{\|\hat{s}_k^{\mathrm{CP}}\|} \hat{s}_k^{\mathrm{CP}} \right) \right\| \\
&\le\; \left(1 - \frac{\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\}}{\|\hat{s}_k^{\mathrm{CP}}\|}\right) \|F(x_k)\| \\
&\qquad + \frac{\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\}}{\|\hat{s}_k^{\mathrm{CP}}\|} \|F(x_k) + F'(x_k)\, \hat{s}_k^{\mathrm{CP}}\| \qquad (2.15) \\
&\le\; \left[1 - \frac{\min\{\delta_{\min}, \|s_k^{\mathrm{IN}}\|\}}{\|\hat{s}_k^{\mathrm{CP}}\|} (1 - \hat{\eta}_k^{\mathrm{CP}})\right] \|F(x_k)\| \\
&\le\; \left[1 - \min\left\{ \frac{\delta_{\min}}{M}, \frac{\|s_k^{\mathrm{IN}}\|}{\|\hat{s}_k^{\mathrm{CP}}\|} \right\} (1 - \eta_{\max}^{\mathrm{CP}})\right] \|F(x_k)\|.
\end{aligned}$$

Inequality (2.10) again holds, and (2.10) and (2.14) yield

$$\frac{\|s_k^{\mathrm{IN}}\|}{\|\hat{s}_k^{\mathrm{CP}}\|} \ge \frac{\epsilon(1-\eta_{\max})}{\|F'(x_k)\| \, \| \left[D^{-1}F'(x_k)^T D\right]^{-1} \|} \ge \inf_{x\in N_*} \frac{\epsilon(1-\eta_{\max})}{\|F'(x)\| \, \| \left[D^{-1}F'(x)^T D\right]^{-1} \|} > 0.$$

It follows that there is a $\bar{\delta} > 0$ independent of k such that

$$\bar{\delta} \le \min\left\{ \frac{\delta_{\min}}{M}, \frac{\|s_k^{\mathrm{IN}}\|}{\|\hat{s}_k^{\mathrm{CP}}\|} \right\},$$

and we have from (2.15) that

$$\|F(x_k) + F'(x_k)\, s_k\| \le \left[1 - \bar{\delta}(1 - \eta_{\max}^{\mathrm{CP}})\right] \|F(x_k)\|.$$

As before, one concludes that (2.12) holds whenever $x_k \in N_*$, with $\bar{\eta}$ now given by

$$\bar{\eta} \equiv \max\{\eta_{\max}^{\mathrm{CP}}, \eta_{\max}, 1-\bar{\delta}(1-\eta_{\max}^{\mathrm{CP}})\} < 1,$$

and we again have that $\sum_{k=0}^{\infty} pred_k/\|F(x_k)\|$ diverges. Since $F'(x_*)$ is nonsingular, it follows from Theorem 2.2 below that $x_k \to x_*$.

To complete the proof, we note that, since $x_k \to x_*$ as $k \to \infty$ with $F(x_*) = 0$ and $F'(x_*)$ nonsingular, it follows from (2.14) that $\hat{s}_k^{\mathrm{CP}} \to 0$ as $k \to \infty$. Additionally, writing $F'(x_k)\, s_k = -F(x_k) + r_k$ with $\|r_k\| \le \eta_k \|F(x_k)\|$, we have that

$$\begin{aligned} \|s_k^{\mathrm{IN}}\| \; &= \; \|F'(x_k)^{-1}[-F(x_k)+r_k]\| \le \|F'(x_k)^{-1}\|\,(1+\eta_k)\,\|F(x_k)\| \\ &\le \; 2\,\|F'(x_k)^{-1}\|\,\|F(x_k)\|. \end{aligned}$$

Consequently, $s_k^{\mathrm{IN}} \to 0$ as $k \to \infty$. One easily verifies that, for all sufficiently large k, every admissible $s_k \in \hat{\Gamma}_k^{\mathrm{DL}}$ must be sufficiently small that $ared_k \ge t \cdot pred_k$ holds; thus, for all sufficiently large k, the initial s_k is accepted without modification. Finally, for all sufficiently large k, one has $\|s_k^{\mathrm{IN}}\| \le \delta_{\min} \le \delta$, and $s_k = s_k^{\mathrm{IN}}$ is admissible. □

The following is the result from [12] used in the proof of Theorem 2.1.

THEOREM 2.2 (see [12, Cor. 3.6]). *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable and assume that $\{x_k\} \subset \mathbb{R}^n$ is such that $pred_k \ge 0$ and $ared_k \ge t \cdot pred_k$ for each k, where $t \in (0,1)$ is independent of k and $ared_k$ and $pred_k$ are given by (2.2) with $s_k \equiv x_{k+1} - x_k$. If $\sum_{k\ge 0} pred_k/\|F(x_k)\|$ is divergent, then $F(x_k) \to 0$. If, in addition, x_* is a limit point of $\{x_k\}$ such that $F'(x_*)$ is invertible, then $F(x_*) = 0$ and $x_k \to x_*$.*

Applications to Newton–GMRES and Newton–Arnoldi methods. It follows from observations in [3, section 4] that if either GMRES or the Arnoldi method [35] is used (without restarting) to solve (1.2), then the orthogonal projection of $d(x_k)$ determined by (2.1) onto each Krylov subspace generated by the method is available at very little cost. Using results from [4], we show how Theorem 2.1 can be applied when these projections are used in Algorithm INDL to approximate the vectors $d(x_k)$, provided that the condition numbers $\kappa(F'(x_k)) \equiv \|F'(x_k)\|\,\|F'(x_k)^{-1}\|$ are bounded uniformly in k. In the spirit of [4], we show this for orthogonal projections onto subspaces that are not necessarily Krylov subspaces.

Suppose that $x \in \mathbb{R}^n$, $\mathcal{K}$ is a subspace of $\mathbb{R}^n$, and $s \in \mathcal{K}$ is such that $\|F(x) + F'(x)s\| \le \eta\|F(x)\|$ for some $\eta \in [0,1)$. Let $\hat{d}$ denote the orthogonal projection onto $\mathcal{K}$ of $d(x)$ given by (2.1). A straightforward extension of [4, Cor. 3.5] to the context of a general inner-product norm gives

$$\|\hat{d}\| \ge \frac{1-\eta}{(1+\eta)\kappa(F'(x))}\|d(x)\|.$$

Since $\langle \hat{d}, d(x)\rangle = \|\hat{d}\|^2$, it follows that

$$\frac{|\langle \hat{d}, d(x)\rangle|}{\|\hat{d}\|\,\|d(x)\|} = \frac{\|\hat{d}\|}{\|d\|} \ge \frac{1-\eta}{(1+\eta)\kappa(F'(x))}. \tag{2.16}$$

This immediately gives the following corollary of Theorem 2.1.

COROLLARY 2.3. *Assume that F is continuously differentiable and that $\{x_k\}$ is produced by Algorithm* INDL. *For each k, suppose that there is a subspace $\mathcal{K}_k \subseteq \mathbb{R}^n$*

such that $s_k^{\mathrm{IN}} \in \mathcal{K}_k$*, and let* $\hat{d}_k$ *denote the orthogonal projection onto* $\mathcal{K}_k$ *of* $d(x_k)$ *determined by* (2.1)*. If* $\{\kappa(F'(x_k))\}$ *is bounded and if* x_* *is a limit point of* $\{x_k\}$*, then* $F(x_*) = 0$ *and* $x_k \to x_*$*. Moreover, for all sufficiently large* k*, the initial* s_k *is accepted without modification in the while-loop, and* $s_k = s_k^{\mathrm{IN}}$ *is an admissible step.*

Proof. Let M be such that $\kappa(F'(x_k)) \le M$ for all k. Then, since $\eta_k \in [0, \eta_{\max}]$, (2.16) implies that

$$\frac{|\langle \hat{d}_k, d(x_k)\rangle|}{\|\hat{d}_k\| \, \|d(x_k)\|} \ge \frac{1-\eta_{\max}}{(1+\eta_{\max})M}$$

for each k, and (2.3) holds with $\epsilon = \frac{1-\eta_{\max}}{(1+\eta_{\max})M}$. The result follows from Theorem 2.1. □

3. Step selection strategies.

Algorithm INDL allows great flexibility in determining each step $s_k \in \hat{\Gamma}_k^{\mathrm{DL}}$, and particular choices of the steps may strongly affect the behavior of the algorithm. In this section, we discuss specific strategies for selecting admissible steps, first recalling the traditional strategy and then suggesting an alternative that may have advantages. We conclude by outlining certain refinements with which these strategies can be augmented.

3.1. Two strategies.

The traditional strategy outlined in the following is a straightforward adaptation of the usual procedure for determining dogleg steps in the exact-Newton context (see, e.g., [11]).

Procedure 3.1. Traditional strategy

```
If $\|s_k^{\mathrm{IN}}\| \le \delta$, set $s_k = s_k^{\mathrm{IN}}$.
Else,
    If $\|\hat{s}_k^{\mathrm{CP}}\| \ge \delta$, set $s_k = (\delta/\|\hat{s}_k^{\mathrm{CP}}\|)\hat{s}_k^{\mathrm{CP}}$.
    Else, set $s_k = (1-\gamma)\hat{s}_k^{\mathrm{CP}} + \gamma s_k^{\mathrm{IN}}$ for $\gamma \in (0,1)$
      such that $\|s_k\| = \delta$.
```

This procedure always determines s_k uniquely, notwithstanding that, as noted in section 1, the usual exact-Newton characterization of the dogleg step may fail to do so. However, there are still issues that may be of concern. As seen in the left illustration in Figure 1.1, for any $\eta_k > 0$, no matter how small, s_k determined by Procedure 3.1 may not minimize the local linear model norm along $\hat{\Gamma}_k^{\mathrm{DL}}$ within the trust region. Perhaps of greater concern is that, if η_k is not small, then, as seen in the right illustration in Figure 1.1, the procedure may specify $s_k = s_k^{\mathrm{IN}}$ without taking $\hat{s}_k^{\mathrm{CP}}$ into account, even though there are steps in the $\hat{s}_k^{\mathrm{CP}}$-direction, or biased toward it, that give significantly greater reduction of the local linear model norm along $\hat{\Gamma}_k^{\mathrm{DL}}$ within the trust region.

As a step toward addressing these concerns, we offer the following.

Procedure 3.2. Alternative strategy

```
If $\|\hat{s}_k^{\mathrm{CP}}\| \ge \delta$, set $s_k = (\delta/\|\hat{s}_k^{\mathrm{CP}}\|)\hat{s}_k^{\mathrm{CP}}$.
Else,
    If $\|F(x_k) + F'(x_k)\,\hat{s}_k^{\mathrm{CP}}\| \le \eta_k \|F(x_k)\|$, set $s_k = \hat{s}_k^{\mathrm{CP}}$.
    Else,
        If $\|s_k^{\mathrm{IN}}\| \le \delta$, set $s_k = s_k^{\mathrm{IN}}$.
        Else, set $s_k = (1-\gamma)\hat{s}_k^{\mathrm{CP}} + \gamma s_k^{\mathrm{IN}}$ for $\gamma \in (0,1)$
          such that $\|s_k\| = \delta$.
```

In Procedure 3.2, $\hat{s}_k^{\rm CP}$ is examined first. If it lies outside the trust region, then, as in the traditional strategy in the exact-Newton context, s_k is a step in the $\hat{s}_k^{\rm CP}$-direction and scaled to lie on the trust-region boundary. If $\hat{s}_k^{\rm CP}$ lies within the trust region and also satisfies the prescribed inexact Newton condition, then $\hat{s}_k^{\rm CP}$ achieves the desired linear model norm reduction, and we take $s_k = \hat{s}_k^{\rm CP}$. If $\hat{s}_k^{\rm CP}$ does not satisfy the inexact Newton condition, then $s_k^{\rm IN}$ is examined, and s_k is taken to be either $s_k^{\rm IN}$, if it lies within the trust region, or, otherwise, the unique point between $\hat{s}_k^{\rm CP}$ and $s_k^{\rm IN}$ at which $\hat{\Gamma}_k^{\rm DL}$ intersects the trust-region boundary, as in the traditional strategy in the exact-Newton context.

Note that, in a practical implementation within Algorithm INDL, Procedure 3.2 would require computing $\hat{s}_k^{\rm CP}$ for each k but would require computing $s_k^{\rm IN}$ only when $\hat{s}_k^{\rm CP}$ lies within the trust region and does not satisfy the inexact Newton condition. In contrast, Procedure 3.1 would require computing $s_k^{\rm IN}$ for each k but would require computing $\hat{s}_k^{\rm CP}$ only when $s_k^{\rm IN}$ lies outside the trust region. Also, in the Newton-iterative context, Procedure 3.2 offers the possibility of using $\hat{s}_k^{\rm CP}$ as the initial approximate solution in the iterative solver when it is necessary to compute $s_k^{\rm IN}$, thereby allowing advantage to be taken of the linear model norm reduction already achieved by $\hat{s}_k^{\rm CP}$.

3.2. Further refinements. As noted in section 1 and seen in the left illustration in Figure 1.1, the local linear model norm may not be monotone decreasing along the dogleg curve and, indeed, may not be minimized at the intersection of the dogleg curve and the trust-region boundary. Note that, in this illustration, this intersection would be the dogleg step determined by both Procedure 3.1 and Procedure 3.2. We reproduce this illustration on the left in Figure 3.1 and include the point s_{min} at which the local linear model norm is minimized along the dogleg curve within the trust region. As we show in the following, minimizers such as s_{min} can be readily computed; thus s_{min} is likely to be the preferred dogleg step.

To show further possibilities, we include the illustration on the right in Figure 3.1, in which the minimizer of the local linear model norm within the trust region along the line joining $s_k^{\rm CP}$ and $s_k^{\rm IN}$ (again denoted by s_{min}) occurs *beyond* the end of the dogleg curve. In this illustration, both Procedure 3.1 and Procedure 3.2 would determine the dogleg step to be $s_k^{\rm IN}$. (In practice, Procedure 3.1 would do so without computing $s_k^{\rm CP}$, while Procedure 3.2 would compute both $s_k^{\rm CP}$ and $s_k^{\rm IN}$.) Since s_{min} satisfies the inexact Newton condition satisfied by $s_k^{\rm IN}$, we can regard it as an admissible dogleg step simply by replacing $s_k^{\rm IN}$ by s_{min} in determining the dogleg curve. Since this s_{min} can also be readily computed, it is likely to be the preferred dogleg step in this case.

In the following, we formulate refinements of Procedures 3.1 and 3.2 that determine dogleg steps to be local linear model norm minimizers such as those illustrated in Figure 3.1. In these refinements, we modify the procedures to minimize explicitly only along the line joining $\hat{s}_k^{\rm CP}$ and $s_k^{\rm IN}$ in the final steps, when both $\hat{s}_k^{\rm CP}$ and $s_k^{\rm IN}$ would have been computed in a practical implementation. Minimizing along the line joining zero and $\hat{s}_k^{\rm CP}$ is already ensured by the definition of $\hat{s}_k^{\rm CP}$ and the convexity of the local linear model norm along that line. Additionally, it seems unlikely to be worthwhile to compute either $\hat{s}_k^{\rm CP}$ or $s_k^{\rm IN}$ only for the purposes of minimization when the procedures do not otherwise require them. For example, in the illustration on the right in Figure 1.1, computing both $\hat{s}_k^{\rm CP}$ (which is just $s_k^{\rm CP}$ in this case) and $s_k^{\rm IN}$ would, in fact, allow one to determine the minimizer along the entire dogleg curve within the trust region; however, not much would be gained over the scaled step in the $s_k^{\rm CP}$-direction determined by Procedure 3.2. In any event, it would be easy to further modify the procedures to perform this additional minimization, if desired.

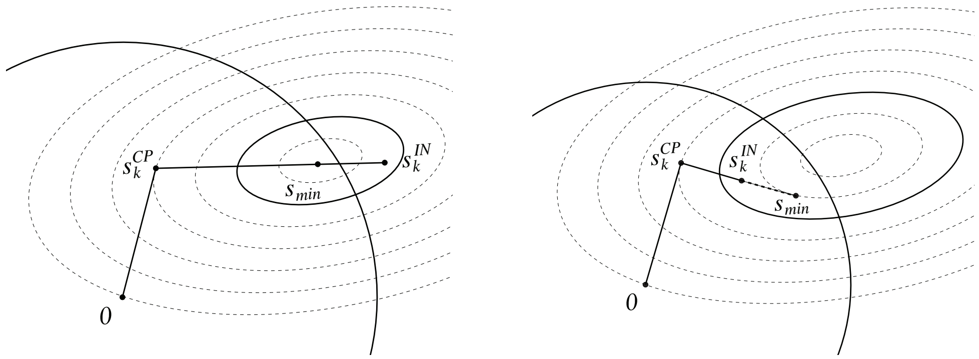

FIG. 3.1. *Illustrative local linear model norm minimizers in* $\boldsymbol{R}^2$. *The ellipses represent level curves of the local linear model norm. The solid ellipses represent* $\{s \in \boldsymbol{R}^n : \|F(x_k) + F'(x_k)s\| = \eta_k \|F(x_k)\|\}$. *The circular arcs represent trust-region boundaries. Each point* s_{min} *is the minimizer of the local linear model norm along the line joining* s_k^{CP} *and* s_k^{IN}.

We begin with two elementary propositions, the proofs of which are left to the reader. These are included primarily to show what might be involved in computing minimizing steps. In these, we define

$$s(\gamma) \equiv (1-\gamma)\hat{s}_k^{\mathrm{CP}} + \gamma s_k^{\mathrm{IN}}.$$

PROPOSITION 3.3. *If* $F'(x_k)(\hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}}) \neq 0$, *then* $\|F(x_k) + F'(x_k)s(\gamma)\|$ *is minimized at* $\gamma = \gamma_{\min}$, *given by*

$$(3.1) \quad \gamma_{\min} \equiv \frac{\langle F(x_k) + F'(x_k)\hat{s}_k^{\mathrm{CP}},\ F'(x_k)(\hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}})\rangle}{\|F'(x_k)(\hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}})\|^2} = \frac{\langle \hat{r}_k^{\mathrm{CP}},\ \hat{r}_k^{\mathrm{CP}} - r_k^{\mathrm{IN}}\rangle}{\|\hat{r}_k^{\mathrm{CP}} - r_k^{\mathrm{IN}}\|^2},$$

where $\hat{r}_k^{\mathrm{CP}} \equiv F(x_k) + F'(x_k)\hat{s}_k^{\mathrm{CP}}$ *and* $r_k^{\mathrm{IN}} \equiv F(x_k) + F'(x_k)s_k^{\mathrm{IN}}$.

One sees from (3.1) that, if $\hat{r}_k^{\mathrm{CP}}$ and r_k^{IN} are already available, then $\gamma_{\min}$ can be determined with only a modest amount of arithmetic. If these are not available, then one or, at most, two Jacobian-vector products may be required in addition to some arithmetic.

PROPOSITION 3.4. *If* $\|\hat{s}_k^{\mathrm{CP}}\| < \delta$ *and* $s_k^{\mathrm{IN}} \neq \hat{s}_k^{\mathrm{CP}}$, *then* $\|s(\gamma)\| = \delta$ *for exactly two values* $\gamma = \gamma_\delta^+$ *and* $\gamma = \gamma_\delta^-$, *given by*

$$(3.2) \quad \gamma_\delta^\pm = \frac{\langle \hat{s}_k^{\mathrm{CP}},\ \hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}}\rangle \pm \sqrt{\langle \hat{s}_k^{\mathrm{CP}},\ \hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}}\rangle^2 + (\delta^2 - \|\hat{s}_k^{\mathrm{CP}}\|^2)\|\hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}}\|^2}}{\|\hat{s}_k^{\mathrm{CP}} - s_k^{\mathrm{IN}}\|^2}.$$

One sees from (3.2) that $\gamma_\delta^\pm$ can be computed with a modest amount of arithmetic. Also, we have that $\gamma_\delta^- < 0 < \gamma_\delta^+$. Additionally, in the context of interest, viz., the final steps of Procedures 3.1 and 3.2, the assumption that $\|\hat{s}_k^{\mathrm{CP}}\| < \delta$ and $s_k^{\mathrm{IN}} \neq \hat{s}_k^{\mathrm{CP}}$ always holds. Indeed, in this context, we have $\|\hat{s}_k^{\mathrm{CP}}\| < \delta < \|s_k^{\mathrm{IN}}\|$ in Procedure 3.1 and both $\|\hat{s}_k^{\mathrm{CP}}\| < \delta$ and $\|F(x_k) + F'(x_k)s_k^{\mathrm{IN}}\| \leq \eta_k < \|F(x_k) + F'(x_k)\hat{s}_k^{\mathrm{CP}}\|$ in Procedure 3.2. Note that the first pair of inequalities implies that $\gamma_\delta^+ < 1$, and it follows from the final pair that $\gamma_{\min} > 0$.

The following are our refinements of Procedures 3.1 and 3.2. In these, $\gamma_{\min}$ and $\gamma_\delta^\pm$ are defined by (3.1) and (3.2), respectively. In Procedure 3.6, some simplification is allowed in the last step by the fact that $\gamma_{\min} > 0$ there, as noted above.

Procedure 3.5. Traditional strategy with model norm minimization

```
If $\|s_k^{\rm IN}\| \le \delta$, set $s_k = s_k^{\rm IN}$.
Else,
    If $\|\hat{s}_k^{\rm CP}\| \ge \delta$, set $s_k = (\delta/\|\hat{s}_k^{\rm CP}\|)\hat{s}_k^{\rm CP}$.
    Else, set $s_k = (1-\gamma)\hat{s}_k^{\rm CP} + \gamma s_k^{\rm IN}$ for $\gamma = \max\{\gamma_\delta^-, \min\{\gamma_{\min}, \gamma_\delta^+\}\}$.
```

Procedure 3.6. Alternative strategy with model norm minimization

```
If $\|\hat{s}_k^{\rm CP}\| \ge \delta$, set $s_k = (\delta/\|\hat{s}_k^{\rm CP}\|)\hat{s}_k^{\rm CP}$.
Else,
    If $\|F(x_k) + F'(x_k)\,\hat{s}_k^{\rm CP}\| \le \eta_k \|F(x_k)\|$, set $s_k = \hat{s}_k^{\rm CP}$.
    Else, set $s_k = (1-\gamma)\hat{s}_k^{\rm CP} + \gamma s_k^{\rm IN}$ for $\gamma = \min\{\gamma_{\min}, \gamma_\delta^+\}$.
```

Note that, in the circumstance illustrated on the left in Figure 3.1, both Procedure 3.5 and Procedure 3.6 would yield $s_k = s_{min}$. However, in the circumstance illustrated on the right, Procedure 3.6 would produce $s_k = s_{min}$, but Procedure 3.5 would result in $s_k = s_k^{\rm IN}$ because $s_k^{\rm CP}$ would not be computed in this case.

It is observed in section 3.1 that, in a Newton-iterative implementation of Procedure 3.2, it may be advantageous to use $\hat{s}_k^{\rm CP}$ as the initial approximate solution in the iterative solver when it is necessary to compute $s_k^{\rm IN}$. If this is done and if the iterative method is unrestarted GMRES or a mathematically equivalent method, then the minimization of Procedure 3.6 would be unneccesary. Indeed, Procedure 3.2 would produce the same step as Procedure 3.6, since the GMRES step from $\hat{s}_k^{\rm CP}$ to $s_k^{\rm IN}$ already minimizes the local linear model norm over the Krylov subspace in which it lies and, in particular, minimizes it along the line joining $\hat{s}_k^{\rm CP}$ and $s_k^{\rm IN}$.

We also note an additional possibility with Procedures 3.2 and 3.6: If $\|\hat{s}_k^{\rm CP}\| < \delta$ and $\|F(x_k) + F'(x_k)\,\hat{s}_k^{\rm CP}\| \le \eta_k\|F(x_k)\|$, then, rather than take $s_k = \hat{s}_k^{\rm CP}$, one can reduce η_k so that $\eta_k < \|F(x_k) + F'(x_k)\,\hat{s}_k^{\rm CP}\|$ and compute $s_k^{\rm IN}$ for the reduced η_k. With this $s_k^{\rm IN}$, the final steps of Procedures 3.2 and 3.6 should yield steps that further reduce the local linear model norm within the trust region.

4. Numerical experiments.

In this section, we report on numerical experiments in which a Newton–GMRES implementation of Algorithm INDL employing step-selection strategies from section 3 and certain other options was applied to several well-known benchmark problems involving the steady-state Navier–Stokes equations. In the following, we outline details of the implementation and the overall computational environment, describe the test problems, and give the results of the experiments. These experiments involve challenging large-scale problems and advanced computing platforms but are somewhat limited in scope. Our primary goal is to demonstrate the basic effectiveness of Algorithm INDL with the selected strategies and options on these test problems, and not to provide a definitive comparison of all algorithmic possibilities on a broad variety of applications.

4.1. The implementation and the computational environment.

In our implementation of Algorithm INDL, each inexact Newton step $s_k^{\rm IN}$ was determined using restarted GMRES with a restart value of 200. Up to three restarts (600 total GMRES iterations) were allowed for each k. The preconditioner was an additive Schwarz preconditioner with an ILUT [36] factorization and solve on each subdomain. With this preconditioner, GMRES almost never restarted in our experiments. For each k, the initial approximate solution in GMRES was either zero or, optionally, the approximate Cauchy point $\hat{s}_k^{\rm CP}$ determined by $\hat{d}_k = -\nabla f(x_k)$, which could be evaluated analytically by our codes and for which condition (2.3) of Theorem 2.1 holds for $0 < \epsilon \le \kappa_2(D)^{-1/2}$ (see Remark 2 following Theorem 2.1). Products of the Jacobian with vectors needed by GMRES were also evaluated analytically.

We used two choices for each forcing term η_k in our experiments: a small constant choice $\eta_k = 10^{-4}$, which should result in fast linear convergence in a neighborhood of a solution; and an adaptive choice from [13], called "Choice 1" there and given by

$$(4.1)\quad \eta_k = \min\left\{ \eta_{\max}, \frac{\Big| \|F(x_k)\| - \|F(x_{k-1}) + F'(x_{k-1})\, s_{k-1}\| \Big|}{\|F(x_{k-1})\|} \right\}, \ k = 1, 2, \ldots,$$

which should yield a certain superlinear convergence near a solution [13, Th. 2.2]. As in [13], (4.1) was followed with the safeguard

$$\eta_k \leftarrow \max\{\eta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\} \quad \text{whenever } \eta_{k-1}^{(1+\sqrt{5})/2} > 0.1$$

to prevent the forcing terms from becoming too small too quickly away from a solution. In our implementation, we used $\eta_0 = .01$ and $\eta_{\max} = .9$.

Dogleg steps were selected using the standard and alternative strategies described in Procedures 3.1 and 3.2, respectively. Since, as noted previously, GMRES rarely restarted, Procedure 3.6 almost always produced the same steps as Procedure 3.2 when GMRES was started from the approximate Cauchy point. Thus there seemed to be no significant advantage in using Procedure 3.6 rather than Procedure 3.2 in this case. For consistency, then, we used Procedures 3.1 and 3.2 rather than Procedures 3.5 and 3.6 throughout.

In the test $\mathit{ared}_k < t \cdot \mathit{pred}_k$ at the top of the while-loop, we took $t = 10^{-4}$, so that even very modest agreement would result in accepting the step. Within the while-loop, we used the fixed choice $\theta = .25$ for each update $\delta \leftarrow \max\{\theta\delta, \delta_{\min}\}$. In updating δ in $[\delta_{\min}, \infty)$ following the while-loop, we used a procedure similar to that outlined in [11], in which the trust region is shrunk (subject to the constraint $\delta \ge \delta_{\min}$) if $\|F(x_k + s_k)\|$ and $\|F(x_k) + F'(x_k)\, s_k\|$ do not agree well, expanded if they agree especially well, and left unchanged otherwise. The specific procedure, in which $0 < \rho_{\rm s} < \rho_{\rm e} < 1$, $0 < \beta_{\rm s} < 1 < \beta_{\rm e}$, and $\delta_{\max} > \delta_{\min}$, is as follows:

Procedure 4.1.

```
If $\mathit{ared}_k/\mathit{pred}_k < \rho_{\rm s}$,
    If $\|s_k^{\rm IN}\| < \delta$, update $\delta \leftarrow \max\{\|s_k^{\rm IN}\|, \delta_{\min}\}$.
    Else, update $\delta \leftarrow \max\{\beta_{\rm s}\delta, \delta_{\min}\}$.
Else,
    If $\mathit{ared}_k/\mathit{pred}_k > \rho_{\rm e}$ and $\|s_k\| = \delta$, update $\delta \leftarrow \min\{\beta_{\rm e}\delta, \delta_{\max}\}$.
```

In our implementation, we took $\rho_{\rm s} = 0.1$, $\rho_{\rm e} = 0.75$, $\beta_{\rm s} = .25$, $\beta_{\rm e} = 4.0$, $\delta_{\min} = 10^{-6}$, and $\delta_{\max} = 10^{10}$. The initial δ was determined after the computation of $s_0^{\rm IN}$ as follows: If $\|s_0^{\rm IN}\| < \delta_{\min}$, then $\delta = 2\delta_{\min}$; otherwise, $\delta = \|s_0^{\rm IN}\|$.

Our implementation of Algorithm INDL was done within the NOX nonlinear solver package [26], a C++ object-oriented library developed at Sandia National Laboratories for the efficient solution of systems of nonlinear equations. The GMRES implementation and preconditioners were provided by the AztecOO package [22], an extension of the Aztec library [47], which provides an extensive suite of Krylov solvers and preconditioners for the parallel solution of linear systems. The parallel finite-element reacting-flow code MPSalsa [40] was used to set up the finite-element discretization of the test problems described in section 4.2 and also to invoke the solvers. MPSalsa uses Chaco [21], a general graph partitioning tool, to partition the FE mesh into subdomains and assign subdomains to processors. Successful termination of the nonlinear solution algorithm was declared if both of the following held: (1) $\frac{1}{n}\|Ws_k\| < 1$, where n is the total number of unknowns and W is a diagonal weighting matrix with entries $W_{ii} = 1/(\varepsilon_r|x_k^{(i)}| + \varepsilon_a)$, in which $x_k^{(i)}$ is the ith component of x_k (see [1]); (2) $\|F(x_k)\| \le \epsilon_F \|F(x_0)\|$. In our tests, $\varepsilon_r = 10^{-3}$, $\varepsilon_a = 10^{-8}$, and $\epsilon_F = 10^{-2}$. The first criterion is employed by MPSalsa and, in our experiments, was typically more stringent than the second and necessary to ensure that finer physical details of solutions are adequately resolved.

Our experiments were performed on an IBM cluster at Sandia National Laboratories having 16 nodes, with each node containing two one-GHz Pentium III processors with one GB of RAM each. Tests on the 2D problems were done using four nodes (eight processors); tests on the 3D problems were done using 15 nodes (30 processors).

4.2. The test problems. The three benchmark problems considered here have been widely studied. We describe these in only qualitative terms and refer the reader to [30] and [41] for more specific descriptions of the governing partial differential equations, boundary conditions, and discretizations used in these experiments.

The problems are flow problems involving the steady-state Navier–Stokes equations for low-speed, incompressible flow, specifically the equations for momentum transport, total mass conservation, and, in one case, thermal energy transport. The unknowns are the fluid velocity vector, the hydrodynamic pressure, and, in the one case, the temperature. See [30] for precise formulations of the equations and the boundary conditions in each case. In each of our experiments, an algebraic system of the form (1.1) was obtained from the PDEs using a stabilized finite-element formulation following [23] and [45]. The Jacobian of the system is nonsymmetric and, when needed, was evaluated analytically in our tests.

Our first test problem is the thermal-convection problem [8]. This involves modelling the flow of fluid in a differentially heated square box in the presence of gravity. The PDEs are the equations for momentum and thermal energy transport and for mass conservation. The unknowns include the temperature as well as the fluid velocity and pressure. The temperature is held fixed at different values on two opposite faces of the box, with zero heat flux imposed on the remaining sides. The fluid velocity is zero on all sides. When suitably nondimensionalized, the equations involve two parameters: the Rayleigh number Ra and the Prandtl number Pr. As Ra increases for fixed Pr, the nonlinear effects of the convection terms increase and the solution becomes increasingly difficult to obtain. In our experiments, we took Pr = 1 and varied Ra. We considered 2D and 3D forms of this problem. The 2D problem was discretized on a 100×100 equally spaced mesh, which resulted in 40,804 unknowns for the discretized problem. In 3D, the discretization was on a $32 \times 32 \times 32$ equally spaced mesh, resulting in 179,685 unknowns.

Our second test problem is the lid-driven cavity problem [18], [39]. This requires simulating flow in the unit square in $\mathbb{R}^2$ or the unit cube in $\mathbb{R}^3$ driven by a moving upper boundary. The PDEs are the momentum transport and mass conservation equations; the unknowns are the fluid velocity and pressure. The fluid velocity on the top side is fixed at a nonzero value in the x-axis direction and held at zero on all other sides. An appropriately nondimensionalized form of the PDEs leads to the Reynolds number Re, a nondimensional parameter expressing the ratio of convective to diffusive momentum transport. As Re increases, the nonlinear components of the equations become more dominant and the problem becomes more difficult. We considered this problem in both 2D and 3D. The 2D computations were done on a 100×100 equally spaced grid, which led to 30,603 unknowns for the discretized problem. In 3D, the discretization was a $32 \times 32 \times 32$ equally spaced grid, yielding 143,748 unknowns.

Our third test problem is the backward-facing step problem [17]. This involves the simulation of flow through a rectangular channel that is initially constricted and subsequently expands over a reentrant backward-facing step. As above, the PDEs are those for momentum transport and mass conservation, and the unknowns are the fluid velocity and pressure. The nondimensionalized form of the equations is again parametrized by the Reynolds number Re, with problem difficulty increasing as Re increases. We considered this problem only in 2D. The flow was computed only in the expanded portion of the channel; flow entering from the constricted portion was simulated by imposing a parabolic velocity profile in the upper half of the inlet boundary and zero velocity on the lower half. See [30] for further details of the domain and boundary conditions. The problem was discretized on a 20×400 unequally spaced mesh (with a finer mesh near the step), which resulted in 25,263 unknowns.

4.3. The test results. In our experiments, we first conducted a robustness study. In this, we applied our implementation of Algorithm INDL with the algorithmic options described in section 4.1 to the test problems in section 4.2 and tabulated the numbers of failures with different options as problem parameters varied. The specific parameter values considered are as follows:

2D and 3D Thermal Convection	$\mathrm{Ra} = 10^3, 10^4, 10^5, 10^6$,
2D Lid-Driven Cavity	$\mathrm{Re} = 1,000, 2,000, \ldots, 10,000$,
3D Lid-Driven Cavity	$\mathrm{Re} = 100, 200, \ldots, 1,000$,
2D Backward-Facing Step	$\mathrm{Re} = 100, 200, \ldots, 700, 750, 800$.

The results of the robustness study are given in Table 4.1. For comparison, the table also includes results for NOX Newton–GMRES implementations with no globalization and with a backtracking globalization from [12] (see also [41], [31], [30]); these results were obtained in an earlier study involving identical test conditions (see [29, Tables 2–3]). Table 4.1 shows that both the dogleg and backtracking globalizations significantly improved robustness in these experiments, reducing the total numbers of failures with both forcing terms by 69% to 77%, compared to the method without globalization. Improvement was especially pronounced when adaptive forcing terms were used, a result consistent with other studies of globalized Newton–GMRES methods [31], [41], [30]. As also observed elsewhere [3], [41, section 5], [30], the dogleg and backtracking globalizations performed fairly comparably overall on these test problems, although there were some differences in the numbers of failures and where failures occurred. In view of the limited scope of these tests, we feel that further testing is needed to assess more fully the comparative robustness of these methods.

We next conducted an efficiency study of the dogleg methods, in which we compiled run times and other statistics for Algorithm INDL with different options on a selected set of test problem cases. This set included all cases considered in the robustness study in which all of the dogleg methods succeeded. Additionally, since all dogleg methods using the small constant forcing terms failed on the 2D lid-driven cavity problem for each value of Re above $1,000$, we included cases of this problem with $100 \leq \mathrm{Re} \leq 1,000$. The specific cases considered are as follows:

Table 4.1

Robustness study: For each method, the upper and lower rows to the right show the numbers of failures with, respectively, the adaptive and small constant forcing terms. "TC," "LDC," and "BFS" denote, respectively, the thermal-convection, lid-driven cavity, and backward-facing step problems. **GMRES solves starting from zero.* ***GMRES solves starting from the approximate Cauchy point.*

Method	2D problems			3D problems		Failure totals	
	TC	LDC	BFS	TC	LDC		
Algorithm INDL, Procedure 3.1	0	0	0	0	0	0	10
	1	9	0	0	0	10	
Algorithm INDL, Procedure 3.2*	0	2	0	0	0	2	12
	0	9	0	0	1	10	
Algorithm INDL, Procedure 3.2**	0	0	0	0	0	0	9
	0	9	0	0	0	9	
Backtracking globalization	0	0	0	0	0	0	10
	0	9	1	0	0	10	
No globalization	1	9	5	1	0	16	39
	1	10	7	1	4	23	

2D Thermal Convection	$\mathrm{Ra} = 10^3,\ 10^4,\ 10^5,$
3D Thermal Convection	$\mathrm{Ra} = 10^3,\ 10^4,\ 10^5,\ 10^6,$
2D Lid-Driven Cavity	$\mathrm{Re} = 100,\ 200,\ \ldots,\ 1{,}000,$
3D Lid-Driven Cavity	$\mathrm{Re} = 100,\ 200,\ \ldots,\ 900,$
2D Backward-Facing Step	$\mathrm{Re} = 100,\ 200,\ \ldots,\ 700,\ 750,\ 800.$

The results of the efficiency study are given in Table 4.2.[6] One sees that, as before, the forcing terms significantly influenced the performance of the methods. The adaptive forcing terms resulted in more inexact Newton steps on average than the more aggressive small constant forcing terms; however, the adaptive forcing terms led to significantly fewer GMRES iterations, both overall and per inexact Newton step. In the balance, the adaptive forcing terms yielded smaller run times. For a particular choice of forcing terms (adaptive or small constant), the methods performed fairly similarly, although when small constant forcing terms were used, the methods with Procedure 3.2 were somewhat less efficient in all statistics.

5. Concluding summary.

In the preceding, we provide a theoretical framework and practical strategies for implementing dogleg globalizations of general inexact Newton methods for solving a nonlinear system (1.1).

In section 2, we outline a very general inexact Newton dogleg method. In this, the dogleg curves are defined using general inexact Newton steps together with approximate steepest-descent directions and Cauchy points. To be admissible, trial steps along these curves must satisfy only mild minimum-length requirements in addition to being within trust regions. A convergence theorem for this method is given; this asserts that if the approximate steepest-descent directions are bounded away from

[6] See [30, Table 5.2] for a similar study comparing Algorithm INDL using Procedure 3.1 with backtracking and linesearch globalizations.

TABLE 4.2

Efficiency study: For each method, the upper and lower rows to the right show results with the adaptive and small constant forcing terms, respectively. All statistics are geometric means. Times are relative to that of Algorithm INDL *with Procedure* 3.1 *and small constant forcing terms. "INS" stands for "Inexact Newton Steps."*

**GMRES solves starting from zero.* ***GMRES solves starting from the approximate Cauchy point.*

Method	Inexact Newton steps	Function evaluations	GMRES iterations	GMRES iterations per INS	Normalized time
Algorithm INDL, Procedure 3.1	18.82	21.11	1160	61.65	0.87
	10.30	12.45	1330	129.2	1.00
Algorithm INDL, Procedure 3.2*	16.62	18.87	1200	72.21	0.92
	11.27	13.71	1650	146.4	1.21
Algorithm INDL, Procedure 3.2**	18.40	20.57	1122	60.99	0.88
	11.15	13.55	1634	146.6	1.23

orthogonality with the exact steepest-descent directions, then every limit point of a sequence of iterates generated by the method is a stationary point of $\|F\|$. Moreover, if there is a limit point at which the Jacobian of F is nonsingular, then that point must be a solution of (1.1) and the iterates must converge to it. In this case, initial trial steps are ultimately accepted without modification by the method, and the inexact Newton steps are ultimately admissible trial steps.

In section 3, we discuss strategies for selecting trial steps along the dogleg curves. It is noted that the step-selection strategy traditionally used in the exact-Newton case may exhibit certain shortcomings, principally that, when the forcing term is not small, this strategy may select the inexact Newton step even though a step biased toward the approximate Cauchy point may give significantly greater linear model norm reduction within the trust region. An alternative strategy is proposed that avoids this shortcoming. As a refinement of these strategies, simple modifications are introduced that further determine each dogleg step to minimize the local linear model norm along the dogleg curve within the trust region, provided both the approximate Cauchy point and the inexact Newton step have been computed.

In section 4, we report on numerical experiments in which a Newton–GMRES implementation of the general inexact Newton dogleg method in section 2 using the step-selection strategies in section 3 was applied to several two- and three-dimensional benchmark problems involving the steady-state Navier–Stokes equations. Although somewhat limited in scope, these experiments demonstrate the robustness of the method with these step-selection strategies and other algorithmic options on these challenging problems. They also provide an indication of the relative efficiency of these strategies and options. The use of adaptive forcing terms resulted in significantly greater robustness and some improvement in efficiency compared to the constant forcing term strategy; these results are consistent with previous studies of globalized inexact Newton methods. Otherwise, these experiments do not show dramatic differences in performance among the strategies and options that we tested. We feel that this may reflect the particular nature of the limited test set and also the rather modest problem sizes; there may be more pronounced differences in performance on other, broader test sets and higher-resolution problems. A more comprehensive numerical study will be the subject of future work.

Acknowledgment. The authors thank the anonymous referees for their thoughtful reviews, which led to significant improvements in this paper.

REFERENCES

[1] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-value problems in differential-algebraic equations*, Classics Appl. Math. 14, SIAM, Philadelphia, 1996.

[2] P. N. Brown, *A local convergence theory for combined inexact-Newton/finite-difference projection methods*, SIAM J. Numer. Anal., 24 (1987), pp. 407–434.

[3] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[4] P. N. Brown and Y. Saad, *Convergence theory of nonlinear Newton–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

[5] P. N. Brown, H. F. Walker, R. Wasyk, and C. S. Woodward, *On using approximate finite differences in matrix-free Newton–Krylov methods*, SIAM J. Numer. Anal., 46 (2008), pp. 1892–1911.

[6] E. Catinas, *Inexact perturbed Newton methods and applications to a class of Krylov solvers*, J. Optim. Theory Appl., 108 (2001), pp. 543–570.

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS-SIAM Ser. Optim., SIAM, Philadelphia, 2000.

[8] G. De Vahl Davis and C. P. Jones, *Natural convection in a square cavity: A comparison exercise*, Int. J. Numer. Methods Fluids, 3 (1983), pp. 227–248.

[9] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[10] R. S. Dembo and T. Steihaug, *Truncated Newton algorithms for large-scale optimization*, Math. Programming, 26 (1983), pp. 190–212.

[11] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Ser. Comput. Math., Prentice-Hall, Englewood Cliffs, NJ, 1983.

[12] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[13] S. C. Eisenstat and H. F. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[14] M. El Hallabi, *A Global Convergence Theory for Arbitrary Norm Trust-Region Methods for Nonlinear Equations*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1987.

[15] M. El Hallabi and R. A. Tapia, *A Global Convergence Theory for Arbitrary Norm Trust-Region Methods for Nonlinear Equations*, Technical report TR87-25, Department of Mathematical Sciences, Rice University, Houston, TX, 1987, revised 1989.

[16] R. W. Freund, G. H. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, Acta Numer., 1 (1992), pp. 57–100.

[17] D. K. Gartling, *A test problem for outflow boundary conditions–flow over a backward facing step*, Internat. J. Numer. Methods Fluids, 11 (1990), pp. 953–967.

[18] U. Ghia, K. N. Ghia, and C. T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys., 48 (1982), pp. 387–411.

[19] G. H. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[20] M. H. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations*, Acta Numer., 6 (1997), pp. 271–397.

[21] B. Hendrickson and R. Leland, *The Chaco User's Guide–Version* 1.0, Technical report Sand93-2339, Sandia National Laboratories, Albuquerque NM, 1993.

[22] M. Heroux, *AztecOO: Object-Oriented Aztec Linear Solver Package*, available online at http://software.sandia.gov/trilinos/packages/aztecoo/index.html.

[23] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert, *A new finite element formulation for computational fluid dynamics:* VII. *The Galerkin/least-squares method for advective-diffusive equations*, Comput. Methods Appl. Mech. Engrg., 73 (1989), pp. 173–189.

[24] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Fundam. Algorithms, SIAM, Philadelphia, 2003.

[25] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton–Krylov methods: A survey of approaches*

and applications, J. Comput. Phys., 193 (2004), pp. 357–397.

[26] T. G. Kolda and R. P. Pawlowski, *NOX Nonlinear Solver Project*, available online at http://software.sandia.gov/nox.

[27] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[28] S. G. Nash, *Truncated Newton Methods*, Ph.D. thesis, Computer Science Department, Stanford University, Palo Alto, CA, 1982.

[29] R. P. Pawlowski, J. N. Shadid, J. P. Simonis, and H. F. Walker, *Globalization Techniques for Newton–Krylov Methods and Applications to the Fully-Coupled Solution of the Navier–Stokes Equations*, Technical report Sand2004-1777, Sandia National Laboratories, Albuquerque NM, 2004.

[30] R. P. Pawlowski, J. N. Shadid, J. P. Simonis, and H. F. Walker, *Globalization techniques for Newton–Krylov methods and applications to the fully-coupled solution of the Navier–Stokes equations*, SIAM Rev., 48 (2006), pp. 700–721.

[31] M. Pernice and H. F. Walker, *NITSOL: A Newton iterative solver for nonlinear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 302–318.

[32] M. J. D. Powell, *A FORTRAN Subroutine for Unconstrained Minimization, Requiring First Derivatives of the Objective Function*, Technical report AERE-R. 6469, Mathematics Branch, A.E.R.E. Harwell, Berkshire, England, 1970.

[33] M. J. D. Powell, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Gordon and Breach, London, 1970, pp. 87–114.

[34] M. J. D. Powell, *General algorithms for discrete nonlinear approximation calculations*, in Approximation Theory IV, C. K. Chui, L. L. Schumaker, and J. D. Ward, eds., Academic Press, New York, 1983, pp. 187–218.

[35] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.

[36] Y. Saad, *ILUT: A dual threshold incomplete ILU factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.

[37] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

[38] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual method for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[39] R. Schreiber and H. B. Keller, *Driven cavity flows by efficient numerical techniques*, J. Comput. Phys., 49 (1983), pp. 310–333.

[40] J. N. Shadid, H. K. Moffat, S. A. Hutchinson, G. L. Hennigan, K. D. Devine, and A. G. Salinger, *MPSalsa: A Finite Element Computer Program for Reacting Flow Problems Part* 1: *Theoretical Development*, Technical report Sand95-2752, Sandia National Laboratories, Albuquerque NM, 1996.

[41] J. N. Shadid, R. S. Tuminaro, and H. F. Walker, *An inexact Newton method for fully coupled solution of the Navier–Stokes equations with heat and mass transport*, J. Comput. Phys., 137 (1997), pp. 155–185.

[42] J. N. Shadid, R. S. Tuminaro, and H. F. Walker, *On backtracking failure in Newton–GMRES methods with a demonstration for the Navier–Stokes equations*, J. Comput. Phys., 180 (2002), pp. 549–558.

[43] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

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

[45] T. E. Tezduyar, *Stabilized finite element formulations for incompressible flow computations*, Adv. App. Mech., 28 (1992), pp. 1–44.

[46] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 57–87.

[47] R. S. Tuminaro, M. Heroux, S. A. Hutchinson, and J. N. Shadid, *Aztec User's Guide–Version* 2.1, Technical report Sand99-8801J, Sandia National Laboratories, Albuquerque NM, 1999.

[48] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monogr. Appl. Comput. Math. 13, Cambridge University Press, Cambridge, 2003.

[49] C. S. Woodward, *private communication*, 2006.