SAND REPORT

SAND2004-1777
Unlimited Release
Printed May 2004

Globalization techniques for Newton–Krylov methods and applications to the fully-coupled solution of the Navier–Stokes equations

Roger P. Pawlowski, John N. Shadid, Joseph P. Simonis, and Homer F. Walker

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Of£ce of Scienti£c and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Spring£eld, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/ordering.htm

SAND2004-1777
Unlimited Release
Printed May 2004

Globalization techniques for Newton–Krylov methods and applications to the fully-coupled solution of the Navier–Stokes equations

Roger P. Pawlowski and John N. Shadid
Computational Sciences Department
Sandia National Laboratories
P.O. Box 5800, MS-0316
Albuquerque, NM 87185-0316

Joseph P. Simonis and Homer F. Walker
Department of Mathematical Sciences
Worcester Polytechnic Institute
Worcester, MA 01609-2280

September 18, 2005

Abstract

A Newton–Krylov method is an implementation of Newton's method in which a Krylov subspace method is used to solve approximately the linear subproblems that determine Newton steps. To enhance robustness when good initial approximate solutions are not available, these methods are usually *globalized*, i.e., augmented with auxiliary procedures (*globalizations*) that improve the likelihood of convergence from a poor starting point. In recent years, globalized Newton–Krylov methods have been used increasingly for the fully-coupled solution of large-scale CFD problems. In this paper, we review several representative globalizations, discuss their properties, and report on a numerical study aimed at evaluating their relative merits on large-scale 2D and 3D problems involving the steady-state Navier–Stokes equations.

Acknowledgements

The authors would like to acknowledge Tamara Kolda for many discussions on nonlinear algorithms design and working with the NOX library, Mike Heroux in supporting our work with the Epetra and AztecOO libraries, Paul Lin in working with MPSalsa, and helpful discussions with Jorge Moré.

This work was partially supported by the DOE Office of Science MICS Program and the DOE NNSA ASCI Program.

The work of Joseph Simonis and Homer Walker was supported in part by Sandia National Laboratories under the ASCI program and in part by the Sandia National Laboratories Computer Science Research Institute through contracts 16099 and 198313. Walker's work was done in part during a sabbatical visit to Sandia National Laboratories; he gratefully acknowledges support for this through contract 198313. Walker's work was also supported in part by the University of Utah Center for the Simulation of Accidental Fires and Explosions (C-SAFE), funded by the Department of Energy, Lawrence Livermore National Laboratory, under subcontract B524196.

Contents

Figures

Tables

1 Introduction

Krylov subspace methods constitute a broad and widely used class of iterative linear algebra methods that includes, most notably, the classical conjugate gradient method for symmetric positive-definite systems [26] and more recently developed methods for nonsymmetric linear systems such as GMRES[1] [42], which is of particular interest here, and also Bi-CGSTAB [53], TFQMR [17], and related methods. An extensive discussion of these methods is beyond the scope of this work; we refer the reader to the surveys [18] and [22] and the books [21], [41], and [54].

A *Newton–Krylov method* (see, e.g., [3], [29], [31]) is an implementation of Newton's method in which a Krylov subspace method is used to solve approximately the linear systems that characterize steps of Newton's method. Specifically, if we seek a zero of a residual function $F : \mathbb{R}^n \to \mathbb{R}^n$ and if $u \in \mathbb{R}^n$ is a current approximate solution, then a Krylov subspace method is applied to solve approximately the *Newton equation*

$$F'(u)s = -F(u), \tag{1.1}$$

where $F'(u) \in \mathbb{R}^{n\times n}$ is the Jacobian (matrix) of F at u. A Newton–Krylov method that uses a specific Krylov subspace method is often designated by appending the name of the method to "Newton," as in "Newton-GMRES" or "Newton–BiCGSTAB." (The term "truncated Newton method" is also widely used when the Krylov subspace method is the conjugate gradient method; cf. [11] and [35].) Krylov subspace methods have special advantages in the solution of (1.1). In particular, most of these methods, including those named above, require only products of $F'(u)$ with vectors[2] and thus allow "matrix-free" Newton–Krylov implementations, in which these products are evaluated or approximated without creating or storing $F'(u)$. (See, e.g, [31].)

A Newton–Krylov method is usually implemented as an *inexact Newton method* [10], the basic form of which is as follows:

Algorithm IN: Inexact Newton Method [10]

LET u_0 BE GIVEN.

FOR $k = 0,\ 1,\ \ldots$ (UNTIL CONVERGENCE) DO:

CHOOSE $\eta_k \in [0, 1)$ AND s_k SUCH THAT

$$\|F(u_k) + F'(u_k)\, s_k\| \le \eta_k \|F(u_k)\|. \tag{1.2}$$

SET $u_{k+1} = u_k + s_k$.

In the Newton–Krylov context, one chooses for each k a *forcing term* $\eta_k \in [0, 1)$ (cf. [14]) and then applies the Krylov subspace method until an iterate s_k satisfies the *inexact Newton*

[1] For convenience in the following, we usually do not distinguish between GMRES and its restarted version GMRES(m).

[2] Some Krylov subspace methods require products of $F'(u)^T$ as well.

condition (1.2). The forcing terms determine the local convergence properties of the method: by choosing $\{\eta_k\}$ appropriately, one can achieve desirably fast rates of local convergence, up to the rate of exact Newton's method (typically quadratic) [10]. Additionally, by reducing the likelihood of *oversolving*, i.e., obtaining unproductive accuracy in approximately solving (1.1), well-chosen forcing terms may significantly improve the efficiency of the method and, in some applications, the robustness as well ([14],[49], [52]).

Newton–Krylov methods, like all Newton-like methods, must usually be *globalized*, i.e., augmented with certain auxiliary procedures (*globalizations*) that increase the likelihood of convergence when good initial approximate solutions are not available. Globalizations are typically structured to test whether a step gives satisfactory progress toward a solution and, if necessary, to modify it to obtain a step that does give satisfactory progress. There are two major categories of globalizations[3]: *backtracking* (*line-search*, *damping*) methods, in which step lengths are adjusted (usually shortened) to obtain satisfactory steps; and *trust region* methods, in which a step from an approximate solution u is ideally chosen to minimize the norm of $F(u) + F'(u)s$, the *local linear model* of F, within a specified "trust region." (More specifically, the trust region step is ideally $\arg\min_{\|s\| \le \delta} \|F(u) + F'(u)s\|$, where $\delta > 0$ is the *trust region radius*.) Both backtracking and trust region methods have strong theoretical support; see, e.g., [12] and [13]. Backtracking methods are relatively easy to implement; however, each step direction is restricted to be that of the initial trial step, which may be a weak descent direction, especially if the Jacobian is ill-conditioned [52]. 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 may be problematic. In general, it is not feasible to compute the ideal trust region step accurately, and popular ways of approximating this step require products of the transpose of the Jacobian with vectors. These products may be difficult or impractical to compute in some applications, especially in the Newton–Krylov context, in which the Jacobian may not be known. Additionally, a step produced by a Newton–Krylov method (or any iterative method) may not be well suited for use in these popular approaches unless it solves (1.1) fairly accurately, and the necessary accuracy may be difficult to determine a priori. We comment further on these issues in §2.4.

The purpose of this paper is to review several representative globalizations of Newton–Krylov methods, discuss their properties, and report on extensive numerical experiments with Newton–GMRES implementations that demonstrate their relative merits in large-scale applications involving the steady-state Navier–Stokes equations. Our main goal is to provide an accessible introduction to the methods of interest and a thorough study of their performance on an important class of large-scale problems. We also offer pointers to publicly available, downloadable software implementations used in our tests and report new experimental data on the numerical solution of several 3D benchmark problems. This work is meant to be a cohesive study of a representative variety of practically effective techniques rather than an exhaustive survey. We do not cover other robust solution techniques such as homotopy, continuation, pseudo-transient continuation, or mesh sequencing methods but refer the reader to [56], [57], [1], [33], [7], [30], [31], and the references in those works.

[3]See [12, Ch. 6] for a general discussion of classical globalizations.

In an earlier study [49], we considered a backtracking globalization from [13] and showed in experiments that it significantly improves the robustness of a Newton-GMRES method on the applications of interest here, especially when combined with adaptively determined forcing terms from [14]. Here, we extend that study to include the backtracking globalization of [13] and [49], a certain refinement of that globalization, an alternative line-search procedure from [34], and a *dogleg* trust region implementation ([39], [12]). This dogleg implementation is feasible because our testing environment allows the computation of products of the transpose of the Jacobian with vectors. Further aspects of the implementation and associated issues are discussed in §2.4.

The test problems are 2D and 3D versions of three benchmark flow problems, viz., the thermal convection, backward-facing step, and lid driven cavity problems. (The 2D versions of these and the 3D backward-facing step problem were also used in [49].) These problems are all large-scale, with the number of equations and unknowns ranging from 25,263 to 1,042,236 in our tests; consequently, all numerical experiments were necessarily run on parallel computers, with the number of processors employed ranging from eight to 100. An important aspect of this study is describing the algorithmic features that were used, beyond the basic globalized Newton–Krylov methods, to make the implementations effective on these platforms in the problem regimes of interest.

In §2 below, we discuss the numerical methods of interest and outline their theoretical support. In §3, we introduce the governing PDEs and the discretized equations. In §4, the test problems and the computing environment are described. We summarize the test results in §5, comment further on failure and robustness in §6, and draw conclusions in §7. Proofs of some theorems and complete details of the test results are given in an appendix. Throughout the paper, the norm $\|\cdot\|$ is assumed to be an inner-product norm but is otherwise arbitrary.

2 The numerical methods

2.1 The forcing terms

Although the focus of this study is on globalization procedures, it has been seen in previous studies ([14],[49], [52]) that the forcing terms may affect the robustness of a Newton-Krylov method, globalization notwithstanding. Accordingly, we consider two choices of the forcing terms here to assess their effects on the globalizations of interest. The first is a small constant value, viz., $\eta_k = 10^{-4}$ for each k, which requires solving each instance of (1.1) with fairly high accuracy and should produce a close approximation of the exact Newton step. The second is an adaptive forcing term, called Choice 1 in [14] and determined as follows:

Select any $\eta_0 \in [0,1)$ and set

$$\eta_k = \frac{\Big|\|F(u_k)\| - \|F(u_{k-1}) + F'(u_{k-1})\, s_{k-1}\|\Big|}{\|F(u_{k-1})\|}, \qquad k = 1, 2, \ldots. \tag{2.1}$$

In keeping with practice elsewhere ([14], [49], [38]), we also follow (2.1) with the safeguard

$$\eta_k \leftarrow \min\Big\{\eta_{\max}, \max\{\eta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\}\Big\}, \quad \text{whenever } \eta_{k-1}^{(1+\sqrt{5})/2} > 0.1, \tag{2.2}$$

which is intended to prevent the forcing terms from becoming too small too quickly away from a solution and also to keep them below a prescribed $\eta_{\max} \in [0,1)$. (In our implementations, we used $\eta_0 = .01$ and $\eta_{\max} = .9$.) The exponent $(1+\sqrt{5})/2$ is related to a local convergence rate associated with these forcing terms; see the remark following Theorem 2.3 below.

To briefly state the local convergence properties of Algorithm IN with these two choices of the forcing terms, we formulate the following:

Assumption 2.1 (a) *$F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable in a neighborhood of $u_* \in \mathbb{R}^n$ such that $F(u_*) = 0$ and $F'(u_*)$ is nonsingular.*

(b) *F is Lipschitz continuously differentiable at u_*, i.e., there is a constant Γ for which $\|F'(u) - F'(u_*)\| \le \Gamma\|u - u_*\|$ for all u sufficiently near u_*.*

With this assumption, we have the results below from [10] and [14].

Theorem 2.2 ([10],Th. 2.3) *Suppose that Assumption* 2.1(a) *holds, and let $\{u_k\}$ be a sequence produced by Algorithm* IN *with $0 \le \eta_k \le \eta_* < 1$ for each k. If u_0 is sufficiently near u_*, then $\{u_k\}$ converges to u_* and, provided $u_k \ne u_*$ for all k,*

$$\limsup_{k\to\infty} \|u_{k+1} - u_*\|_* / \|u_k - u_*\|_* \le \eta_*, \tag{2.3}$$

where $\|v\|_* \equiv \|F'(u_*)v\|$ for each $v \in \mathbb{R}^n$.

It follows that if $\eta_k = 10^{-4}$ for each k, then, under Assumption 2.1(a), Algorithm IN exhibits fast local q-linear convergence[4] to a solution u_*; specifically, (2.3) holds with $\eta_* = 10^{-4}$.

Theorem 2.3 ([14], Th. 2.2) *Suppose that Assumption* 2.1 *holds, and let $\{u_k\}$ be a sequence produced by Algorithm* IN *with each η_k given by* (2.1). *If u_0 is sufficiently near u_*, then $\{u_k\}$ converges to u_* with*

$$\|u_{k+1} - u_*\| \le \gamma\, \|u_k - u_*\|\, \|u_{k-1} - u_*\|, \qquad k = 1, 2, \ldots \tag{2.4}$$

for a constant γ independent of k.

[4]For definitions of the kinds of convergence referred to here, see [12].

As observed in [14], it follows from (2.4) that the convergence is q-superlinear, two-step q-quadratic, and of r-order $(1+\sqrt{5})/2$. Also, the conclusions of the theorem still hold if each η_k is determined by (2.1) followed by the safeguard (2.2).

2.2 The backtracking methods

We consider here the following general backtracking method from [13].

Algorithm INB: Inexact Newton Backtracking Method [13]

LET u_0, $\eta_{\max} \in [0,1)$, $t \in (0,1)$, AND $0 < \theta_{\min} < \theta_{\max} < 1$ BE GIVEN.

FOR $k = 0,\ 1,\ \ldots$ (UNTIL CONVERGENCE) DO:

CHOOSE INITIAL $\eta_k \in [0, \eta_{\max}]$ AND s_k SUCH THAT
$$\|F(u_k) + F'(u_k)\, s_k\| \le \eta_k \|F(u_k)\|.$$

WHILE $\|F(u_k + s_k)\| > [1 - t(1-\eta_k)]\, \|F(u_k)\|$ DO:

CHOOSE $\theta \in [\theta_{\min}, \theta_{\max}]$.

UPDATE $s_k \leftarrow \theta s_k$ AND $\eta_k \leftarrow 1 - \theta(1-\eta_k)$.

SET $u_{k+1} = u_k + s_k$.

In Algorithm INB, the backtracking globalization resides in the while-loop, in which steps are tested and shortened as necessary until the acceptability condition

$$\|F(u_k + s_k)\| \le [1 - t(1-\eta_k)]\, \|F(u_k)\| \tag{2.5}$$

holds. As noted in [13], if F is continuously differentiable, then this globalization produces a step for which (2.5) holds after a finite number of passes through the while-loop; furthermore, the inexact Newton condition (1.2) still holds for the final s_k and η_k. The condition (2.5) is a "sufficient decrease" condition on $\|F(u_k + s_k)\|$. To illuminate it further, we follow [13] and [49] and define

$$\begin{aligned} ared_k &\equiv \|F(u_k)\| - \|F(u_k + s_k)\|, \\ pred_k &\equiv \|F(u_k)\| - \|F(u_k) + F'(u_k)\, s_k\|, \end{aligned} \tag{2.6}$$

respectively, the *actual reduction* in $\|F\|$ and the *predicted reduction* given by the local linear model. It is easily verified that (1.2) is equivalent to $pred_k \ge (1-\eta_k)\|F(u_k)\|$ and (2.5) is equivalent to $ared_k \ge t(1-\eta_k)\|F(u_k)\|$. Thus, if (1.2) requires the predicted reduction to be at least $(1-\eta_k)\|F(u_k)\|$, then (2.5) requires the actual reduction to be at least the fraction t of that amount. In our implementation, we used $t = 10^{-4}$ so that, consistent with recommendations in [12], only a very modest decrease in $\|F\|$ is required for a step to be accepted.

Theoretical support for Algorithm INB is provided in [13] by the following result.

Theorem 2.4 ([13], Th. 6.1) *Assume that F is continuously differentiable. If $\{u_k\}$ produced by Algorithm* INB *has a limit point u_* such that $F'(u_*)$ is nonsingular, then $F(u_*) = 0$ and $u_k \to u_*$. Furthermore, the initial s_k and η_k are accepted without modification in the while-loop for all sufficiently large k.*

A consequence of the theorem is that if $\{u_k\}$ converges to a solution u_* such that $F'(u_*)$ is nonsingular, then, under Assumption 2.1, the convergence is ultimately governed by the initial η_k's. In particular, if $\eta_k = 10^{-4}$ for each k, then (2.3) holds with $\eta_* = 10^{-4}$, and if each η_k is given by (2.1) followed by (2.2), then an inequality (2.4) holds for sufficiently large k and a γ independent of k.

Restricting each step-length reduction factor θ to lie in $[\theta_{\min}, \theta_{\max}]$ is known as *safeguarded backtracking*. In our implementation, we used the common choices $\theta_{\min} = 1/10$ and $\theta_{\max} = 1/2$ (cf. [12]). We also followed standard practice in choosing $\theta \in [\theta_{\min}, \theta_{\max}]$ to minimize a low-degree polynomial that interpolates values of $\|F\|$. Specifically, in this study, we tested two possibilities: The first is that used in [49], viz., to determine $\theta \in [\theta_{\min}, \theta_{\max}]$ to minimize a quadratic polynomial $p(t)$ that satisfies $p(0) = \frac{1}{2}\|F(u_k)\|^2$, $p(1) = \frac{1}{2}\|F(u_k + s_k)\|^2$, and $p'(0) = \frac{d}{dt}\frac{1}{2}\|F(u_k + ts_k)\|^2\Big|_{t=0}$. The second is a refinement of this idea from [12], as follows: On the first step-length reduction, θ is chosen to minimize an interpolating quadratic polynomial as before. On subsequent reductions, θ is chosen to minimize over $[\theta_{\min}, \theta_{\max}]$ a *cubic* polynomial $p(t)$ for which $p(0)$, $p(1)$, and $p'(0)$ have the same values as before and additionally $p(\theta_{\rm prev}^{-1}) = \frac{1}{2}\|F(u_k + \theta_{\rm prev}^{-1} s_k)\|^2$, where $\theta_{\rm prev}$ is the step-length reduction factor used in the previous reduction and $\|F(u_k + \theta_{\rm prev}^{-1} s_k)\|$ has been retained from that reduction. Formulas for the minimizers of these polynomials are given in [12, Ch. 6].

2.3 The Moré–Thuente line-search method

This line-search procedure from [34] is intended for the unconstrained minimization of a general functional $f : \mathbb{R}^n \to \mathbb{R}^1$ and is adapted to the present setting by taking $f(u) \equiv \frac{1}{2}\|F(u)\|^2$. It differs from the backtracking methods of §2.2 primarily in being more directly focused on approximately minimizing $\|F\|$ in a given search direction and by allowing steps that are longer than the initial trial step if such steps seem warranted by potential further decrease in $\|F\|$.

To set the context, we formulate an inexact Newton method incorporating this line search as follows:

Algorithm INMTL: Inexact Newton Moré–Thuente Line-Search Method

LET u_0 AND $\eta_{\max} \in [0, 1)$ BE GIVEN.

FOR $k = 0,\ 1,\ \ldots$ (UNTIL CONVERGENCE) DO:

CHOOSE $\eta_k \in [0, \eta_{\max}]$ AND INITIAL s_k SUCH THAT

$$\|F(u_k) + F'(u_k)\, s_k\| \le \eta_k \|F(u_k)\|.$$

Apply the Moré–Thuente line search [34] to

determine a final s_k.

Set $u_{k+1} = u_k + s_k$.

To describe the Moré–Thuente line search, we define for a particular k

$$\phi(\lambda) \equiv f(u_k + \lambda s_k) = \frac{1}{2}\|F(u_k + \lambda s_k)\|^2. \tag{2.7}$$

We assume for this discussion that F is continuously differentiable, which implies that f and ϕ are as well. Since the initial s_k is an inexact Newton step from u_k, it is also a *descent direction* for $\|\cdot\|$ and f in the sense that $\phi'(0) < 0$ ([4, Prop. 3.3], [13, Lem. 7.1]).

The goal of the line search is to find a $\lambda > 0$ satisfying the two inequalities

$$\phi(\lambda) \le \phi(0) + \alpha\phi'(0)\lambda, \tag{2.8}$$

$$|\phi'(\lambda)| \le \beta|\phi'(0)|, \tag{2.9}$$

where α and β are given parameters in $(0, 1)$. Once such a λ has been determined by the line search, the initial s_k is updated by $s_k \leftarrow \lambda s_k$ to determine the final s_k. Inequalities (2.8) and (2.9) are sometimes known as the *(strong) Wolfe conditions* [34, Ch. 3]. (The weak Wolfe conditions consist of (2.8) and the inequality $\phi'(\lambda) \ge \beta\phi'(0)$.) Inequality (2.8) is a sufficient decrease condition on $\|F(u_k + \lambda s_k)\|$; cf. (2.5). Since $\phi'(0) < 0$ and $0 < \alpha < 1$, (2.8) holds for sufficiently small $\lambda > 0$. Inequality (2.9) does *not* hold for small $\lambda > 0$, and its primary function is to prevent steps from being too short to give adequate progress toward a solution. Additionally, (2.9) is sometimes called a *curvature condition* since it implies

$$\phi'(\lambda) - \phi'(0) \ge (1 - \beta)|\phi'(0)| > 0,$$

from which it follows that the average curvature of ϕ on $(0, \lambda)$ is positive [34]. Such conditions are used in the optimization setting to ensure that certain quasi-Newton updates inherit positive-definiteness (see, e.g., [36]).

In our context, $\phi(\lambda) \ge 0$ for all λ, and an easy calculus argument shows that there is at least one $\lambda > 0$ that satisfies both (2.8) and (2.9) provided $\beta \ge \alpha$, which is usually the case in practice. (In our implementation, we used $\alpha = 10^{-4}$ and $\beta = .9999$.) If $\beta < \alpha$, then there may be no $\lambda > 0$ that satisfies both (2.8) and (2.9). However, if the set of λ satisfying (2.8) contains a local minimizer of ϕ, then this set contains solutions of (2.9) as well, and small values of β may serve to restrict solutions of (2.8)-(2.9) to be near this minimizer.

The line search generates a sequence of iterates that lie within "intervals of uncertainty" and are additionally constrained to be within an interval $[\lambda_{\min}, \lambda_{\max}]$ for specified $0 < \lambda_{\min} < \lambda_{\max}$. Since ϕ is bounded below, it is possible to determine $\lambda_{\min}$ and $\lambda_{\max}$ for each k to ensure that $[\lambda_{\min}, \lambda_{\max}]$ contains at least one λ satisfying (2.8) and (2.9), provided $\beta \ge \alpha$;

however, in typical practice, fixed $\lambda_{\min}$ and $\lambda_{\max}$ are used for all k. (We used $\lambda_{\min} = 10^{-12}$ and $\lambda_{\max} = 10^6$ in our implementation.)

The intervals of uncertainty are bounded by points λ_ℓ and λ_u that are updated at each iteration in order to maintain λ_ℓ as the "best point so far" in a certain sense and, with high likelihood, ultimately to bracket a minimizer of ϕ within the interval, if that is possible. The rules for updating λ_ℓ and λ_u are complex and are explained in detail in Appendix 1. As noted there, each update may involve multiplication of s_k by the Jacobian evaluated at a new point, which, while presumably feasible in the Newton–Krylov context, will entail additional, possibly significant expense. (In our implementation, these products can be produced either analytically with a freshly-evaluated Jacobian or approximated as directional derivatives using finite-differences of F-values.)

Given a current interval of uncertainty, the next iterate is determined according to sophisticated rules that select and, in some cases, combine minimizers of cubic and quadratic interpolating polynomials, together with safeguards that ensure that the iterate lies within $[\lambda_{\min}, \lambda_{\max}]$ and other desired bounds and still gives adequate movement. We refer the reader to [34] for details.

The possible outcomes of the line search, as explained in Appendix 1, are as follows:

1. The iterates increase monotonically and reach $\lambda_{\max}$ after a finite number of iterations; with $\lambda = \lambda_{\max}$, (2.8) holds but (2.9) may not hold.
2. The iterates decrease monotonically and reach $\lambda_{\min}$ after a finite number of iterations; neither (2.8) nor (2.9) is guaranteed to hold with $\lambda = \lambda_{\min}$.
3. A value of $\lambda \in (\lambda_{\min}, \lambda_{\max})$ is reached for which (2.8) holds and (2.9) also holds with $\beta = \alpha$.
4. A value of $\lambda \in (\lambda_{\min}, \lambda_{\max})$ is reached for which both (2.8) and (2.9) hold.

Note that if one of the first two outcomes occurs, then both (2.8) and (2.9) *may* hold, but this is not guaranteed. Note also that if (2.9) holds with $\beta = \alpha$, then it also holds with $\beta \geq \alpha$. Thus if $\beta \geq \alpha$, as in our implementation, then the third and fourth outcomes become one.

Our global convergence result for Algorithm INMTL is Theorem 2.5 below. The proof is given in Appendix 1.

Theorem 2.5 *Suppose that u_0 is given and that F is Lipschitz continuously differentiable on $\mathcal{L}(u_0) \equiv \{u : \|F(u)\| \leq \|F(u_0)\|\}$, i.e., $F'(u)$ exists everywhere on $\mathcal{L}(u_0)$ and there is a $\Gamma \geq 0$ such that*

$$\|F'(v) - F'(u)\| \leq \Gamma \|v - u\| \tag{2.10}$$

for all $u \in \mathcal{L}(u_0)$ and nearby $v \in \mathcal{L}(u_0)$. Assume that $\{u_k\}$ is produced by Algorithm INMTL *such that, for each k, the λ determined by the Moré–Thuente line search satisfies*

(2.8) *and* (2.9) *with* ϕ *defined in* (2.7). *If* $\{u_k\}$ *has a subsequence* $\{u_{k_j}\}$ *such that* $F'(u_{k_j})$ *is nonsingular for each* j *and* $\{\|F'(u_{k_j})^{-1}\|\}$ *is bounded, then* $F(u_k) \to 0$. *If* $\{u_k\}$ *has a limit point* u_* *such that* $F'(u_*)$ *is nonsingular, then* $F(u_*) = 0$ *and* $u_k \to u_*$.

Note that, in contrast to Theorem 2.4 above and Theorem 2.7 below, Theorem 2.5 provides no assurance that initial inexact Newton steps will ultimately be accepted without modification as the iterates near a solution. Indeed, such an assurance cannot be made without further assumptions. For example, it is well-known that one must require $\alpha < 1/2$ in order to ensure that exact Newton steps will ultimately be acceptable without modification near a solution (see, e.g., [12]). The following lemma, which is proved in the appendix, generalizes this observation to the inexact Newton context. Note that the lemma reduces to standard results (cf. [12, Th. 6.3.4]) if $\lim_{k\to\infty} \eta_k = 0$.

Lemma 2.6 *Suppose that* $\{u_k\}$ *produced by Algorithm* INMTL *converges to* u_* *for which Assumption* 2.1(a) *holds. Then, with* ϕ *defined in* (2.7), (2.8) *holds with* $\lambda = 1$ *for all sufficiently large* k *if*

$$\alpha < \frac{1 - \limsup_{k\to\infty} \eta_k}{2} \tag{2.11}$$

and only if

$$\alpha < \frac{1}{2\,(1 - \liminf_{k\to\infty} \eta_k)}\ . \tag{2.12}$$

Additionally, (2.9) *holds for all sufficiently large* k *if*

$$\beta > \frac{\limsup_{k\to\infty} \eta_k\,(1 + \limsup_{k\to\infty} \eta_k)}{1 - \limsup_{k\to\infty} \eta_k}\ . \tag{2.13}$$

If something is known about $\limsup_{k\to\infty} \eta_k$, then (2.11) and (2.13) may provide useful guidance in specifying α and β. Note, however, that the bound on the right-hand side of (2.13) is not less than one if $\limsup_{k\to\infty} \eta_k \geq \sqrt{2} - 1$, in which case (2.13) is not helpful.

Lemma 2.6 can be used to advantage with the forcing terms considered here. In the case of the adaptive Choice 1 forcing terms, it is easy to show that $\limsup_{k\to\infty} \eta_k = 0$ under the assumptions of the lemma. Thus, in this case, the lemma implies that, for any $\alpha \in (0, \frac{1}{2})$ and $\beta \in (0, 1)$, initial inexact Newton steps are ultimately acceptable without modification and an inequality (2.4) holds for sufficiently large k and a γ independent of k. In the case in which $\eta_k = 10^{-4}$ for each k, it follows from the lemma that, under the scarcely more restrictive conditions $0 < \alpha < (1 - 10^{-4})/2$ and $10^{-4}(1 + 10^{-4})/(1 - 10^{-4}) < \beta < 1$, initial inexact Newton steps are again ultimately acceptable and the convergence obeys (2.3) with $\eta_* = 10^{-4}$. The values $\alpha = 10^{-4}$ and $\beta = .9999$ used in our implementation generously satisfy (2.11) and (2.13) in either case and are typical values used in practice.

2.4 The dogleg method

The traditional dogleg method (cf. [39], [12]) determines, at the kth Newton iteration, a step along the *dogleg curve* Γ_k^{DL}. This is the piecewise linear curve connecting 0, the "Cauchy point" s_k^{CP} (defined to be the minimizer of the local linear model norm in the steepest descent direction), and the Newton step $s_k^{\mathrm{N}} = -F'(u_k)^{-1}F(u_k)$. The dogleg curve has the desirable properties that, as a point s traverses the curve from 0 to s_k^{CP} to s_k^{N}, $\|s\|$ is monotone increasing and $\|F(u_k) + F'(u_k)\, s\|$ is monotone decreasing (see, e.g., [12]). Consequently, if $\delta > 0$ is a given trust region radius, then there is a unique $s_k \in \Gamma_k^{\mathrm{DL}}$ such that $s_k = \arg\min_{s \in \Gamma_k^{\mathrm{DL}}, \|s\| \le \delta} \|F(u_k) + F'(u_k)\, s\|$, and this s_k is characterized as follows: If $\|s_k^{\mathrm{N}}\| \le \delta$, then $s_k = s_k^{\mathrm{N}}$, and if $\|s_k^{\mathrm{N}}\| > \delta$, then s_k is the unique point on the dogleg curve satisfying $\|s_k\| = \delta$. If s_k so chosen is acceptable, then the next approximate solution is $u_{k+1} = u_k + s_k$; if not, then the trust region radius is reduced and a new s_k is similarly determined.

In this study, we use a straightforward adaptation of the traditional method, outlined in general form below, that is suitable for implementation as a Newton–Krylov method. In this, each Newton step s_k^{N} is replaced by an inexact Newton step s_k^{IN}, and the corresponding dogleg curve Γ_k^{DL} connects 0, the Cauchy point s_k^{CP}, and s_k^{IN}. We note that the computation of s_k^{CP} requires the product of $F'(u_k)^T$ with a vector. As indicated in the introduction, these products can be evaluated by our test codes. However, they may not be readily available in other circumstances, especially those involving "matrix-free" Newton–Krylov implementations in which the Jacobian is not created. A Newton–GMRES dogleg adaptation that does not require these products is described in [3]; in this, each Cauchy point s_k^{CP} is replaced by an approximation determined using quantities generated by GMRES.

Algorithm INDL: Inexact Newton Dogleg Method

LET u_0, $\eta_{\max} \in [0, 1)$, $t \in (0, 1)$, $0 < \theta_{\min} < \theta_{\max} < 1$, AND

$0 < \delta_{\min} \le \delta$ BE GIVEN.

FOR $k = 0, 1, \ldots$ (UNTIL CONVERGENCE) DO:

CHOOSE $\eta_k \in [0, \eta_{\max}]$ AND s_k^{IN} SUCH THAT

$$\|F(u_k) + F'(u_k)\, s_k^{\mathrm{IN}}\| \le \eta_k \|F(u_k)\|.$$

EVALUATE s_k^{CP} AND DETERMINE $s_k \in \Gamma_k^{\mathrm{DL}}$.

WHILE $ared_k < t \cdot pred_k$ DO:

CHOOSE $\theta \in [\theta_{\min}, \theta_{\max}]$.

UPDATE $\delta \leftarrow \max\{\theta\delta, \delta_{\min}\}$.

REDETERMINE $s_k \in \Gamma_k^{\mathrm{DL}}$.

SET $u_{k+1} = u_k + s_k$ AND UPDATE δ.

The procedure for determining each $s_k \in \Gamma_k^{\mathrm{DL}}$ is as follows:

- If $\|s_k^{\mathrm{IN}}\| \le \delta$, then $s_k = s_k^{\mathrm{IN}}$.
- Otherwise, if $\|s_k^{\mathrm{CP}}\| \ge \delta$, then $s_k = (\delta/\|s_k^{\mathrm{CP}}\|)\, s_k^{\mathrm{CP}}$.
- Otherwise, $s_k = (1-\tau)s_k^{\mathrm{CP}} + \tau s_k^{\mathrm{IN}}$, where $\tau \in (0,1)$ is uniquely determined so that $\|s_k\| = \delta$.

This procedure always determines s_k uniquely and is standard for dogleg implementations. However, there are several issues that arise as a consequence of using s_k^{IN} in place of s_k^{N}. First, for any $\eta_k \in (0, \eta_{\max}]$, no matter how small, $\|F(u_k) + F'(u_k)\, s\|$ may not be monotone decreasing as s traverses Γ_k^{DL} from s_k^{CP} to s_k^{IN}; consequently, s_k may not minimize the local linear model norm along Γ_k^{DL} within the trust region. However, if η_k is small, then this non-monotonicity can occur only in a small neighborhood of s_k^{IN} and is not a serious concern. Second, unless η_k is sufficiently small, $\|s\|$ may not be monotone increasing as s traverses Γ_k^{DL} from s_k^{CP} to s_k^{IN}; indeed, if $\|s_k^{\mathrm{CP}}\| > \delta$ and η_k is not small, then Γ_k^{DL} may have up to *three* points of intersection with the trust region boundary: one between 0 and s_k^{CP} and one or two between s_k^{CP} and s_k^{IN}. Thus s_k may not be uniquely characterized by $\|s_k\| = \delta$ when $\|s_k^{\mathrm{IN}}\| > \delta$. Third, and perhaps of greatest concern, if η_k is sufficiently large to allow

$$\eta_k \|F(u_k)\| \ge \|F(u_k) + F'(u_k)\, s_k^{\mathrm{IN}}\| \ge \|F(u_k) + F'(u_k)\, s_k^{\mathrm{CP}}\|$$

and $\|s_k^{\mathrm{IN}}\| \le \delta \le \|s_k^{\mathrm{CP}}\|$, then the procedure will specify $s_k = s_k^{\mathrm{IN}}$, even though the step $(\delta/\|s_k^{\mathrm{CP}}\|)s_k^{\mathrm{CP}}$ may (depending on δ, s_k^{CP}, and s_k^{IN}) give greater reduction of the local linear model norm along Γ_k^{DL} within the trust region. Although Algorithm INDL was effective in our tests, as evidenced by the results in §5, these issues remain a potential cause for concern. We are currently exploring alternative strategies that mitigate them [37].

In the while-loop, $ared_k$ and $pred_k$ are defined as in (2.6). In our implementation, we used $t = 10^{-4}$ as in the case of Algorithm INB. In updating δ within the while-loop, we used a simple reduction $\delta \leftarrow .25\delta$. (Alternatives based on minimizing interpolating polynomials are suggested in [12].) In the final update of δ following the while-loop, we used a procedure similar to that in [12], in which the trust region is shrunk if $\|F(u_k + s_k)\|$ and $\|F(u_k) + F'(u_k)\, s_k\|$ do not agree well, expanded if they agree especially well, and left unchanged otherwise. The specific procedure, in which $0 < \rho_{\mathrm{s}} < \rho_{\mathrm{e}} < 1$, $0 < \beta_{\mathrm{s}} < 1 < \beta_{\mathrm{e}}$, and $\delta_{\max} > \delta_{\min}$, is as follows:

- If $ared_k/pred_k < \rho_{\mathrm{s}}$ and $\|s_k^{\mathrm{IN}}\| < \delta$, then $\delta \leftarrow \max\{\|s_k^{\mathrm{IN}}\|, \delta_{\min}\}$.
- Otherwise, if $ared_k/pred_k < \rho_{\mathrm{s}}$, then $\delta \leftarrow \max\{\beta_{\mathrm{s}}\delta, \delta_{\min}\}$.
- Otherwise, if $ared_k/pred_k > \rho_{\mathrm{e}}$ and $\|s_k\| = \delta$, then $\delta \leftarrow \min\{\beta_{\mathrm{e}}\delta, \delta_{\max}\}$.

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

We conclude this subsection with a convergence theorem for the general Algorithm INDL. The proof is given in Appendix 1. The theorem affirms a notable property of Algorithm INDL shared with other trust region methods, viz., that every limit point of $\{u_k\}$ produced by the algorithm must be a *stationary point* of $\|F\|$.[5] (It is possible for line-search methods to produce iterates that converge to non-stationary points at which the Jacobian is singular; see [39] and [5].) Additionally, as in the case of Theorem 2.4, a particular consequence of the theorem is that if $\{u_k\}$ converges to a solution u_* such that $F'(u_*)$ is nonsingular, then the convergence is ultimately governed by the η_k's. Thus, as before, if $\eta_k = 10^{-4}$ for each k, then (2.3) holds with $\eta_* = 10^{-4}$, and if each η_k is given by (2.1) followed by (2.2), then an inequality (2.4) holds for sufficiently large k and a γ independent of k.

Theorem 2.7 *Assume that F is continuously differentiable. If u_* is a limit point of $\{u_k\}$ produced by Algorithm* INDL*, then u_* is a stationary point of $\|F\|$. If additionally $F'(u_*)$ is nonsingular, then $F(u_*) = 0$ and $u_k \to u_*$; furthermore, $s_k = s_k^{\mathrm{IN}}$ for all sufficiently large k.*

3 The discretized equations

In this section the governing transport PDEs are briefly presented. These equations describe the conservation of linear momentum and mass along with the thermal energy equation for a variable density low-speed flow. The physical transport mechanisms include diffusion, convection, a volumetric continuity constraint, external surface forces set by pressure boundary conditions and buoyancy forces that are incorporated in our examples by using a Boussinesq approximation [8].

To describe the system that defines our nonlinear residual equations, we present the transport equations below in residual form. In these equations the unknown quantities are $\mathbf{u}$, P, and T; these are, respectively, the fluid velocity vector, the hydrodynamic pressure, and the temperature.

Momentum Transport:

$$\mathbf{R_m} = \rho\mathbf{u}\cdot\nabla\mathbf{u} - \nabla\cdot\mathbf{T} - \rho\mathbf{g} \tag{3.1}$$

Total Mass Conservation:

$$R_P = \rho\nabla\cdot\mathbf{u} \tag{3.2}$$

Energy Transport:

$$R_T = \rho C_p\mathbf{u}\cdot\nabla T + \nabla\cdot\mathbf{q} \tag{3.3}$$

In these equations, ρ, $\mathbf{g}$, and C_p are, respectively, the density, the gravity vector, and the specific heat at constant pressure. The necessary constitutive equations for the stress tensor, $\mathbf{T}$ and the heat flux vector, $\mathbf{q}$ are given by (3.4)–(3.5) below.

[5] A vector $u \in \mathbb{R}^n$ is a stationary point of $\|F\|$ if $\|F(u)\| \le \|F(u) + F'(u)\,s\|$ for every $s \in \mathbb{R}^n$.

Stress Tensor:

$$\mathbf{T} = -P\mathbf{I} + \mu\{\nabla\mathbf{u} + \nabla\mathbf{u}^T\} \tag{3.4}$$

Heat Flux:

$$\mathbf{q} = -\kappa\nabla T \tag{3.5}$$

Here μ and κ are, respectively, the dynamic viscosity and the thermal conductivity.

The above equations are derived by assuming a low Mach number flow where density can vary only with temperature and viscous dissipation effects can be neglected in the energy transport equation (3.3). More information on this system of equations can be found in [48].

Finally, to complete the system, boundary conditions are imposed on (3.1)–(3.3) by taking combinations of Dirichlet conditions on $\mathbf{u}$, P, and T and specified flux conditions on $\mathbf{T}$ and $\mathbf{q}$. In §4.1, we discuss the specific boundary conditions for each test problem in more detail.

To obtain an algebraic system of equations $F(u) = 0$, a stabilized finite element formulation of (3.1)–(3.3) is employed. The stabilized method allows equal order interpolation of velocity and pressure and also provides stabilization of the convection operators to limit oscillations due to high grid Reynolds and Peclet number effects. This formulation follows the work of Hughes et al. [28] and Tezduyar [50]. Specifically, the discrete equations are obtained from the following equations.

Momentum:

$$F_{\mathbf{u_i}} = \int_\Omega R_{m_i}\Phi\, d\Omega + \sum_e \int_{\Omega^e} \tau_m(\mathbf{u}\cdot\nabla\Phi)R_{m_i}\, d\Omega \tag{3.6}$$

Total Mass:

$$F_P = \int_\Omega R_P\Phi\, d\Omega + \sum_e \int_{\Omega^e} \rho\tau_m\nabla\Phi\cdot\mathbf{R_m}\,\mathbf{d\Omega} \tag{3.7}$$

Energy:

$$F_T = \int_\Omega R_T\Phi\, d\Omega + \sum_e \int_{\Omega^e} \tau_T(\mathbf{u}\cdot\nabla\Phi)R_T\, d\Omega \tag{3.8}$$

In these equations the stability parameters (the τ's) are functions of the fluid velocity $\mathbf{u}$ and the transport coefficients of the PDEs and are given in [28], [50], [48].

To form the Jacobian F' of the system, the equations (3.6)–(3.8) are linearized. The discrete form of these linearized terms is then determined by expanding the unknowns $\mathbf{u}$, P, and T and the weighting function Φ in terms of a linear finite element basis. The resulting Newton equation (1.1) is a fully-coupled nonsymmetric linear system.

4 The test problems and algorithm evaluation framework

4.1 The test problems

The three test problems described below are standard benchmark problems used in the verification of fluid flow codes and solution algorithms [49]. In the numerical tests, we employed both 2D and 3D forms of these.

The thermal convection problem [9]

This problem concerns the flow of a fluid driven by thermal convection in a differentially heated square box in the presence of gravity. It requires the solution of the momentum transport, energy transport, and total mass conservation equations on the unit square in $I\!R^2$ or the unit cube in $I\!R^3$. In this formulation, a Boussinesq approximation to the body force terms is employed. When the equations and boundary conditions are suitably nondimensionalized, two nondimensional parameters appear. These are 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.

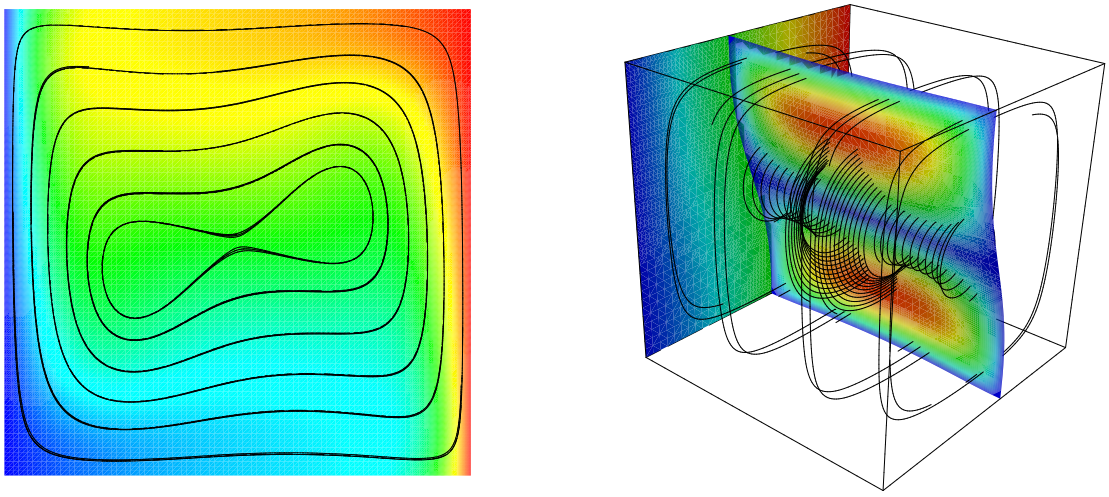
In this study, we took Pr = 1 and used Ra = 10^3, 10^4, 10^5, 10^6. The following Dirichlet and Neumann boundary conditions were imposed:

$$\begin{aligned} T = T_{cold},\ \mathbf{u} = 0 \quad & at\ x = 0, \\ T = T_{hot},\ \mathbf{u} = 0 \quad & at\ x = 1, \\ \frac{\partial T}{\partial y} = 0,\ \mathbf{u} = 0 \quad & at\ y = 0\ and\ y = 1, \end{aligned}$$

with the additional boundary condition in 3D

$$\frac{\partial T}{\partial z} = 0,\ \mathbf{u} = 0 \quad at\ z = 0\ and\ z = 1.$$

All solutions in 2D were computed on a 100×100 equally spaced mesh, which resulted in 40,804 unknowns for the discretized problem. In 3D, solutions were computed on a $32 \times 32 \times 32$ equally spaced grid, resulting in 179,685 unknowns for the discretized problem. Figure 1 depicts representative solutions of the problem.



Ra= 1.0E5
(a)

Ra= 1.0E3
(b)

Figure 1. Thermal convection problem: (a) 2D Model: Colored contour plot shows temperature magnitude with streamlines. (b) 3D Model: Colored side wall shows a temperature contour plot along with a temperature iso-surface plot colored by velocity vector magnitude with streamlines.

The backward-facing step problem [19]

This problem involves the simulation of flow through a rectangular channel that is initially constricted and subsequently expands over a reentrant backward-facing step. It requires the solution of the momentum transport equation and the total mass conservation equation. The nondimensionalized form of the equations and boundary conditions depends on the Reynolds number Re. As this parameter is increased, the nonlinear components of the equations become more dominant and the problem becomes more difficult. As the fluid flows downstream, it produces a recirculation zone on the lower channel wall, and for sufficiently high Reynolds numbers, it also produces a recirculation zone farther downstream on the upper wall. In our experiments, we used Re = 100, 200, . . . , 700, 750, 800.

In our 2D problem, the flow was computed only in the expanded portion of the channel, which had a 1×30 aspect ratio. Flow entering from the constricted portion was simulated by introducing a parabolic velocity profile in the upper half of the inlet boundary and imposing

zero velocity on the lower half. The boundary conditions were as follows:

$$\begin{aligned} \mathbf{u} = (24y(0.5-y),0)^T \quad & at\ x = 0,\ 0 \le y \le 0.5, \\ \mathbf{u} = 0 \quad & at\ x = 0,\ -0.5 \le y \le 0, \\ \mathbf{T}_{xx} = 0,\ \mathbf{T}_{xy} = 0 \quad & at\ x = 30,\ -0.5 \le y \le 0.5, \\ \mathbf{u} = 0 \quad & at\ y = -0.5\ and\ y = 0.5. \end{aligned}$$

The discretization was a 20×400 unequally spaced mesh (with a finer mesh near the step), which resulted in 25,263 unknowns.

In 3D, we computed the flow over the entire domain, with the step placed one-fourth of the distance along the channel. The height-length and height-width ratios for the constricted portion of the channel were 1×20 and 1×8, respectively. For the expanded portion, these ratios were 1×30 and 1×4, respectively.

The boundary conditions were as follows:

$$\begin{aligned} \mathbf{u} = (U_0,0,0)^T \quad & at\ x = 0,\ 0.1 \le y \le 0.2,\ -0.4 \le z \le 0, \\ \mathbf{T}_{xx} = 0,\ \mathbf{T}_{xy} = 0,\ \mathbf{T}_{xz} = 0 \quad & at\ x = 8,\ 0 \le y \le 0.2,\ -0.4 \le z \le 0, \\ \mathbf{u} = 0 \quad & on\ all\ other\ walls. \end{aligned}$$

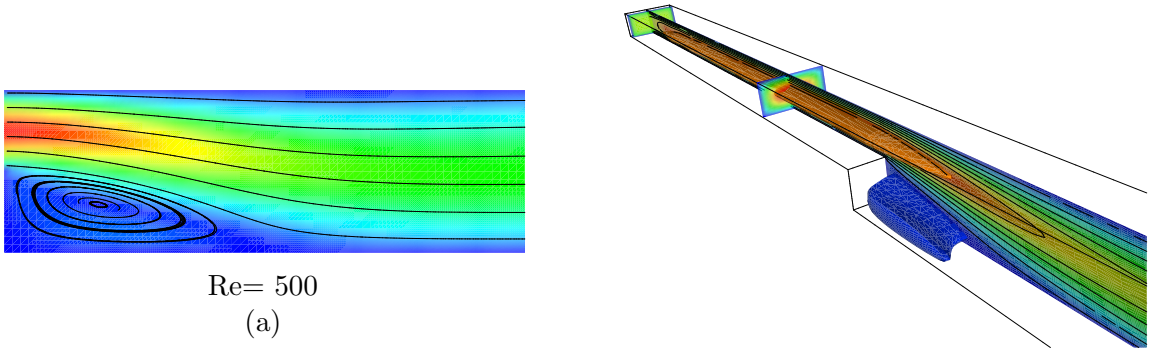
To provide an example of a problem with larger scale, we used a finer discretization on this problem than on the others, viz., a $20 \times 400 \times 30$ unequally spaced mesh. (As in the 2D case, the mesh was refined near the step.) This resulted in 1,042,236 unknowns. Figure 2 depicts representative solutions of the problem.

The lid driven cavity problem [20], [45]

The third test problem addresses a confined flow in the unit square in $\mathbb{R}^2$ or the unit cube in $\mathbb{R}^3$ driven by a moving upper boundary. Like the backward-facing step problem above, it requires solving the momentum transport equation and the total mass conservation equation, and a suitably nondimensionalized formulation leads to the appearance of the Reynolds number Re. As above, the problem becomes more difficult as Re increases, in this case with increasingly prominent regions of countercirculation appearing in the corners of the domain.

For the tests performed in 2D, we took $\mathrm{Re} = 1,000,\ 2,000,\ \ldots,\ 10,000$. (In one study, we also considered $\mathrm{Re} = 100,\ 200,\ \ldots,\ 1,000$ in order to obtain a larger data set.) In the 3D case, there is evidence that the stability of the solution is questionable for $\mathrm{Re} > 700$. Accordingly, for the 3D tests, we used the smaller values $\mathrm{Re} = 100,\ 200,\ \ldots,\ 1,000$. The boundary conditions were as follows:

$$\begin{aligned} \mathbf{u} = 0 \quad & at\ x = 0\ and\ x = 1, \\ \mathbf{u} = 0 \quad & at\ y = 0, \\ \mathbf{u} = \begin{cases} (U_0,0)^T & \text{in 2D} \\ (U_0,0,0)^T & \text{in 3D} \end{cases} \quad & at\ y = 1, \end{aligned}$$



Re= 500
(a)

Re= 200
(b)

Figure 2. Backward facing step problem: (a) 2D Model: Colored contour plot shows the velocity vector magnitude along with streamlines. (b) 3D Model: Colored iso-surfaces show a positive (orange) and negative (blue) x-velocity; a colored contour plot with iso-lines is shown on a central slice plane.

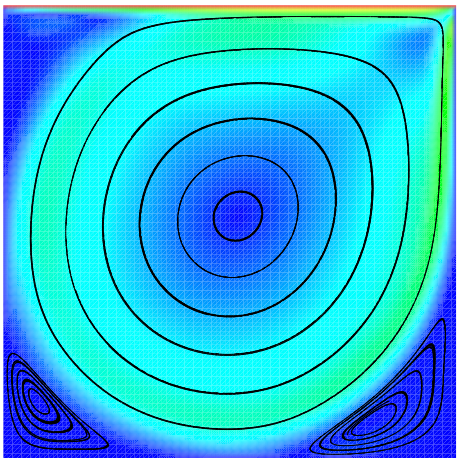
with the additional boundary condition in 3D

$$\mathbf{u} = 0 \quad at\ z = 0\ and\ z = 1.$$

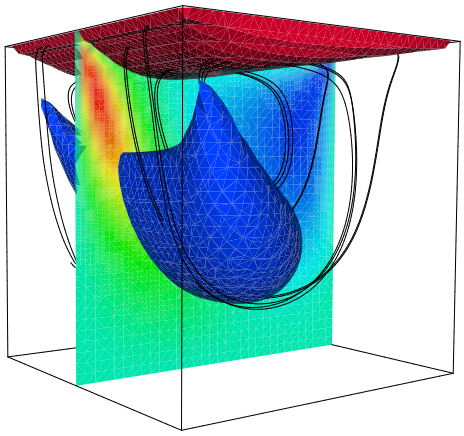
The 2D problem was discretized on a 100×100 equally spaced grid, which resulted in 30,603 unknowns for the discretized problem. In 3D, the discretization was a $32 \times 32 \times 32$ equally spaced grid, resulting in 143,748 unknowns. Figure 3 depicts representative solutions of the problem.

4.2 The algorithm evaluation framework

For our numerical tests, we used implementations of the nonlinear solution algorithms in §2 provided by the NOX nonlinear solver package [32]. NOX is a C++ object-oriented library designed for the efficient solution of systems of nonlinear equations. It offers various globalized Newton-based solvers, including those in §2, and other techniques such as tensor and Broyden methods. The GMRES implementation and preconditioners used to solve the linear subproblems were provided by the AztecOO package [24], an extension of the Aztec library [51], which provides an easy-to-use framework for the efficient parallel solution of linear systems through an extensive suite of Krylov solvers and preconditioners. We give more details below about the particular GMRES algorithm and preconditioners used in our tests. NOX and AztecOO were developed as part of the Trilinos project ([25], [15]), an effort to develop parallel solver algorithms and libraries within an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific applications.



Re= 5000
(a)



Re= 200
(b)

Figure 3. Lid driven cavity problem: (a) 2D Model: Colored contour plot of the velocity vector magnitude along with streamlines. (b) 3D Model: Colored iso-surfaces of positive (red) and negative (blue) x-velocities are shown along with a colored contour plot of y-velocity on a central slice plane. A few streamlines in the volume are also included.

The parallel finite element reacting flow code MPSalsa [48] was used to set up the finite element discretization described in §3 and to invoke the solvers. In brief, the parallel implementation of the finite element equations proceeds as follows: Chaco [23], a general graph partitioning tool, is used to partition the FE mesh into subdomains and assign subdomains to processors; then, using the decomposition created by Chaco, MPSalsa sets up the finite element discretization, with each processor producing a subset of the discretized equations and Jacobian entries corresponding to its assigned subdomain. Chaco uses a variety of new and established graph partitioning heuristics to construct partitions and subdomain mappings that result in low communication volume, good load balance, few message start-ups, and only small amounts of network congestion. For the results in this paper, multi-level methods with Kernighan–Lin improvement were used. For a detailed description of parallel FE data structures and a discussion of the strong link between partitioning quality and parallel efficiency, see [47].

In our tests, we used the restarted GMRES implementation in AztecOO with a restart value (maximum Krylov subspace size) of 200 and a maximum number of GMRES iterations equal to 600 (i.e., three restarts). Products of the Jacobian F' or, when needed, its transpose with vectors were evaluated by analytically evaluating F' as indicated in §3 and subsequently using the stored value of F' or its transpose to perform the necessary matrix-vector multiplications. (An exception was in the case of the Moré–Thuente line-search iterations: there, products of F' with vectors were, in most instances, approximated with finite-differences of F-values. See the remark in §5 for more details.)

In large-scale PDE applications such as those of interest here, preconditioning is usually a very important factor in GMRES performance and is the key to scalability on massively parallel machines. The preconditioners available in AztecOO include numerous algebraic and polynomial preconditioners and also additive Schwarz domain decomposition preconditioners, which use incomplete LU factorizations for subdomain solves and allow variable levels of overlap. (AztecOO also allows using a coarse grid with additive Schwarz preconditioning; however, we did not enable this in our experiments.) In our experiments, we used an additive Schwarz preconditioner with an ILUT(*fill-in*,*drop*) incomplete factorization [40], which allows the user to specify a fill-in parameter and a drop tolerance. For these, we used *fill-in* $= 1$ and *drop* $= 0$, resulting in no additional fill-in and no entries dropped due to small magnitude. In the parallel domain decomposition implementation, each processor is assigned one subdomain and performs ILUT factorizations and solves on that subdomain. To increase robustness, the preconditioners can effectively expand the individual subdomains to include FE nodes assigned to neighboring processors by allowing more overlap between subdomains. In a geometric sense, this overlap corresponds to increasing the size of the locally defined subdomain to include additional levels of FE nodes outside of the processor's assigned nodes. Thus a single level of overlapping uses only information from FE nodes that are connected by an edge (in the FE connectivity graph) that was cut by the original subdomain partition. Successive levels of overlap use this method recursively by considering the previously overlapped points now to be assigned nodes in the ILU preconditioner setup phase of the algorithm. The results in §5 were obtained using one level of overlap.

In all of our tests, we imposed left-sided diagonal scaling determined by the dependent variables. In this, at the kth step and for $i = 1, \ldots, n$, the ith dependent variable $F_i(u_k)$ was scaled by the inverse of the sum of the absolute values of the entries in the ith row of $F'(u_k)$. This scaling was very important to success in many cases. It was used not only in the GMRES solve but also throughout the algorithm by incorporating it as a diagonal scaling matrix in the inner product.

Successful termination of the Newton–Krylov iterations was declared if $\|F(u_k)\| \leq \varepsilon_F \|F(u_0)\|$, where $\varepsilon_F = 10^{-2}$ in our tests, *and* the following step-length criterion is also satisfied:

$$\frac{1}{n}||Ws_k||_2 < 1,$$

where n is the total number of unknowns and W is a diagonal weighting matrix with entries

$$W_{ii} = \frac{1}{\varepsilon_r |u_k^{(i)}| + \varepsilon_a},$$

in which $u_k^{(i)}$ is the ith component of u_k and $\varepsilon_r = 10^{-3}$ and $\varepsilon_a = 10^{-8}$ in our tests. In our experience, this second criterion is typically more stringent and is necessary to ensure that finer physical details of the flow and transport are adequately resolved. Essentially, it requires that each variable of the Newton correction be small relative to its current value, which assures that all variables are treated equitably in determining when to terminate. This weight-matrix definition is similar to a criterion used to dynamically control time step sizes and is standard in general purpose ODE packages such as LSODE [27].

All numerical experiments were performed on parallel machines located at Sandia National Laboratories in Albuquerque, New Mexico. All tests except those on the 3D backward-facing step problem were done on an IBM cluster having 16 nodes, with each node containing two 1 GHz Pentium III processors with 1 GB of RAM each. On this machine, the 2D tests were done using four nodes (eight processors), and the 3D tests on the thermal convection and lid driven cavity problems were done using 15 nodes (30 processors). The tests on the 3D backward-facing step problem were performed on the Sandia Institutional Cluster. This machine has 256 nodes, with each node containing two 3 GHz Pentium IV processors with 1 GB of RAM each; 50 nodes (100 processors) were used in our tests on the 3D backward-facing step problem.

5 Results

We performed extensive numerical tests involving the application of the methods in §2 to the benchmark problems in §4. For comparison, we also included in these tests a non-globalized method, i.e., a method taking full, unmodified inexact Newton steps. For each method, we used both small constant (10^{-4}) forcing terms and adaptive (Choice 1) forcing terms, as discussed in §2.1. In every test case, the initial approximate solution was the zero vector.

In the following, we begin with a report on two studies based on these tests. The first is a robustness study, in which we observed the numbers of failures of the methods on the test

problems as problem parameters varied over specified ranges. The second is an efficiency study, in which we tabulated run times and other statistics for the methods when applied to a selected subset of the test problems. We conclude this section with graphs that provide further details and additional perspectives on these studies.

In the robustness study, to give insight into the distribution of failures of the methods in §2, we divided the problem regimes into "easier" and "harder" parameter ranges, as follows:

		Easier	Harder
2D and 3D Thermal Convection	Ra =	10^3, 10^4, 10^5	10^6
2D and 3D Backward Facing Step	Re =	100, ..., 500	600, 700, 750, 800
2D Lid Driven Cavity	Re =	$1,000$, ..., $5,000$	$6,000$, ..., $10,000$
3D Lid Driven Cavity	Re =	100, ..., 500	600, ..., $1,000$

The numbers of instances in which the methods failed in these ranges are shown in Tables 1 and 2. For convenience, total numbers of failures are summarized in Table 3. In these tables and in Table 4 that follows, "Backtracking, Quadratic/Cubic" refers to Algorithm INB with $\theta \in [\theta_{\min}, \theta_{\max}]$ determined at each inexact Newton step by minimizing a quadratic interpolating polynomial on the first step-length reduction and then minimizing cubic interpolating polynomials on all subsequent step-length reductions. "Backtracking, Quadratic Only" refers to the method with minimization of only quadratic interpolating polynomials. "Full Step" designates the non-globalized method that takes full, unmodified inexact Newton steps. For each method, the two table rows to the right show the numbers of failures with adaptive forcing terms (upper row) and with constant forcing terms (lower row).

The results summarized in Table 3 indicate that, overall, each of the globalizations of interest significantly improved robustness in these experiments. The backtracking methods and the dogleg method suffered fewer failures overall than the Moré–Thuente line-search method by a modest margin. The more detailed results in Tables 1 and 2 show that the degree of improvement resulting from globalization varied considerably among the test cases. For example, the improvement was striking in all cases involving the 2D backward-facing step problem but only marginal in the tests with small constant forcing terms on the 2D lid driven cavity problem. No globalization was always superior to the others in these tests. For example, the backtracking/line-search methods succeeded in every case of the 2D thermal convection problem, while the dogleg method failed in one instance; however, the dogleg method was the only method to succeed in every case of the 2D backward-facing step problem. Overall, the adaptive forcing terms significantly improved the robustness of all methods, including the "Full Step" method; for the globalized methods, the improvement was dramatic. This outcome is consistent with the study in [49], which included results for other adaptive forcing terms from [14] and a larger constant forcing term (10^{-1}) in addition to the forcing terms considered here. Overall, the combination of adaptive forcing terms and globalization is very effective on these problems; however, even this combination did not lead to success in every case.

Method	2D Thermal Convection		2D Lid Driven Cavity		2D Backward Facing Step	
	Easier	Harder	Easier	Harder	Easier	Harder
Backtracking, Quadratic/Cubic	0	0	0	0	0	1
	0	0	4	5	0	0
Backtracking, Quadratic Only	0	0	0	0	0	0
	0	0	4	5	0	1
Moré–Thuente Line Search	0	0	0	0	0	1
	0	0	4	5	0	0
Dogleg	0	0	0	0	0	0
	0	1	4	5	0	0
Full Step	0	1	4	5	1	4
	0	1	5	5	3	4

Table 1. Distribution of failures for the 2D problems: For each method, the upper and lower rows to the right show the numbers of failures with the adaptive and constant forcing terms, respectively.

A further remark is in order concerning the results for the Moré–Thuente line search. As noted in §2.3, each iteration of the line search requires multiplication of a vector by the Jacobian evaluated at a new point. For all of the test problems except one, these products were satisfactorily approximated in our experiments using finite-differences of F-values. The exception was the 3D backward-facing step problem, in which the finite-difference approximations were at times sufficiently in error to cause the algorithm to fail. (Specifically, the error in these cases was such that the algorithm declared the trial steps not to be descent directions and terminated with failure.) For this problem, it was necessary to evaluate each of these products analytically, at the cost of a fresh Jacobian evaluation and a matrix-vector multiplication, in order to obtain the level of success shown in Table 2.

In the efficiency study, we compiled various statistics for a selected set of test problem cases. This set included all cases considered in the robustness study in which all of the globalized methods succeeded. Additionally, since at least one globalized method failed on the 2D lid driven cavity problem for each value of Re above $1,000$, we included cases of this problem with $100 \leq \mathrm{Re} \leq 1,000$. (We did not include the non-globalized "Full Step" method in this study because its high incidence of failure would have made the test set undesirably small.) The specific cases considered are as follows:

Method	3D Thermal Convection		3D Lid Driven Cavity		3D Backward Facing Step	
	Easier	Harder	Easier	Harder	Easier	Harder
Backtracking, Quadratic/Cubic	0	0	0	0	0	0
	0	0	0	0	0	0
Backtracking, Quadratic Only	0	0	0	0	0	0
	0	0	0	0	0	0
Moré–Thuente Line Search	0	0	0	0	0	1
	0	0	0	0	0	2
Dogleg	0	0	0	0	0	0
	0	0	0	0	0	0
Full Step	0	1	0	0	0	3
	0	1	0	4	1	4

Table 2. Distribution of failures for the 3D problems: For each method, the upper and lower rows to the right show the numbers of failures with the adaptive and constant forcing terms, respectively.

2D Thermal Convection	Ra =	10^3, 10^4, 10^5
3D Thermal Convection	Ra =	10^3, 10^4, 10^5, 10^6
2D and 3D Backward Facing Step	Re =	100, 200, ..., 700
2D and 3D Lid Driven Cavity	Re =	100, 200, ..., 1,000

The results for these test cases are given in Table 4, which shows for each method mean numbers of inexact Newton steps, backtracks per inexact Newton step, total function evaluations, total GMRES iterations, and GMRES iterations per inexact Newton step, and also mean run times relative to that of the "Backtracking, Quadratic Only" method with constant forcing terms. All values are geometric means except for numbers of backtracks per inexact Newton step, which are arithmetic means.

The results in Table 4 indicate that, for a particular choice of the forcing terms, the globalized methods performed rather similarly on this test set. There are some differences in performance, but, in view of the modest size of the test set, these seem unlikely to have much significance. In contrast, notable differences are seen with each method when different forcing terms were used. Compared to the small constant forcing terms, the adaptive forcing

Method	2D Problems		3D Problems		All Problems	
Backtracking, Quadratic/Cubic	1	10	0	0	1	10
	9		0		9	
Backtracking, Quadratic Only	0	10	0	0	0	10
	10		0		10	
Moré–Thuente Line Search	1	10	1	3	2	13
	9		2		11	
Dogleg	0	10	0	0	0	10
	10		0		10	
Full Step	15	33	4	14	19	47
	18		10		28	

Table 3. Failure totals: For each method, the upper and lower rows in the second, fourth, and sixth columns show the numbers of failures with the adaptive and constant forcing terms, respectively; the third, fifth, and seventh columns show total numbers of failures.

terms resulted in greatly reduced mean numbers of GMRES iterations per inexact Newton step and significantly reduced numbers of total GMRES iterations. On the other hand, the small constant forcing terms required significantly fewer inexact Newton steps on average. In the balance, the adaptive forcing terms yielded considerably better run times than those obtained with the small constant forcing terms (except in the case of the Moré–Thuente linesearch, in which the improvement was slight). A likely explanation is that, on most inexact Newton steps, the small constant forcing terms were smaller than the adaptive forcing terms. Consequently, they often achieved more nonlinear residual norm reduction, but only at the cost of significant oversolving that ultimately outweighed this advantage.

The summary results presented in the two studies above, while providing very useful views of overall performance, necessarily contain less than full information about the test outcomes. Specifically, Tables 1–3 do not show particular instances of failure, and Table 4 does not include information from test cases in which one or more of the globalized methods failed. For completeness and to provide further details of method performance on the entire test set, we include Figures 4–9 below. The bar graphs in these figures show run times (in seconds) on all problems for all methods (including the non-globalized "Full Step" method) except the "Backtracking, Quadratic/Cubic" method, which has been omitted because its performance was so similar to that of the "Backtracking, Quadratic Only" method. The failure of a method in a particular case is indicated by a negative bar. Methods using

Method	Inexact Newton Steps	Backtracks per INS	Total Function Evaluations	Total GMRES Iterations	GMRES Iterations per INS	Normalized Time
Backtracking, Quadratic/Cubic	16.0	0.13	19.2	989.6	61.9	0.77
	9.20	0.20	11.8	1498	163	1.02
Backtracking, Quadratic Only	16.0	0.13	19.2	997.1	62.2	0.77
	9.23	0.18	11.7	1502	163	1.0 (REF)
Moré–Thuente Line Search	15.2	0.17	43.4	979.7	64.3	0.90
	8.67	0.17	25.1	1408	162	0.96
Dogleg	17.0	NA	18.9	1454	85.3	0.83
	10.7	NA	12.6	1799	168	1.01

Table 4. Efficiency Study: For each method, the upper and lower rows to the right show results with the adaptive and constant forcing terms, respectively. Times are relative to the "Backtracking, Quadratic Only" method with constant forcing terms. "INS" stands for "Inexact Newton Steps."

the small constant forcing terms are shown in shades of blue; methods using the adaptive forcing terms are shown in shades of red.

The results for the thermal convection problem in Figures 4 and 5 show failures at only the largest Rayleigh number Ra of 10^6. At this Ra value, the non-globalized "Full Step" method always failed; in the 2D case, the dogleg method with constant forcing terms also failed. For the backward-facing step problem, Figures 6 and 7 show many more failures. At low values of the Reynolds number Re, all methods converged. As the problems became more difficult with increasing Re, the first method to fail in both the 2D and 3D cases was the "Full Step" method with a small constant forcing term. Increasing Re further resulted in failure of the "Full Step" method with adaptive forcing terms as well. At the two largest values of Re, failures of the globalized methods were observed. The only methods that converged over the entire range of Re values were the "Backtracking, Quadratic Only" method with adaptive forcing terms and the dogleg algorithm with both adaptive and small constant forcing terms. The 2D lid driven cavity problem (Figure 8) was more difficult to solve. No method succeeded with small constant forcing terms beyond $\mathrm{Re} = 1,000$; however, all globalized methods in combination with adaptive forcing terms attained convergence over the entire range. In the 3D cases (Figure 9), the methods exhibited excellent convergence, with only the "Full Step" method with small constant forcing terms failing.

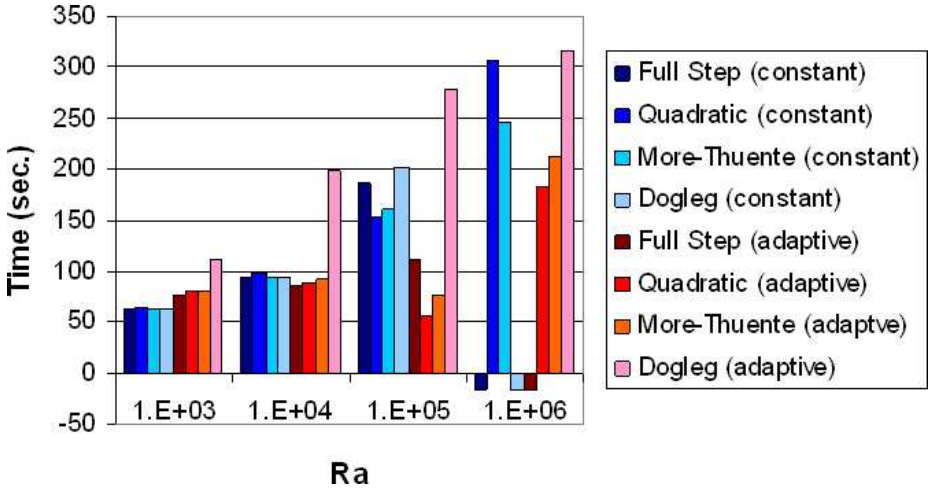


Figure 4. 2D Thermal Convection timings

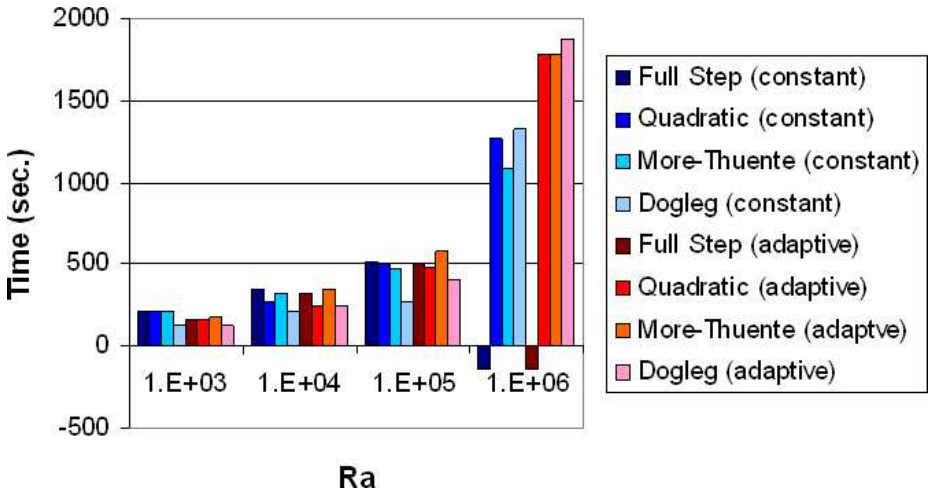


Figure 5. 3D Thermal Convection timings

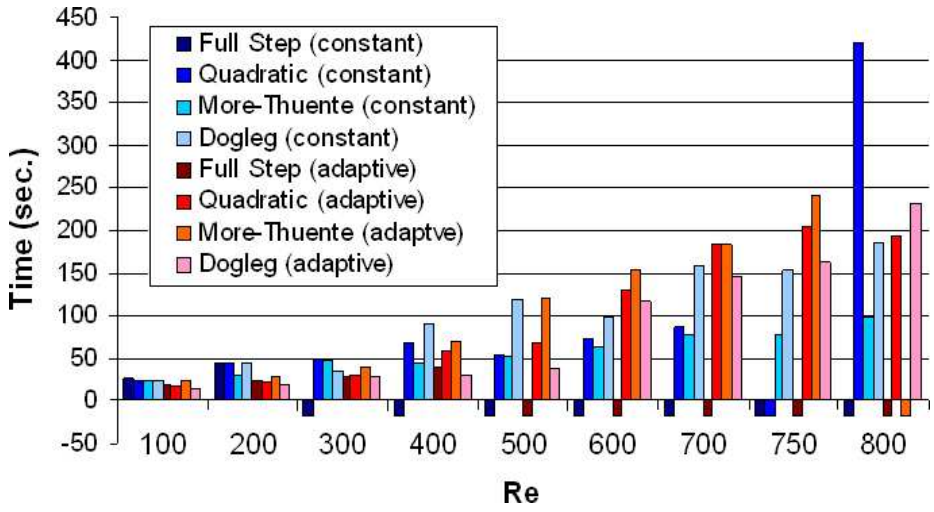


Figure 6. 2D Backward Facing Step timings

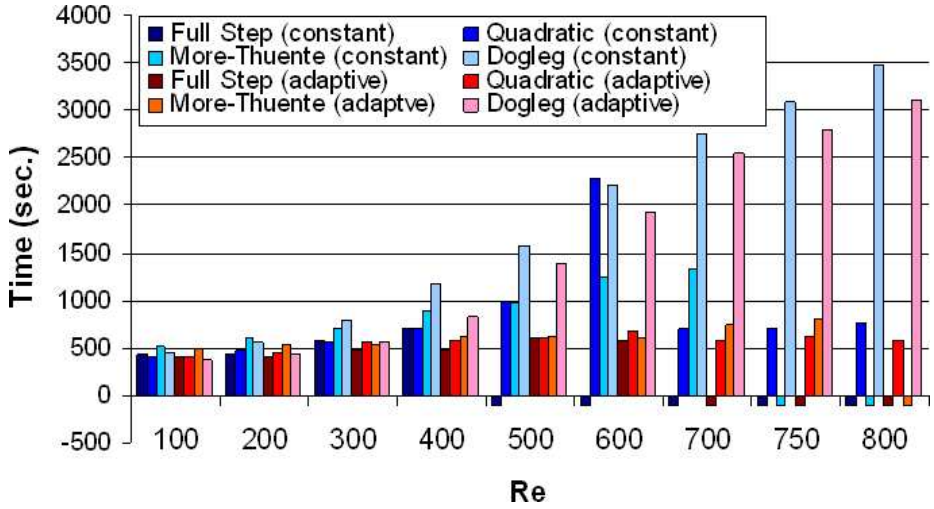


Figure 7. 3D Backward Facing Step timings

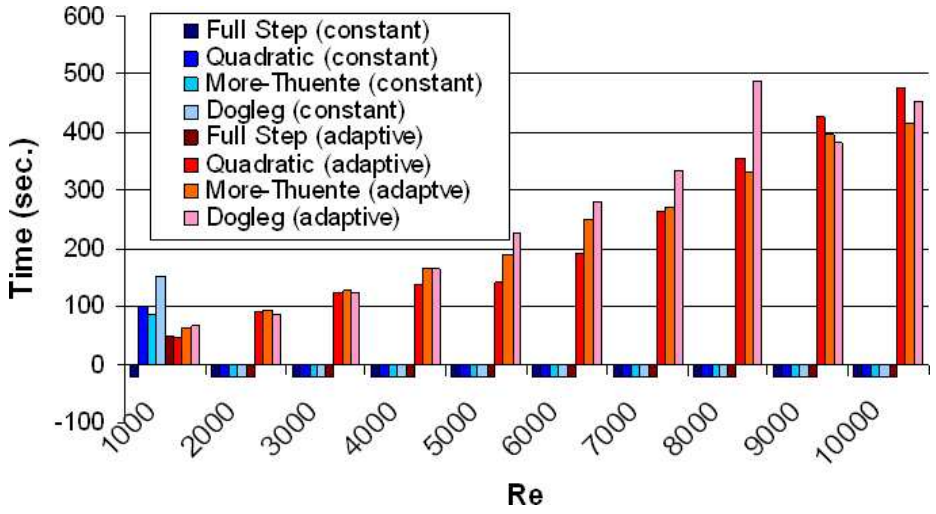


Figure 8. 2D Lid Driven Cavity timings

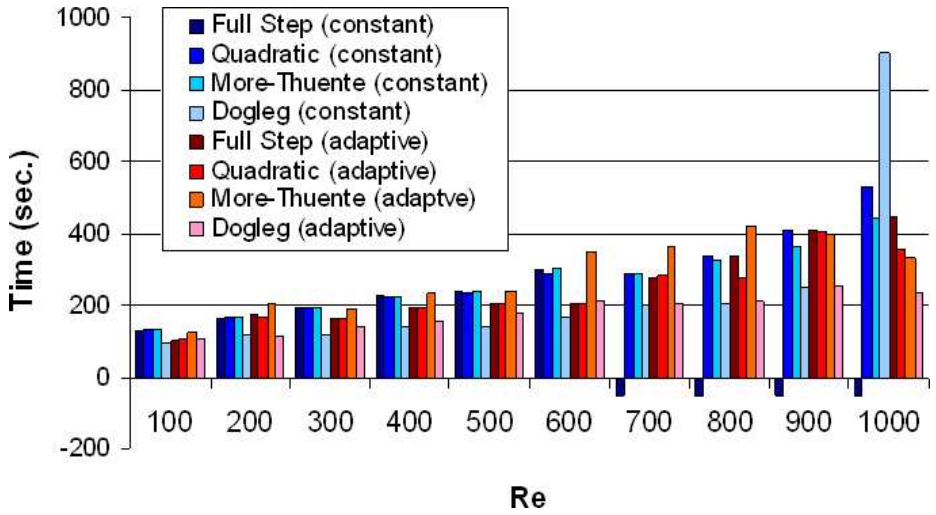


Figure 9. 3D Lid Driven Cavity timings

6 Additional remarks on failure and robustness

In practice, globalized Newton–Krylov methods can fail in a number of ways, and there are a variety of factors that contribute to their success or failure. We describe below several general failure modes and comment on the extent to which these were observed in our tests.

- *Fatal near-stagnation.* In this, the method achieves sufficient residual norm reduction at each step to proceed but not enough in the aggregate to achieve success before the maximum allowable number of steps has been reached. This mode accounted for most of the failures in our tests: 26 out of 33 failures for the backtracking and line-search methods and all 10 failures for the dogleg method. In our tests, we specified generous maximum allowable numbers of steps, specifically, 300 steps for the 2D lid driven cavity problem and 200 steps in all other cases; in no failure case did it seem likely that increasing these numbers would have ultimately resulted in success. Likely possible causes underlying this failure mode include problem conditioning and nonlinearity. (See [6] for interesting perspectives.)

- *Globalization failure.* In failures of this type, the globalization fails to determine an acceptable step at some Newton–Krylov iteration. For example, a backtracking routine might fail to produce an acceptable step within the maximum allowable number of step-length reductions. In our tests, such failures accounted for seven of 33 failures for the backtracking and line-search methods. Like fatal near-stagnation, failures of this type can result from problem conditioning and nonlinearity. (See [52] for a detailed example involving the 2D lid driven cavity problem.) A particular circumstance in which such failures may occur is convergence of the iterates to a local residual norm minimizer that is not a solution, at which the residual function is non-zero and the Jacobian is singular. In this event, because of the singularity of the Jacobian at the minimizer, initial steps produced by the Krylov solver may become increasingly long as the minimizer is approached. If this occurs, then steps may have to be reduced increasingly in length in order to be acceptable; as a result, the number of necessary step-length reductions may eventually exceed the maximum number allowed. We did not verify that this particular cause of failure occurred in our tests but did observe an instance of backtracking failure in which, at the iteration when failure occurred, the initial step returned by GMRES, while unacceptable, would have led ultimately to success if it had been taken. (Our implementation optionally allows taking this "recovery step" in the event of backtracking failure.) This outcome is consistent with the GMRES step "escaping" a basin of attraction of a local residual norm minimizer.

- *Divergence.* This failure mode is characterized by clear failure of the iterates to converge. This is likely to be manifested in unbounded growth of the iterates, which we observed in some of our tests involving the Newton–GMRES method with no globalization. We did not observe unbounded iterate growth in any of our tests of the globalized methods. However, this can occur for globalized methods; see [13, p. 400] for a simple example. It is also possible for the sequence of iterates to have several limit points, with the Jacobian singular at each; however, while this has been shown to occur in contrived examples [13, p. 400], it seems unlikely to occur in real applications.

- *Component failure.* By this, we mean the failure of one or more "components" of the

algorithm, such as the Krylov solver, the preconditioner, or the function evaluation routine. We saw no failures of this type in our tests. Such failures can occur for a variety of reasons and are best addressed through thoughtful algorithm and code design.

There are many ancillary factors that may affect the robustness of a globalized Newton–Krylov method on problems such as those of interest here. We conclude this section by discussing several that we have found to be influential.

• *The nonlinear solution structure of the continuous PDE problem.* Large-scale coupled nonlinear PDE systems often have characteristic parameters, such as the Reynolds number, the Rayleigh number, and the Prandtl number in the problems considered here. These parameters may strongly affect problem difficulty (see §4.1), and the limits of practical solvability may occur near parameter values at which the steady-state solution becomes unstable. More generally, the structure of the nonlinear solution branches obtained by varying the parameters can be very complex: multiple solutions can co-exist; isolated solution branches can appear; solution branches can intersect; stable steady-state solution branches can become unstable and vice versa; and steady-state solutions can become unstable to time-dependent disturbances. If the goal is to map out this complex nonlinear solution space, then the most appropriate methods may be continuation methods, which can follow stable and unstable solution branches and track critical points as parameters are varied (see [33] and [7] and the references therein). If the goal is to find a stable steady-state solution within a complex nonlinear landscape, then pseudo-transient continuation methods can also be used [30]. We note that, for large-scale problems, Newton–Krylov methods are often used as nonlinear solvers within these methods (cf. [43], [44], [55]).

• *The discretization of the PDE problem.* For complex PDE systems, the convergence of globalized Newton–Krylov methods is also affected by the spatial discretization. Failure of the spatial discretization to adequately reflect the underlying physics of the continuous problem can cause convergence difficulties. For example, in the case of strong convection (large Reynolds numbers) in our prototype problems, common spatial discretizations, such as centered finite-difference or Galerkin finite-element methods, become unstable when the computational mesh is too coarse, exhibiting non-physical spatial oscillations and producing Jacobian matrices that are ill-conditioned [16]. This ill-conditioning is likely to result in poor convergence of the Krylov solver, which can in turn lead to fatal near-stagnation of the Newton–Krylov method. (In our tests, we used a stabilized finite-element method [28] closely related to the streamline upwind Petrov-Galerkin (SUPG) method. In general, these schemes attempt to suppress spurious oscillations on coarser grids and also produce Jacobians that are better conditioned.) Additionally, ill-chosen spatial discretizations can produce spurious non-physical solutions. This has been demonstrated for the straightforward centered-difference discretization of the 2D lid driven cavity problem ([46], [55]). In this case, Newton–Krylov iterates may converge to these non-physical solutions from an unfortunate initial guess.

• *The convergence of the Krylov solver and preconditioning.* In general, the robustness and efficiency of a Newton–Krylov method are critically tied to the performance of the Krylov subspace method on the linear subproblems. In turn, effective preconditioning techniques (see, for example, [41] and [2]) are essential for good Krylov solver performance on the large,

complex nonlinear PDE systems of interest here. For these systems, we have already pointed out a number of issues that make the linear subproblems challenging. For high-resolution discretizations, an additional concern is the ill-conditioning that results from using very fine mesh spacing and/or very large aspect-ratio cells or elements in the discretization of the PDE problem, and well-chosen preconditioning is necessary to address this. Also, as indicated in §4.2, appropriate preconditioning techniques are central to the scalability of the Krylov solver on massively parallel platforms and, therefore, a crucial factor in the scalability of the overall Newton–Krylov method.

• *Scaling.* Proper scaling of variables can be an important contributor to method success. In our case, we found it effective to implement scaling consistently throughout the algorithm by incorporating it into a specific weighted norm defined by row-sums of the Jacobian matrix. This technique, described in §4.2, significantly improved the robustness of all methods in our tests.

• *Accuracy.* Finally, the accuracy of computations within the algorithm is important. As noted in §5, finite-difference approximations of Jacobian-vector products within the Moré–Thuente line search were not sufficiently accurate for good success on the 3D backward-facing step problem; analytic evaluations were necessary in this case. Also, in preliminary experimentation, we found that high-accuracy Jacobian evaluations, which were used to produce matrix-vector products in GMRES solves, resulted in much better method performance than certain cheaper but less accurate alternatives that were available in our codes. These high-accuracy evaluations were used in obtaining the results reported in §5.

7 Conclusions

We have considered several representative globalizations that can be used to improve the robustness of a Newton–Krylov method: two variants of a backtracking method from [13], a line-search procedure from [34], and a dogleg implementation of a trust region method ([39], [12]). (The backtracking variants differ in their step-length reduction strategies, with one minimizing both quadratic and cubic interpolating polynomials and the other minimizing only quadratics.) These methods all have strong global convergence properties, as indicated by the theoretical results outlined in §2. For each method, the main consequence of these results is that, under mild assumptions, if a sequence of iterates produced by the method has a limit point at which the Jacobian is nonsingular, then that point must be a solution and the iterates must converge to it. (The dogleg method has the additional property that, convergence aside, every limit point of the sequence of iterates must be a stationary point of the residual norm.) Moreover, the steps between successive iterates are ultimately the unmodified approximate solutions of (1.1) produced by the Krylov solver; hence, the speed of convergence of the iterates is ultimately governed by the forcing terms that determine the accuracy with which (1.1) is solved.

We extensively tested Newton–GMRES implementations of these methods on large-scale benchmark problems involving the steady-state 2D and 3D Navier–Stokes equations; the results are given in §5. Each of the globalizations considered here significantly improved

robustness in our tests. Overall, the two backtracking methods and the dogleg method were the most robust, with the two backtracking methods producing slightly better run times. (The two backtracking step-length reduction strategies produced very similar results.) The line-search procedure of [34] performed almost as well. These overall results notwithstanding, no method was better than the others in every test, and the only methods to succeed in every case were the backtracking method (with quadratic minimization only) and the dogleg method, with each using adaptive forcing terms.

The use of adaptive forcing terms resulted in major improvements in the robustness of all methods, including the method with no globalization. For the globalized methods, the improvement was dramatic. Using adaptive forcing terms also contributed significantly to the efficiency of the globalized methods.

Among the globalizations considered here, the backtracking method may be a first choice for implementation because of its simplicity as well as its effectiveness in our tests. In our backtracking tests, we saw no reason to prefer the step-length reduction strategy using both cubic and quadratic polynomials over the simpler strategy that uses only quadratic polynomials. We stress, though, that no method was uniformly superior in our experiments. Additionally, our test set was limited to a particular class of problems, and results on other types of problems may differ. Ideally, one would have several globalizations available to determine which works best in a particular application. Similarly, while the adaptive forcing terms greatly improved the performance of the globalized methods in these experiments, no particular choice of the forcing terms is best for all problems. (Indeed, the small constant value of 10^{-4} was the most effective forcing term among a number of alternatives in a 3D chemical vapor deposition reactor simulation described in [49].) Thus it would be ideal to have available several forcing term choices as well as several globalizations to determine the most effective combination.

Finally, as noted in additional remarks on failure and robustness in §6, there are many factors other than the globalization and forcing terms that may affect the performance of a Newton–Krylov method on problems such as those of interest here, and these should be considered in formulating problems and algorithms to solve them. Additionally, there are other robust solution techniques, such as homotopy, continuation, pseudo-transient continuation, and mesh-sequencing, that should be kept in mind as possible alternatives to globalized Newton–Krylov methods. In particular, if the goal is to traverse or map out a complex nonlinear solution set as problem parameters vary, then some form of continuation is likely to be preferred.

References

[1] E. Allgower and K. Georg. Continuation and path following. *Acta Numerica 1993*, 2:1–64, 1993.

[2] M. Benzi. Preconditioning techniques for large linear systems: A survey. *J. Comp. Physics*, 182:418–477, 2002.

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

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

[5] R. H. Byrd, M. Marazzi, and J. Nocedal. On the convergence of Newton iterations to non-stationary points. Technical Report OTC 2001/01, Optimization Technology Center, Northwestern University, Evanston, IL 60208, Dec. 2002. To appear in Math. Prog.

[6] X.-C. Cai and D. E. Keyes. Nonlinear preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.*, 24:183–202, 2002.

[7] K. A. Cliffe, A. Spence, and S. J. Tavener. The numerical analysis of bifurcation problems with application to fluid mechanics. *Acta Numerica*, pages 39–131, 2000.

[8] I. G. Currie. *Fundametal Mechanics of Fluids*. McGraw-Hill, 1974.

[9] G. De Vahl Davis and C. P. Jones. Natural convection in a square cavity: A comparison exercise. *Int. J. Numer. Methods Fluids*, 3, 1983.

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

[11] R. S. Dembo and T. Steihaug. Truncated Newton algorithms for large-scale optimization. *Math. Prog.*, 26:190–212, 1983.

[12] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[13] S. C. Eisenstat and H. F. Walker. Globally convergent inexact Newton methods. *SIAM J. Optimization*, 4:393–422, 1994.

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

[15] M. Heroux et al. An overview of Trilinos. Technical Report Sand2003-2927, Sandia National Laboratories, Albuquerque NM, 87185, Aug. 2003.

[16] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics*, volume II of *Computational Physics*. Springer-Verlag, Berlin Heidelberg, 1988.

[17] R. W. Freund. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Comput.*, 14:470–482, 1993.

[18] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. *Acta Numerica 1992*, 1:57–100, 1992. Cambridge University Press.

[19] D. K. Gartling. A test problem for outflow boundary conditions — flow over a backward facing step. *Int. J. Numer. Methods Fluids*, 11:953, 1990.

[20] 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.*, 48:387, 1982.

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

[22] M. H. Gutknecht. Lanczos-type solvers for nonsymmetric linear systems of equations. *Acta Numerica 1997*, 6:271–397, 1997. Cambridge University Press.

[23] B. Hendrickson and R. Leland. The Chaco user's guide–version 1.0. Technical Report Sand93-2339, Sandia National Laboratories, Albuquerque NM, 87185, 1993.

[24] M. Heroux. AztecOO: Object-Oriented Aztec Linear Solver Package. http://software.sandia.gov/trilinos/packages/aztecoo/index.html.

[25] M. Heroux. Trilinos solver project. http://software.sandia.gov/trilinos.

[26] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–435, 1952.

[27] A. C. Hindmarsh. LSODE and LSODEI: Two new initial value ordinary differential equation solvers. *ACM Signum Newsletter*, 15(4):10–11, 1980.

[28] 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. *Computer Methods in Applied Mechanics and Engineering*, 73:173–189, 1989.

[29] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*, volume 1 of *Fundamentals of Algorithms*. SIAM, Philadelphia, PA, 2003.

[30] C. T. Kelley and D. E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM J. Numer. Anal.*, 35:508–523, 1998.

[31] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, to appear.

[32] T. G. Kolda and R. P. Pawlowski. NOX nonlinear solver project. http://software.sandia.gov/nox.

[33] M. Kubicek and M. Marek. *Computational Methods in Bifurcation Theory and Dissipative Structures*. Computational Physics. Springer-Verlag, New York, NY, 1983.

[34] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20:286–307, Sept. 1984.

[35] S. G. Nash. *Truncated Newton Methods*. PhD thesis, Computer Science Department, Stanford University, 1982.

[36] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[37] R. P. Pawlowski, J. N. Shadid, J. P. Simonis, and H. F. Walker. Inexact Newton dogleg methods. 2005. Submitted for publication.

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

[39] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114. Gordon and Breach, London, 1970.

[40] Y. Saad. ILUT: a dual threshold incomplete ILU factorization. *Numer. Lin. Alg. Appl.*, 1:387–402, 1994.

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

[42] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual method for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[43] A. G. Salinger, R. B. Lehoucq, R. P. Pawlowski, and J. N. Shadid. Computational bifurcation and stability studies of the 8:1 thermal cavity problem. *Int. J. Num. Meth. Fluids*, 40:1059–1073, 2002.

[44] A.G. Salinger, E.A. Burroughs, R.P. Pawlowski, E.T. Phipps, and L.A. Romero. Bifurcation analysis algorithms for large-scale applications. In preparation for *Int. J. Bif. Chaos*, 2004.

[45] R. Schreiber and H. B. Keller. Driven cavity flows by efficient numerical techniques. *J. Comput. Phys.*, 49:310, 1983.

[46] R. Schreiber and H. B. Keller. Spurious solutions in driven cavity calculations. *J. Comput. Phys.*, 49:165–172, 1983.

[47] J. N. Shadid, S. A. Hutchinson, G. L. Hennigan, H. K. Moffat, K. D. Devine, and A. G. Salinger. Efficient parallel computation of unstructured finite element reacting flow solutions. *Parallel Computing*, 23:1307–1325, 1997.

[48] 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, 87185, May. 1996.

[49] 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.*, 137:155–185, 1997.

[50] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Adv. App. Mech.*, 28:1, 1992.

[51] 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, 87185, Nov. 1999.

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

[53] H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1992.

[54] H. A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge, 2003.

[55] H. F. Walker. An adaptation of Krylov subspace methods to path following problems. *SIAM J. Sci. Comput.*, 21:1191–1198, 1999.

[56] L. T. Watson. Globally convergent homotopy algorithms for nonlinear systems of equations. *Nonlinear Dynamics*, 1:143–191, 1990.

[57] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker. HOMPACK90: a suite of FORTRAN90 codes for globally convergent homotopy algorithms. *ACM Trans. Math. Software*, 23:514–549, 1997.

Appendix

A1 The Moré-Thuente line-search method

Details and outcomes of the interval updating algorithm

The algorithm for updating the intervals of uncertainty proceeds in two stages. The first stage employs an auxiliary function

$$\psi(\lambda) \equiv \phi(\lambda) - \phi(0) - \alpha\phi'(0)\lambda$$

on which the interval updating rules are based, as follows:

> *Updating Algorithm.* Given an iterate λ_+ in the current uncertainty interval with endpoints λ_ℓ and λ_u, update as follows:

1. If $\psi(\lambda_+) > \psi(\lambda_\ell)$, then $\lambda_u \leftarrow \lambda_+$ and λ_ℓ is unchanged.
2. If $\psi(\lambda_+) \leq \psi(\lambda_\ell)$ and $\psi'(\lambda_+)(\lambda_\ell - \lambda_+) \geq 0$, then $\lambda_\ell \leftarrow \lambda_+$ and λ_u is unchanged.
3. If $\psi(\lambda_+) \leq \psi(\lambda_\ell)$ and $\psi'(\lambda_+)(\lambda_\ell - \lambda_+) < 0$, then $\lambda_u \leftarrow \lambda_\ell$ and $\lambda_\ell \leftarrow \lambda_+$.

Note that each occurrence of the second or third case requires the evaluation of $F'(u_k + \lambda_+ s_k)s_k$ as well as $F(u_k + \lambda_+ s_k)$. Even though evaluating this Jacobian-vector product is presumably feasible in the Newton-GMRES context, it may entail significant expense.

It is usual to take $\lambda_\ell = 0$ initially, which we do in our implementation. Then, since $\psi(0) = 0$, it follows from the updating rules that $\psi(\lambda_\ell) \leq 0$ always and, hence, that (2.8) always holds with $\lambda = \lambda_\ell$. It may be that the first case above always holds, and $\lambda_\ell = 0$ always. If this occurs, then the procedures for determining successive values of λ_+ force them to decrease monotonically and reach $\lambda_{\min}$ after a finite number of iterations, and the algorithm fails to determine a point in $[\lambda_{\min}, \lambda_{\max}]$ that satisfies (2.8). (Note that ψ must have a minimizer in $(0, \lambda_{\min})$ in this event.) However, if either the second or third case ever holds, then all subsequent values of λ_ℓ lie in $[\lambda_{\min}, \lambda_{\max}]$ and satisfy (2.8). A possibility is that only the first case holds prior to some iteration, then the third case holds at that iteration, and only the second case holds at all subsequent iterations. If this occurs, then the iterates again decrease monotonically and reach $\lambda_{\min}$ after a finite number of iterations. However, in this event (2.8) holds with $\lambda = \lambda_{\min}$; moreover, we must have $\psi'(\lambda_{\min}) \geq 0$. If $\psi'(\lambda_{\min}) > 0$, then ψ must have a minimizer in $(0, \lambda_{\min})$. In the unlikely event that $\psi'(\lambda_{\min}) = 0$, then $\lambda = \lambda_{\min}$ satisfies not only (2.8) but also (2.9), provided $\beta \geq \alpha$.

Since λ_u plays no role in determining the cases above, it can be initialized to any value. If only the second case holds throughout the iterations, then λ_u is never assigned a new value, and the procedures for determining successive values of λ_+ force them to increase monotonically and reach $\lambda_{\max}$ after a finite number of iterations. (Note that if this occurs, then (2.8) holds with $\lambda = \lambda_{\max}$.) However, if either the first or third case

holds at some iteration, then λ_u is assigned a new value, and clearly $\psi(\lambda_\ell) \leq \psi(\lambda_u)$ for all subsequent iterations. Moreover, one can verify that also $\psi'(\lambda_\ell)(\lambda_u - \lambda_\ell) < 0$ for all subsequent iterations, provided $\psi'(\lambda_\ell)$ is never zero. Indeed, we have $\text{sign}\{\psi'(\lambda_\ell)(\lambda_u - \lambda_\ell)\} = \text{sign}\{\psi'(\lambda_\ell)(\lambda_+ - \lambda_\ell)\}$ for all subsequent iterations and, since $\psi'(0) = (1-\alpha)\phi'(0) < 0$ and, therefore, $\psi'(\lambda_\ell)(\lambda_+ - \lambda_\ell) < 0$ initially, it follows by induction that $\psi'(\lambda_\ell)(\lambda_+ - \lambda_\ell) < 0$ always.

Note that the iterates converge to $\lambda_{\min}$ or $\lambda_{\max}$ only if λ_ℓ and λ_u are never simultaneously in $[\lambda_{\min}, \lambda_{\max}]$. If both λ_ℓ and λ_u are in $[\lambda_{\min}, \lambda_{\max}]$ at some iteration, then they are in $[\lambda_{\min}, \lambda_{\max}]$ for all subsequent iterations.

We summarize these observations as follows (cf. [34, Th. 2.2]): With $\lambda_\ell = 0$ initially, the possible outcomes are the following:

1. The iterates increase monotonically and reach $\lambda_{\max}$ after a finite number of iterations, and (2.8) holds with $\lambda = \lambda_{\max}$.
2. The iterates decrease monotonically and reach $\lambda_{\min}$ after a finite number of iterations. Either ψ has a minimizer in $(0, \lambda_{\min})$ and (2.8) may or may not hold, or $\lambda = \lambda_{\min}$ satisfies both (2.8) and (2.9), provided $\beta \geq \alpha$.
3. Subsequent to some iteration, both λ_ℓ and λ_u always lie in $[\lambda_{\min}, \lambda_{\max}]$ and satisfy $\psi(\lambda_\ell) \leq 0$, $\psi(\lambda_\ell) \leq \psi(\lambda_u)$, and, provided $\psi'(\lambda_\ell)$ is never zero, $\psi'(\lambda_\ell)(\lambda_u - \lambda_\ell) < 0$.

Note that if the third outcome holds with $\psi'(\lambda_\ell)$ never zero, then ultimately ψ has a minimizer between each λ_ℓ and λ_u. If λ_* is such a minimizer, then $\psi(\lambda_*) < 0$ and, therefore, (2.8) holds with $\lambda = \lambda_*$. Moreover, $\psi'(\lambda_*) = 0$, and therefore (2.9) also holds with $\lambda = \lambda_*$, provided $\beta \geq \alpha$. If ever $\psi'(\lambda_\ell) = 0$, then similarly both (2.8) and (2.9) hold with $\lambda = \lambda_\ell$, provided $\beta \geq \alpha$.

A possibility is to continue to iterate until either one of the first two outcomes above occurs or until a value of $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ is reached for which (2.8) holds and (2.9) also holds for all $\beta \geq \alpha$. However, to allow for $\beta < \alpha$, we terminate the first stage of the algorithm if an iterate λ_+ is reached for which $\psi(\lambda_+) \leq 0$ and $\phi'(\lambda_+) \geq \min\{\alpha, \beta\}\phi'(0)$. By slightly modifying the arguments in [34, Th. 3.1], one can verify that the first such λ_+ is greater than λ_ℓ and either (a) satisfies both (2.8) and (2.9), or (b) is such that $\phi'(\lambda_+) > 0$, in which case $[\lambda_\ell, \lambda_+]$ contains a minimizer of ϕ and, therefore, contains points that satisfy both (2.8) and (2.9). Thus it is reasonable to terminate the first stage at this iteration and either declare success, if both (2.8) and (2.9) hold with $\lambda = \lambda_+$, or continue with a second stage, in which we work directly toward finding a minimizer of ϕ.

If continuing, we proceed to iterate as before, but with ψ replaced by ϕ in the Updating Algorithm above. One can verify that, throughout the second stage, $\phi(\lambda_\ell) \leq \phi(\lambda_u)$ and, provided $\phi'(\lambda_\ell)$ is never zero, $\phi'(\lambda_\ell)(\lambda_u - \lambda_\ell) < 0$ as well. It is easy to see that the iterates cannot converge to $\lambda_{\max}$ in the second stage. It is possible for the iterates to converge to $\lambda_{\min}$ in the second stage, but only if λ_ℓ has remained at $\lambda_\ell = 0$ throughout the first stage. If ever $\lambda_{\min} \leq \lambda_\ell \leq \lambda_{\max}$ during the first stage, then λ_ℓ and λ_u will lie in $[\lambda_{\min}, \lambda_{\max}]$

throughout the second stage, and eventually both (2.8) and (2.9) will hold for some iterate in $(\lambda_{\min}, \lambda_{\max})$.

It is noted in [34] that the first stage may never terminate, in which case the iterates converge to a point at which (2.8) holds and (2.9) also holds with $\beta = \alpha$. In this case, the algorithm terminates when $|\lambda_\ell - \lambda_u|$ falls below a specified tolerance.

From the above observations, the possible outcomes can be conveniently summarized as follows:

1. The iterates increase monotonically and reach $\lambda_{\max}$ after a finite number of iterations; with $\lambda = \lambda_{\max}$, (2.8) holds but (2.9) may not hold.
2. The iterates decrease monotonically and reach $\lambda_{\min}$ after a finite number of iterations; neither (2.8) nor (2.9) is guaranteed to hold with $\lambda = \lambda_{\min}$.
3. A value of $\lambda \in (\lambda_{\min}, \lambda_{\max})$ is reached for which (2.8) holds and (2.9) also holds with $\beta = \alpha$.
4. A value of $\lambda \in (\lambda_{\min}, \lambda_{\max})$ is reached for which both (2.8) and (2.9) hold.

Proof of Theorem 2.5

Throughout the proof, s_k denotes for each k the initial inexact Newton step satisfying

$$\|F(u_k) + F'(u_k)\, s_k\| \le \eta_k \|F(u_k)\|,$$

where $0 \le \eta_k \le \eta_{\max} < 1$. For convenience, we assume that $D \in \mathbb{R}^{n\times n}$ is a symmetric, positive-definite matrix for which $\langle u, v\rangle = u^T D v$ for each u, $v \in \mathbb{R}^n$. (Any inner product on $\mathbb{R}^n$ can be represented in this way for a uniquely determined such D.) We take $f(u) \equiv \frac{1}{2}\|F(u)\|^2$ as above and note that $\nabla f(u) = F'(u)^T D F(u)$.[6]

Clearly, ∇f is Lipschitz continuous on $\mathcal{L}(u_0)$ since F' is. Also, since each λ determined by the Moré–Thuente line search satisfies (2.9), it also satisfies the weaker condition

$$\phi'(\lambda) \ge \beta \phi'(0). \tag{A1.1}$$

Then, since f is bounded below by zero, it follows immediately from [12, Th. 6.3.3][7] that either $\nabla f(u_k) = 0$ for some k, or

$$\lim_{k\to\infty} \frac{\nabla f(u_k)^T s_k}{\|s_k\|_2} = 0. \tag{A1.2}$$

In the first case, $F(u_k) = 0$ if $F'(u_k)$ is nonsingular; in any event, the algorithm should terminate at this iteration. Therefore, for the present purposes, we assume that (A1.2) holds.

[6] Here, ∇f is the usual vector of first partial derivatives of f.

[7] It is assumed in Theorem 6.3.3 of [12] that ∇f is Lipschitz continuous on all $\mathbb{R}^n$, but, since $\{u_k\} \subset \mathcal{L}(u_0)$, it is enough to assume this on $\mathcal{L}(u_0)$.

Writing

$$F'(u_k)s_k = -F(u_k) + r_k, \text{ where } \|r_k\| \leq \eta_k\|F(u_k)\| \leq \eta_{\max}\|F(u_k)\|, \tag{A1.3}$$

we have

$$\begin{aligned}\nabla f(u_k)^T s_k &= F(u_k)^T DF'(u_k)s_k = F(u_k)^T D[-F(u_k) + r_k] \\ &\leq -\|F(u_k)\|^2 + \eta_{\max}\|F(u_k)\|^2 \\ &= -(1-\eta_{\max})\|F(u_k)\|^2 < 0,\end{aligned}$$

whence

$$\left|\nabla f(u_k)^T s_k\right| \geq (1-\eta_{\max})\|F(u_k)\|^2. \tag{A1.4}$$

Suppose that $\{u_{k_j}\}$ is a subsequence of $\{u_k\}$ such that each $F'(u_{k_j})$ is nonsingular. Then

$$\begin{aligned}\|s_{k_j}\| &\leq \|F'(u_{k_j})^{-1}\|\,\|F'(u_{k_j})s_{k_j}\| \\ &= \|F'(u_{k_j})^{-1}\|\,\|-F(u_{k_j}) + r_{k_j}\| \\ &\leq \|F'(u_{k_j})^{-1}\|(1+\eta_{\max})\|F(u_{k_j})\|,\end{aligned} \tag{A1.5}$$

and we obtain from (A1.4) and (A1.5) that

$$\frac{\left|\nabla f(u_{k_j})^T s_{k_j}\right|}{\|s_{k_j}\|} \geq \frac{(1-\eta_{\max})}{\|F'(u_{k_j})^{-1}\|(1+\eta_{\max})}\|F(u_{k_j})\|.$$

Since $\|\cdot\|$ and $\|\cdot\|_2$ are equivalent, it follows from (A1.2) that the left-hand side of this inequality goes to zero. Consequently, if $\{\|F'(u_{k_j})^{-1}\|\}$ is bounded, then we must have $F(u_{k_j}) \to 0$ and, since $\{\|F(u_k)\|\}$ is monotone decreasing, $F(u_k) \to 0$ as well.

Now let u_* be a limit point of $\{u_k\}$ such that $F'(u_*)$ is nonsingular. Since u_* is clearly the limit of a subsequence $\{u_{k_j}\}$ such that each $F'(u_{k_j})$ is nonsingular and $\{\|F'(u_{k_j})^{-1}\|\}$ is bounded, we must have $F(u_k) \to 0$ and, in particular, $F(u_*) = 0$. Let $\delta > 0$, $M_J > 0$, $M_{J^{-1}} > 0$, and $M_F > 0$ be such that whenever $\|u - u_*\| \leq \delta$, we have $u \in \mathcal{L}(u_0)$, $\|F'(u)\| \leq M_J$, $\|F'(u)^{-1}\| \leq M_{J^{-1}}$, and $\|F(u)\| \leq M_F$.

Suppose that, for some k, $\|u_k - u_*\| < \delta$ and $\|u_{k+1} - u_*\| < \delta$. Then, with (A1.3) and reasoning as in (A1.5), we have

$$\|s_k\| \leq \|F'(u_k)^{-1}\|(1+\eta_{\max})\|F(u_k)\| \leq M_{J^{-1}}(1+\eta_{\max})\|F(u_k)\|. \tag{A1.6}$$

Also, since (2.8) holds, we have for $u_{k+1} = u_k + \lambda s_k$ that

$$\begin{aligned}\frac{1}{2}\|F(u_{k+1})\|^2 &\leq \frac{1}{2}\|F(u_k)\|^2 + \alpha F(u_k)^T DF'(u_k)s_k\lambda \\ &= \frac{1}{2}\|F(u_k)\|^2 + \alpha F(u_k)^T D\left(-F(u_k) + r_k\right)\lambda \\ &\leq \frac{1}{2}\|F(u_k)\|^2 + \alpha\{-\|F(u_k)\|^2 + F(u_k)^T Dr_k\}\lambda \\ &\leq \frac{1}{2}\|F(u_k)\|^2 - \alpha(1-\eta_{\max})\|F(u_k)\|^2\lambda,\end{aligned}$$

which yields

$$\|F(u_{k+1})\| \leq \{1 - 2\alpha(1 - \eta_{\max})\lambda\}^{1/2}\|F(u_k)\|. \tag{A1.7}$$

Then necessarily

$$\lambda \leq \frac{1}{2\alpha(1 - \eta_{\max})}, \tag{A1.8}$$

which, with (A1.6), gives

$$\|u_{k+1} - u_k\| = \|\lambda s_k\| \leq M_u\|F(u_k)\|, \quad M_u \equiv \frac{M_{J^{-1}}(1 + \eta_{\max})}{2\alpha(1 - \eta_{\max})}. \tag{A1.9}$$

We also establish a positive lower bound on λ. With (A1.1), we have

$$\begin{aligned}(1 - \beta)|\phi'(0)| &= (\beta - 1)\phi'(0) \leq \phi'(\lambda) - \phi'(0) \\ &= F(u_k + \lambda s_k)^T DF'(u_k + \lambda s_k)s_k - F(u_k)^T DF'(u_k)s_k \\ &= \{[F(u_k + \lambda s_k) - F(u_k)]^T DF'(u_k + \lambda s_k) + \\ &\qquad F(u_k)^T D\left[F'(u_k + \lambda s_k) - F'(u_k)\right]\}s_k.\end{aligned}$$

With (2.10) and our assumptions on δ, one verifies that

$$\begin{aligned}(1 - \beta)|\phi'(0)| &\leq \{M_J(\lambda\|s_k\|)M_J + M_F\gamma(\lambda\|s_k\|)\}\,\|s_k\| \\ &= \left(M_J^2 + M_F\gamma\right)\|s_k\|^2\lambda,\end{aligned}$$

which, with (A1.6), implies

$$\lambda \geq \frac{(1 - \beta)|\phi'(0)|}{\left(M_J^2 + M_F\gamma\right)M_{J^{-1}}^2(1 + \eta_{\max})^2\|F(u_k)\|^2}.$$

Since, with (A1.3),

$$\begin{aligned}\phi'(0) &= F(u_k)^T DF'(u_k)s_k = F(u_k)^T D(-F(u_k) + r_k) \\ &\leq -(1 - \eta_{\max})\|F(u_k)\|^2,\end{aligned}$$

we have

$$|\phi'(0)| \geq (1 - \eta_{\max})\|F(u_k)\|^2,$$

and it follows that

$$\lambda \geq \epsilon_\lambda \equiv \frac{(1 - \beta)(1 - \eta_{\max})}{\left(M_J^2 + M_F\gamma\right)M_{J^{-1}}^2(1 + \eta_{\max})^2} > 0. \tag{A1.10}$$

Remark. If we allow $\delta \to 0$, then this lower bound can approach

$$\frac{(1 - \beta)(1 - \eta_{\max})}{\kappa(F'(u_*))(1 + \eta_{\max})^2} < 1,$$

where $\kappa(F'(u_*)) = \|F'(u_*)\| \, \|F'(u_*)^{-1}\|$, while for the upper bound (A1.8), we have $1/2\alpha(1-\eta_{\max}) > 1$. However, this is not to say that (2.8) and (2.9)/(A1.1) ultimately hold with $\lambda = 1$ as $k \to \infty$.

Having (A1.10), we obtain from (A1.7) that

$$\|F(u_{k+1})\| \leq \rho \|F(u_k)\|, \quad \rho \equiv \{1 - 2\alpha(1-\eta_{\max})\epsilon_\lambda\}^{1/2}.$$

Since $\rho \in [0, 1)$, this yields

$$\|F(u_k)\| \leq \frac{1}{1-\rho}[\|F(u_k)\| - \|F(u_{k+1})\|]. \tag{A1.11}$$

Suppose that $\{u_k\}$ does not converge to u_*. Then there is some $\delta_1 \in (0, \delta)$ for which $\|u_k - u_*\| > \delta_1$ for infinitely many values of k. By taking δ_1 smaller if necessary, we can assume that $\|F(u_k)\| < M_u^{-1}(\delta - \delta_1)$ whenever $\|u_k - u_*\| \leq \delta_1$, with M_u defined in (A1.9). Then with (A1.9) we have

$$\|u_{k+1} - u_*\| \leq \|u_k - u_*\| + \|u_{k+1} - u_k\| \leq \delta_1 + M_u\|F(u_k)\| < \delta$$

whenever $\|u_k - u_*\| \leq \delta_1$.

Choose $\delta_2 \in (0, \delta_1)$ and suppose that $\|u_k - u_*\| < \delta_2$. (Since u_* is a limit point of $\{u_k\}$, there are infinitely many such u_k.) Let $\ell \geq 1$ be such that $\|u_{k+j} - u_*\| \leq \delta_1$ for $j = 0, 1, \ldots, \ell - 1$ and $\delta_1 < \|u_{k+\ell} - u_*\| < \delta$. Then with (A1.9) and (A1.11), we obtain

$$\begin{aligned}
\delta_1 - \delta_2 < \|u_{k+\ell} - u_k\| &\leq \sum_{j=0}^{\ell-1} \|u_{k+j+1} - u_{k+j}\| \\
&\leq M_u \sum_{j=0}^{\ell-1} \|F(u_{k+j})\| \\
&\leq \frac{M_u}{1-\rho} \sum_{j=0}^{\ell-1} [\|F(u_{k+j})\| - \|F(u_{k+j+1})\|] \\
&= \frac{M_u}{1-\rho}[\|F(u_k)\| - \|F(u_{k+\ell})\|],
\end{aligned}$$

which yields

$$\|F(u_{k+\ell})\| \leq \|F(u_k)\| - \frac{(1-\rho)(\delta_1 - \delta_2)}{M_u}. \tag{A1.12}$$

Since there are infinitely many such (k, ℓ)-pairs and since $\{\|F(u_k)\|\}$ is monotone decreasing, it follows from (A1.12) that $\|F(u_k)\| \to -\infty$. Since $\|F(u_k)\| \geq 0$ always, this is a contradiction. Hence, $\{u_k\}$ must converge to u_*, and the proof is complete.

Proof of Lemma 2.6

In the following, we use the usual "little oh" convention, e.g., we write $o(\|s_k\|)$ to mean any expression such that $o(\|s_k\|)/\|s_k\| \to 0$ as $\|s_k\| \to 0$. As in the proof of Theorem 2.5, we

write $F'(u_k)s_k = -F(u_k) + r_k$ for each k, where $\|r_k\| \leq \eta_k\|F(u_k)\|$, and we also assume that $D \in \mathbb{R}^{n\times n}$ is a symmetric, positive-definite matrix for which $\langle u, v\rangle = u^T Dv$ for each u, $v \in \mathbb{R}^n$. Then, with $\lambda = 1$, (2.8) becomes

$$\frac{1}{2}\|F(u_k + s_k)\|^2 \leq \frac{1}{2}\|F(u_k)\|^2 + \alpha F(u_k)^T DF'(u_k)s_k. \tag{A1.13}$$

We show below that (A1.13) holds for all sufficiently large k if (2.11) holds and only if (2.12) holds.

For the left-hand side of (A1.13), we have

$$\begin{aligned}\tfrac{1}{2}\|F(u_k + s_k)\|^2 &= \tfrac{1}{2}\|F(u_k) + F'(u_k)s_k + o(\|s_k\|)\|^2 \\ &= \tfrac{1}{2}\|F(u_k) + F'(u_k)s_k\|^2 + \\ &\qquad o(\|F(u_k) + F'(u_k)s_k\|\ \|s_k\|) + o(\|s_k\|^2) \\ &\leq \eta_k^2 \cdot \tfrac{1}{2}\|F(u_k)\|^2 + o(\|F(u_k)\|^2)\end{aligned} \tag{A1.14}$$

for all sufficiently large k. The last inequality in (A1.14) follows since $\|F(u_k)+F'(u_k)s_k\| \leq \eta_k\|F(u_k)\|$ for each k and, for sufficiently large k,

$$\begin{aligned}\|s_k\| &\leq \|F'(u_k)^{-1}\{-F(u_k) + r + k\}\| \\ &\leq \|F'(u_k)^{-1}\|(1 + \eta_k)\|F(u_k)\| \leq 2M\|F(u_k)\|,\end{aligned} \tag{A1.15}$$

where M is a bound on $\|F'(u_k)^{-1}\|$ for all sufficiently large k. For the right-hand side of (A1.13), we have

$$\begin{aligned}\tfrac{1}{2}\|F(u_k)\|^2 + \alpha F(u_k)^T DF'(u_k)s_k &= \tfrac{1}{2}\|F(u_k)\|^2 - \alpha\|F(u_k)\|^2 + \\ &\qquad \alpha F(u_k)^T Dr_k \\ &\geq \tfrac{1}{2}\|F(u_k)\|^2 - \alpha(1 + \eta_k)\|F(u_k)\|^2 \\ &= [1 - 2\alpha(1 + \eta_k)]\ \tfrac{1}{2}\|F(u_k)\|^2.\end{aligned} \tag{A1.16}$$

It follows from (A1.14) and (A1.16) that, for all sufficiently large k, (A1.13) holds if

$$\eta_k^2 \cdot \frac{1}{2}\|F(u_k)\|^2 + o(\|F(u_k)\|^2) \leq [1 - 2\alpha(1 + \eta_k)]\ \frac{1}{2}\|F(u_k)\|^2,$$

which is equivalent to

$$2\alpha(1 + \eta_k) + o(\|F(u_k)\|^2)/\|F(u_k)\|^2 \leq 1 - \eta_k^2 = (1 - \eta_k)(1 + \eta_k),$$

which is in turn equivalent to

$$\alpha + o(\|F(u_k)\|^2)/\|F(u_k)\|^2 \leq \frac{1 - \eta_k}{2}.$$

Since $F(u_k) \to 0$ and, hence, $o(\|F(u_k)\|^2)/\|F(u_k)\|^2 \to 0$, this last inequality and, therefore, (A1.13) hold for all sufficiently large k if (2.11) holds.

Now suppose that (A1.13) holds for all sufficiently large k. Then for all sufficiently large k, we have

$$\begin{aligned} 0 &\le \tfrac{1}{2}\|F(u_k+s_k)\|^2 \le \tfrac{1}{2}\|F(u_k)\|^2 + \alpha F(u_k)^T DF'(u_k)s_k \\ &= \tfrac{1}{2}\|F(u_k)\|^2 - \alpha\|F(u_k)\|^2 + \alpha F(u_k)^T Dr_k \\ &\le [1-2\alpha(1-\eta_k)]\,\tfrac{1}{2}\|F(u_k)\|^2. \end{aligned}$$

It follows that $\alpha \le 1/[2(1-\eta_k)]$ for all sufficiently large k, which implies (2.12).

To complete the proof, note that, with $\lambda = 1$, (2.9) becomes

$$\left|F(u_k+s_k)^T DF'(u_k+s_k)s_k\right| \le \beta \left|F(u_k)^T DF'(u_k)s_k\right| \tag{A1.17}$$

For all sufficiently large k, we have

$$\begin{aligned} \left|F(u_k+s_k)^T DF'(u_k+s_k)s_k\right| &= |[F(u_k)+F'(u_k)s_k + o(\|s_k\|)]^T D \\ &\qquad [F'(u_k)+o(\|s_k\|)]\,s| \\ &\le |(F(u_k)+F'(u_k)s_k)^T DF'(u_k)s| + o(\|s_k\|^2) \\ &\le \eta_k\|F(u_k)\|\,\|-F(u_k)+r_k\| + o(\|s_k\|^2) \\ &\le \eta_k(1+\eta_k)\|F(u_k)\|^2 + o(\|F(u_k)\|^2), \end{aligned} \tag{A1.18}$$

where (A1.15) is used to replace $o(\|s_k\|^2)$ with $o(\|F(u_k)\|^2)$ in the last inequality. We also have

$$F(u_k)^T DF'(u_k)s_k = -\|F(u_k)\|^2 + F(u_k)^T Dr_k \le -(1-\eta_k)\|F(u_k)\|^2 \le 0,$$

whence

$$\|F(u_k)\|^2 \le \frac{|F(u_k)^T DF'(u_k)s_k|}{1-\eta_k} \le \frac{|F(u_k)^T DF'(u_k)s_k|}{1-\eta_{\max}}.$$

Then (A1.18) yields

$$\begin{aligned} \left|F(u_k+s_k)^T DF'(u_k+s_k)s_k\right| \le \frac{\eta_k(1+\eta_k)}{1-\eta_k}\left|F(u_k)^T DF'(u_k)s_k\right| + \\ o(\left|F(u_k)^T DF'(u_k)s_k\right|^2), \end{aligned}$$

and it follows that (2.9) holds for all sufficiently large k if (2.13) is satisfied. This completes the proof.

A2 The dogleg method: proof of Theorem 2.7

Let $\{u_k\}$ be produced by Algorithm INDL. For convenience, we assume that $F(u_k) \ne 0$ for all k and also that, as in the proof of Theorem 2.5 in §A1.1.2 above, $D \in \mathbb{R}^{n\times n}$ is a symmetric, positive-definite matrix for which $\langle u, v\rangle = u^T Dv$ for each $u,\ v \in \mathbb{R}^n$. A

straightforward but tedious calculation yields

$$\begin{aligned} \|s_k^{\mathrm{CP}}\| &= \frac{\left\langle F(u_k), F'(u_k)F'(u_k)^T DF(u_k)\right\rangle}{\|F'(u_k)F'(u_k)^T DF(u_k)\|^2}\|F'(u_k)^T DF(u_k)\|, \\ &= \frac{\|F'(u_k)^T DF(u_k)\|_2^2}{\|F'(u_k)F'(u_k)^T DF(u_k)\|^2}\|F'(u_k)^T DF(u_k)\|, \end{aligned} \tag{A2.19}$$

$$\begin{aligned} \eta_k^{\mathrm{CP}} &\equiv \frac{\|F(u_k) + F'(u_k)\, s_k^{\mathrm{CP}}\|}{\|F(u_k)\|} \\ &= \sqrt{1 - \frac{\langle F(u_k), F'(u_k)F'(u_k)^T DF(u_k)\rangle^2}{\|F(u_k)\|^2\|F'(u_k)F'(u_k)^T DF(u_k)\|^2}} \\ &= \sqrt{1 - \frac{\|F'(u_k)^T DF(u_k)\|_2^4}{\|F(u_k)\|^2\|F'(u_k)F'(u_k)^T DF(u_k)\|^2}}, \end{aligned} \tag{A2.20}$$

provided none of the denominators is zero.

Let u_* be a limit point of $\{u_k\}$, and suppose u_* is not a stationary point of $\|F\|$. Then $F(u_*) \neq 0$, and we claim that, in addition, $F'(u_*)^T DF(u_*) \neq 0$ and $F'(u_*)F'(u_*)^T DF(u_*) \neq 0$. Indeed, one sees that

$$F'(u_*)^T DF(u_*) = \nabla_s \left(\frac{1}{2}\|F(u_*) + F'(u_*)\, s\|^2\right)\Big|_{s=0},$$

which must be non-zero since u_* is not a stationary point of $\|F\|$. Then

$$\left\langle F(u_*), F'(u_*)F'(u_*)^T DF(u_*)\right\rangle = \|F'(u_*)^T DF(u_*)\|_2^2 \neq 0,$$

which implies that $F'(u_*)F'(u_*)^T DF(u_*) \neq 0$.

It follows from these observations and (A2.19)-(A2.20) that there is a neighborhood N_* of u_* and bounds M and $\eta_{\max}^{\mathrm{CP}} < 1$ such that $\|s_k^{\mathrm{CP}}\| \leq M$ and $\eta_k^{\mathrm{CP}} \leq \eta_{\max}^{\mathrm{CP}}$ whenever $u_k \in N_*$. Then for $u_k \in N_*$ and s_k determined by Algorithm DL, there are three cases, as follows:

Case 1. $s_k = s_k^{\mathrm{IN}}$. In this case, $\|F(u_k) + F'(u_k)\, s_k\| \leq \eta_k\|F(u_k)\| \leq \eta_{\max}\|F(u_k)\|$.

Case 2. s_k lies on Γ_k^{DL} between s_k^{CP} and s_k^{IN}. In this case, it follows from norm convexity that

$$\|F(u_k) + F'(u_k)\, s_k\| \leq \max\{\eta_k^{\mathrm{CP}}, \eta_k\}\|F(u_k)\| \leq \max\{\eta_{\max}^{\mathrm{CP}}, \eta_{\max}\}\|F(u_k)\|.$$

Case 3. s_k lies on Γ_k^{DL} between 0 and s_k^{CP}. In this case, $\delta_{\min} \leq \delta = \|s_k\| \leq \|s_k^{\mathrm{CP}}\|$. Since

the local linear model norm is monotone decreasing along this segment of Γ_k^{DL}, we have

$$\begin{aligned}\|F(u_k) + F'(u_k)\, s_k\| &\le \|F(u_k) + F'(u_k)\, \left(\frac{\delta_{\min}}{\|s_k^{\mathrm{CP}}\|} s_k^{\mathrm{CP}}\right)\| \\ &\le \left(1 - \frac{\delta_{\min}}{\|s_k^{\mathrm{CP}}\|}\right) \|F(u_k)\| + \frac{\delta_{\min}}{\|s_k^{\mathrm{CP}}\|} \|F(u_k) + F'(u_k)\, s_k^{\mathrm{CP}}\| \\ &= \left[1 - \frac{\delta_{\min}}{\|s_k^{\mathrm{CP}}\|}(1 - \eta_k^{\mathrm{CP}})\right] \|F(u_k)\| \\ &\le \left[1 - \frac{\delta_{\min}}{M}(1 - \eta_{\max}^{\mathrm{CP}})\right] \|F(u_k)\|.\end{aligned}$$

One concludes that, whenever $u_k \in N_*$, we have $\|F(u_k) + F'(u_k)\, s_k\| \le \bar{\eta} \|F(u_k)\|$, where

$$\bar{\eta} \equiv \max\{\eta_{\max}, \eta_{\max}^{\mathrm{CP}}, 1 - \frac{\delta_{\min}}{M}(1 - \eta_{\max}^{\mathrm{CP}})\} < 1, \tag{A2.21}$$

and therefore, with $pred_k$ defined as in (2.6),

$$\frac{pred_k}{\|F(u_k)\|} = \frac{\|F(u_k)\| - \|F(u_k) + F'(u_k)\, s_k\|}{\|F(u_k)\|} \ge (1 - \bar{\eta}) > 0. \tag{A2.22}$$

Since $u_k \in N_*$ for infinitely many values of k, (A2.22) implies that $\sum_{k=0}^{\infty} pred_k / \|F(u_k)\|$ diverges, and it follows from [13, Cor. 3.6] that $F(u_*) = 0$. This is a contradiction; therefore, u_* must be a stationary point of $\|F\|$.

Suppose now that $F'(u_*)$ is nonsingular. Then, since u_* is a stationary point of $\|F\|$, we must have $F(u_*) = 0$. In addition, it follows immediately from (A2.19) that there is again a neighborhood N_* of u_* and a bound M such that $\|s_k^{\mathrm{CP}}\| \le M$ whenever $u_k \in N_*$. Furthermore, we have for some $c > 0$ that

$$\liminf_{u \to u_*, u \ne u_*} \frac{\|F'(u)^T DF(u)\|_2^4}{\|F(u)\|^2 \|F'(u)F'(u)^T DF(u)\|^2} \ge c \liminf_{u \to u_*, u \ne u_*} \frac{\|F'(u)^T DF(u)\|_2^4}{\|F(u)\|_2^4} > 0.$$

Consequently, we can assume that N_* is chosen so that there is again a bound $\eta_{\max}^{\mathrm{CP}} < 1$ for which $\eta_k^{\mathrm{CP}} \le \eta_{\max}^{\mathrm{CP}}$ whenever $u_k \in N_*$.

Repeating the earlier argument, one verifies that (A2.22) again holds with $\bar{\eta}$ given by (A2.21), and we have as before that $\sum_{k=0}^{\infty} pred_k / \|F(u_k)\|$ diverges. Since $F'(u_*)$ is nonsingular, it follows from [13, Cor. 3.6] that $u_k \to u_*$.

To complete the proof, we write $F'(u_k)\, s_k = -F(u_k) + r_k$ with $\|r_k\| \le \eta_k \|F(u_k)\|$ and note that

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

provided $F'(u_k)$ is nonsingular. Since $u_k \to u_*$ with $F(u_*) = 0$ and $F'(u_*)$ nonsingular, it follows that $s_k^{\mathrm{IN}} \to 0$ as $k \to \infty$. In particular, for all sufficiently large k, we have $\|s_k^{\mathrm{IN}}\| \le \delta_{\min}$ and, therefore, $s_k = s_k^{\mathrm{IN}}$.

A3 Test Details and Results

The following tables provide full details of the test results underlying the summary results in §5. In the table headings, "Cubic" and "Quadratic" refer, respectively, to the algorithms designated as "Backtracking, Quadratic/Cubic" and "Backtracking, Quadratic Only" in §5. Other aspects of the table headings should be clear. The column headings in each table are as follows:

S/F	A flag indicating success or failure. "0" indicates that a solution was successfully found; "-1" indicates a failure to find a solution.
INS	The total number of inexact Newton steps carried out.
FE	The number of function evaluations performed.
LS	The number of inexact Newton steps for which a backtrack or line search was invoked (backtracking/line-search methods only).
f-LS	The number of backtracks or line searches that failed to find a suitable step (backtracking/line-search methods only). In these cases, the original inexact Newton step was taken.
Bkt	The total number of step reductions used by the backtracking or line-search methods (backtracking/line-search methods only).
0:C	The number of dogleg steps between zero and the Cauchy point (dogleg method only).
C:IN	The number of dogleg steps between the Cauchy point and the inexact Newton step (dogleg method only).
IN	The number of dogleg steps that were the inexact Newton step (dogleg method only).
GMRES	The total number of GMRES iterations.
$\|F\|$	The final 2-norm of the residual vector. One of several termination criteria in our algorithm is that $\|F(u_k)\| \leq 10^{-2}\|F(u_0)\|_2$ for some x_k.
Time	The run time (in seconds) required to reach a solution.

In the case of the "Full Step" method, the "NA" listings in the "LS", "f-LS", and "Bkt" columns indicate that the corresponding values were not computed.

Thermal Convection 2D

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	6	7	0	0	0	1106	$2.83e-15$	85
$Ra\_1.0e04$	0	9	10	0	0	0	1209	$1.57e-14$	87
$Ra\_1.0e05$	0	14	16	1	0	1	801	$2.42e-09$	57
$Ra\_1.0e06$	0	41	65	19	0	23	2259	$3.21e-10$	160

Cubic Constant

	S/F	INS	FE	LS	f −LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	4	5	0	0	0	870	$3.28e-12$	64
$Ra\_1.0e04$	0	6	7	0	0	0	1349	$6.8e-11$	97
$Ra\_1.0e05$	0	8	10	1	0	1	2106	$6.21e-14$	154
$Ra\_1.0e06$	0	11	24	6	0	12	3353	$7.44e-12$	247

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	6	7	0	0	0	1106	$2.83e-15$	80
$Ra\_1.0e04$	0	9	10	0	0	0	1209	$1.57e-14$	87
$Ra\_1.0e05$	0	14	16	1	0	1	801	$2.42e-09$	57
$Ra\_1.0e06$	0	44	68	20	0	23	2595	$1.55e-10$	183

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	4	5	0	0	0	870	$3.28e-12$	65
$Ra\_1.0e04$	0	6	7	0	0	0	1349	$6.8e-11$	98
$Ra\_1.0e05$	0	8	10	1	0	1	2106	$6.21e-14$	152
$Ra\_1.0e06$	0	14	30	9	0	15	4228	$4.15e-13$	306

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	6	19	0	0	0	1106	$2.83e-15$	81
$Ra\_1.0e04$	0	9	28	0	0	0	1209	$1.57e-14$	91
$Ra\_1.0e05$	0	13	44	1	0	2	917	$2.87e-08$	76
$Ra\_1.0e06$	0	37	156	18	0	22	2436	$6.6e-13$	212

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	4	13	0	0	0	870	$3.28e-12$	63
$Ra\_1.0e04$	0	6	19	0	0	0	1349	$6.8e-11$	94
$Ra\_1.0e05$	0	8	31	2	0	3	2321	$5.13e-14$	160
$Ra\_1.0e06$	0	11	62	7	0	14	3450	$5.7e-11$	245

Dogleg Choice 1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e3$	0	5	6	0	0	5	1764	$1.31e-13$	111
$Ra\_1.0e4$	0	13	14	0	0	13	3077	$2.17e-12$	199
$Ra\_1.0e5$	0	20	23	0	3	17	4338	$2.46e-10$	280
$Ra\_1.0e6$	0	44	59	0	25	19	4893	$1.76e-12$	317

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	4	5	0	0	4	870	$3.28e-12$	62
$Ra\_1.0e04$	0	6	7	0	0	6	1349	$6.8e-11$	93
$Ra\_1.0e05$	0	8	11	0	4	4	2913	$7.89e-14$	203
$Ra\_1.0e06$	-1	200	267	0	200	0	102181	0.595	7116

Full Step Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	6	7	NA	NA	NA	1106	$2.83e-15$	77
$Ra\_1.0e04$	0	9	10	NA	NA	NA	1209	$1.57e-14$	85
$Ra\_1.0e05$	0	25	26	NA	NA	NA	1408	$3.99e-14$	112
$Ra\_1.0e06$	-1	100	101	NA	NA	NA	3859	$6.6e+05$	327

Full Step Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	4	5	NA	NA	NA	870	$3.28e-12$	62
$Ra\_1.0e04$	0	6	7	NA	NA	NA	1349	$6.8e-11$	93
$Ra\_1.0e05$	0	11	12	NA	NA	NA	2665	$9.98e-11$	186
$Ra\_1.0e06$	-1	50	51	NA	NA	NA	14397	$5.09e+04$	1021

Backward Facing Step 2D

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	10	11	0	0	0	459	$4.44e-14$	18
Re_200	0	12	14	1	0	1	535	$1.59e-13$	22
Re_300	0	14	17	2	0	2	650	$1.62e-17$	28
Re_400	0	51	77	25	0	25	1348	$2.17e-15$	58
Re_500	0	60	90	29	0	29	1573	$1.93e-12$	71
Re_600	0	81	126	43	0	44	2459	$9.91e-14$	102
Re_700	0	124	207	82	0	82	4635	$8e-16$	193
Re_750	-1	200	660	199	0	459	2379	0.000244	160
Re_800	0	162	294	130	0	131	5905	$1.33e-13$	238

Cubic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	0	595	$1.09e-14$	24
Re_200	0	9	10	0	0	0	1047	$1.25e-15$	49
Re_300	0	9	11	1	0	1	1095	$5.05e-16$	53
Re_400	0	11	16	4	0	4	1441	$1.08e-13$	67
Re_500	0	9	12	2	0	2	1145	$2.73e-15$	53
Re_600	0	11	16	4	0	4	1506	$7.14e-15$	78
Re_700	0	12	18	5	0	5	1745	$1.67e-13$	88
Re_750	0	29	61	22	0	31	4729	$1.4e-15$	263
Re_800	0	31	66	25	0	34	5227	$3.74e-13$	285

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	10	11	0	0	0	459	$4.44e-14$	18
*Re*_200	0	12	14	1	0	1	535	$1.59e-13$	22
*Re*_300	0	14	17	2	0	2	650	$1.62e-17$	30
*Re*_400	0	51	77	25	0	25	1348	$2.17e-15$	59
*Re*_500	0	60	90	29	0	29	1573	$1.93e-12$	68
*Re*_600	0	98	161	61	0	62	3193	$1.39e-15$	131
*Re*_700	0	124	207	82	0	82	4635	$8e-16$	184
*Re*_750	0	142	240	93	0	97	5116	$4.76e-15$	204
*Re*_800	0	130	218	85	0	87	4837	$4.04e-11$	193

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	7	0	0	0	595	$1.09e-14$	24
*Re*_200	0	9	10	0	0	0	1047	$1.25e-15$	46
*Re*_300	0	9	11	1	0	1	1095	$5.05e-16$	49
*Re*_400	0	11	16	4	0	4	1441	$1.08e-13$	68
*Re*_500	0	9	12	2	0	2	1145	$2.73e-15$	54
*Re*_600	0	11	16	4	0	4	1506	$7.14e-15$	72
*Re*_700	0	12	18	5	0	5	1745	$1.67e-13$	87
*Re*_750	−1	200	936	198	0	735	38510	$5.51e-05$	2283
*Re*_800	0	44	104	37	0	59	7634	$1.17e-16$	421

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	10	31	0	0	0	459	$4.44e-14$	23
*Re*_200	0	11	36	1	0	1	581	$2.96e-17$	29
*Re*_300	0	18	67	6	0	6	721	$3.82e-14$	40
*Re*_400	0	38	151	18	0	18	1067	$1.58e-13$	71
*Re*_500	0	69	290	39	0	41	1650	$2.73e-13$	121
*Re*_600	0	80	333	45	0	46	2363	$1.5e-15$	154
*Re*_700	0	87	370	52	0	54	3043	$2.49e-16$	184
*Re*_750	0	111	480	67	0	73	4091	$4.07e-13$	242
*Re*_800	−1	200	1743	199	0	571	4721	0.000265	473

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	19	0	0	0	595	$1.09e-14$	24
*Re*_200	0	7	24	1	0	1	754	$1.16e-13$	32
*Re*_300	0	9	30	1	0	1	1090	$1.08e-13$	48
*Re*_400	0	8	29	2	0	2	981	$8.02e-14$	44
*Re*_500	0	9	32	2	0	2	1158	$4.49e-16$	53
*Re*_600	0	10	37	3	0	3	1362	$1.72e-15$	64
*Re*_700	0	11	42	4	0	4	1593	$4.13e-15$	78
*Re*_750	0	11	42	4	0	4	1584	$7.8e-15$	78
*Re*_800	0	13	56	6	0	8	1955	$1.61e-13$	98

Dogleg Choice 1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_100	0	8	9	0	0	8	408	$4.45e-14$	15
Re_200	0	11	12	0	0	11	461	$5.68e-13$	19
Re_300	0	17	19	0	2	15	728	$5.61e-14$	28
Re_400	0	20	23	0	4	16	833	$8.79e-14$	31
Re_500	0	28	31	0	5	23	1024	$2.68e-13$	39
Re_600	0	83	96	0	63	20	3388	$4.69e-12$	117
Re_700	0	92	106	0	70	22	4276	$6.58e-16$	147
Re_750	0	105	117	0	81	24	4950	$1.94e-11$	164
Re_800	0	125	146	0	104	21	6883	$7.09e-14$	232

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	6	595	$1.09e-14$	23
Re_200	0	9	10	0	0	9	1047	$1.25e-15$	44
Re_300	0	8	10	0	2	6	894	$5.88e-13$	36
Re_400	0	16	18	0	6	10	2095	$1.28e-15$	92
Re_500	0	18	22	0	11	7	2580	$2.72e-14$	120
Re_600	0	15	21	0	6	9	2122	$4.61e-16$	98
Re_700	0	21	27	0	14	7	3246	$1.58e-16$	158
Re_750	0	20	27	0	13	7	3127	$1.6e-13$	153
Re_800	0	23	33	0	17	6	3710	$1.83e-13$	185

Full Step Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	10	11	NA	NA	NA	459	$4.44e-14$	20
Re_200	0	14	15	NA	NA	NA	561	$8.12e-17$	24
Re_300	0	19	20	NA	NA	NA	616	$6.46e-16$	29
Re_400	0	38	39	NA	NA	NA	822	$1.16e-13$	40
Re_500	−1	200	201	NA	NA	NA	17158	0.658	856
Re_600	−1	200	201	NA	NA	NA	12391	86.1	610
Re_700	−1	200	201	NA	NA	NA	21379	2.08	1096
Re_750	−1	200	201	NA	NA	NA	13867	5.39	653
Re_800	−1	200	201	NA	NA	NA	14333	1.17	705

Full Step Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	NA	NA	NA	595	$1.09e-14$	26
Re_200	0	9	10	NA	NA	NA	1047	$1.25e-15$	45
Re_300	−1	200	201	NA	NA	NA	33846	9.27	1840
Re_400	−1	200	201	NA	NA	NA	25806	84.5	1344
Re_500	−1	200	201	NA	NA	NA	36936	0.533	2031
Re_600	−1	200	201	NA	NA	NA	28595	1.95	1527
Re_700	−1	200	201	NA	NA	NA	34524	5.88	1844
Re_750	−1	200	201	NA	NA	NA	57827	0.221	3303
Re_800	−1	200	201	NA	NA	NA	2583	0.126	166

Lid Driven Cavity 2D

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	24	27	2	0	2	799	$6.77e-11$	44
Re_2000	0	33	42	8	0	8	1632	$4.61e-13$	88
Re_3000	0	52	66	13	0	13	2138	$8.31e-13$	118
Re_4000	0	57	73	14	0	15	2230	$1.74e-07$	118
Re_5000	0	58	79	18	0	20	2808	$6.23e-10$	162
Re_6000	0	75	106	28	0	30	3471	$3.45e-10$	184
Re_7000	0	94	148	50	0	53	4984	$6.47e-12$	264
Re_8000	0	106	165	55	0	58	5535	$6.47e-11$	314
Re_9000	0	150	260	106	0	109	7695	$5.73e-12$	421
Re_10000	0	160	279	115	0	118	8581	$6.89e-06$	442

Cubic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	12	18	5	0	5	1621	$1.79e-13$	100
Re_2000	−1	300	1861	246	22	1559	77599	$5.51e+14$	5601
Re_3000	−1	300	1851	287	15	1550	127980	$3.77e+13$	9122
Re_4000	−1	300	3521	293	65	3220	77090	$2.02e+14$	5550
Re_5000	−1	300	4900	298	0	4599	170847	88	12261
Re_6000	−1	300	2624	299	0	2323	175767	294	12523
Re_7000	−1	300	2676	298	0	2375	172883	146	12337
Re_8000	−1	300	4273	299	0	3972	172634	196	12327
Re_9000	−1	300	2129	299	0	1828	177490	480	12733
Re_10000	−1	300	2158	299	0	1857	174191	499	12330

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	24	27	2	0	2	799	$6.77e-11$	45
Re_2000	0	33	42	8	0	8	1632	$4.61e-13$	90
Re_3000	0	52	66	13	0	13	2138	$8.31e-13$	125
Re_4000	0	52	66	12	0	13	2611	$3.44e-12$	139
Re_5000	0	60	78	16	0	17	2722	$4.55e-06$	140
Re_6000	0	76	110	32	0	33	3303	$1.62e-09$	191
Re_7000	0	101	164	61	0	62	5119	$2.73e-09$	264
Re_8000	0	140	248	106	0	107	6881	$4.98e-09$	357
Re_9000	0	143	242	97	0	98	7833	$4.54e-10$	426
Re_10000	0	163	284	119	0	120	8950	$2.34e-08$	476

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	12	18	5	0	5	1621	$1.79e-13$	102
Re_2000	−1	300	1738	298	0	1437	53931	15.6	3759
Re_3000	−1	300	1746	298	0	1445	110022	63.3	7819
Re_4000	−1	300	1712	298	0	1411	85748	42.8	5646
Re_5000	−1	300	1693	298	0	1392	170958	147	12184
Re_6000	−1	300	1979	299	0	1678	174141	520	12333
Re_7000	−1	300	1579	299	0	1278	174954	284	12312
Re_8000	−1	300	1605	299	0	1304	119613	244	8245
Re_9000	−1	300	1732	299	0	1431	176946	997	12524
Re_10000	−1	300	1878	299	0	1577	169707	$2.39e+03$	11982

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	18	63	4	0	4	923	$1.72e-10$	62
Re_2000	0	31	112	9	0	9	1341	$8.87e-07$	94
Re_3000	0	42	159	15	0	16	1759	$5.36e-08$	126
Re_4000	0	50	187	17	0	18	2361	$7.72e-10$	164
Re_5000	0	57	214	20	0	21	2789	$4.47e-07$	188
Re_6000	0	72	285	32	0	34	3572	$4.21e-10$	249
Re_7000	0	82	337	43	0	45	3912	$7.72e-06$	272
Re_8000	0	91	396	59	0	61	4636	$3.42e-10$	333
Re_9000	0	106	461	68	0	71	5609	$8.75e-09$	397
Re_10000	0	119	528	83	0	85	5996	$4.34e-05$	416

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	11	40	3	0	3	1475	$1.13e-13$	86
Re_2000	−1	300	3261	298	0	1180	53161	14	3916
Re_3000	−1	300	1794	272	0	447	129272	$3.9e+13$	8848
Re_4000	−1	300	1676	212	1	387	48967	$2.07e+13$	3422
Re_5000	−1	300	1632	204	0	366	37935	$4.66e+13$	2759
Re_6000	−1	300	3837	298	0	1468	171041	108	11966
Re_7000	−1	300	3737	299	0	1418	173699	298	12147
Re_8000	−1	300	3661	298	0	1380	176236	255	12301
Re_9000	−1	300	3651	299	0	1375	176592	746	12347
Re_10000	−1	300	2751	300	0	925	178988	324	12333

Dogleg Choice 1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_1000	0	22	24	0	2	20	1298	$2.52e-12$	68
Re_2000	0	35	37	0	6	29	1792	$1.72e-07$	87
Re_3000	0	40	46	0	13	27	2519	$1.3e-12$	125
Re_4000	0	69	86	0	35	34	3392	$3e-09$	164
Re_5000	0	93	121	0	57	36	4562	$1.25e-09$	226
Re_6000	0	99	136	0	70	29	5709	$7.24e-09$	280
Re_7000	0	102	136	0	60	42	6629	$7.46e-12$	334
Re_8000	0	128	178	0	98	30	9421	$4.31e-09$	488
Re_9000	0	121	150	0	84	37	7434	$1.59e-10$	384
Re_10000	0	132	170	0	103	29	8546	$1.16e-09$	453

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_1000	0	19	27	0	13	6	2672	$1.38e-11$	153
Re_2000	-1	300	493	0	299	1	50053	6.87	3124
Re_3000	-1	300	347	0	299	1	56876	21.2	3806
Re_4000	-1	300	402	0	299	1	113365	37.1	7397
Re_5000	-1	300	353	0	299	1	101634	29.9	6572
Re_6000	-1	300	373	0	299	1	70575	38.1	4642
Re_7000	-1	300	420	0	299	1	145341	60.7	9575
Re_8000	-1	300	391	0	299	1	174586	194	11617
Re_9000	-1	300	407	0	299	1	146727	74	9653
Re_10000	-1	300	429	0	299	1	141779	81.5	9294

Full Step Choice 1

	S/F	INS	FE	LS	f$-$LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	0	20	21	NA	NA	NA	787	$2.02e-13$	49
Re_2000	-1	300	301	NA	NA	NA	3930	$6.01e+05$	392
Re_3000	-1	300	301	NA	NA	NA	19901	$6.4e+04$	1278
Re_4000	-1	300	301	NA	NA	NA	7688	$2.32e+05$	607
Re_5000	-1	300	301	NA	NA	NA	15935	$1.14e+05$	1003
Re_6000	-1	300	301	NA	NA	NA	22103	$2.53e+05$	1429
Re_7000	-1	300	301	NA	NA	NA	16456	$3.05e+05$	1046
Re_8000	-1	300	301	NA	NA	NA	29532	$2.51e+05$	1855
Re_9000	-1	300	301	NA	NA	NA	14323	$2.69e+06$	996
Re_10000	-1	300	301	NA	NA	NA	26305	$4.02e+06$	1725

Full Step Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_1000	-1	300	301	NA	NA	NA	24845	$4.27e+05$	1635
Re_2000	-1	300	301	NA	NA	NA	11135	$2.21e+05$	779
Re_3000	-1	300	301	NA	NA	NA	15520	$2.96e+05$	1030
Re_4000	-1	300	301	NA	NA	NA	13091	$1.27e+05$	929
Re_5000	-1	300	301	NA	NA	NA	21049	$9.14e+04$	1406
Re_6000	-1	300	301	NA	NA	NA	37531	$8.73e+04$	2404
Re_7000	-1	300	301	NA	NA	NA	24886	$6.34e+05$	1580
Re_8000	-1	300	301	NA	NA	NA	29118	$2.1e+06$	1870
Re_9000	-1	300	301	NA	NA	NA	23179	$1.86e+06$	1565
Re_10000	-1	300	301	NA	NA	NA	12180	$1.65e+05$	874

Lid Driven Cavity 2D (Low Reynolds Numbers)

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	8	9	0	0	0	597	$3.79e-15$	32
Re_200	0	9	10	0	0	0	485	$1.87e-10$	27
Re_300	0	10	12	1	0	1	608	$1.68e-14$	35
Re_400	0	12	15	2	0	2	621	$6.32e-14$	35
Re_500	0	16	18	1	0	1	703	$3.69e-13$	40
Re_600	0	16	19	2	0	2	671	$1.56e-13$	42
Re_700	0	16	19	2	0	2	671	$1.29e-12$	40
Re_800	0	20	23	2	0	2	757	$1.59e-11$	44
Re_900	0	17	21	3	0	3	573	$1.76e-07$	32
Re_1000	0	24	27	2	0	2	799	$6.77e-11$	49

Cubic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	0	714	$3.2e-15$	40
Re_200	0	7	8	0	0	0	867	$7.28e-15$	49
Re_300	0	7	8	0	0	0	871	$1.01e-12$	53
Re_400	0	8	9	0	0	0	1016	$4.88e-14$	59
Re_500	0	9	11	1	0	1	1125	$1.78e-13$	64
Re_600	0	9	11	1	0	1	1126	$3.18e-12$	65
Re_700	0	17	27	7	0	9	2332	$8.78e-14$	139
Re_800	0	10	12	1	0	1	1322	$7.36e-14$	77
Re_900	0	10	12	1	0	1	1267	$6.9e-13$	72
Re_1000	0	12	18	5	0	5	1621	$1.79e-13$	95

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	8	9	0	0	0	597	$3.79e-15$	35
*Re*_200	0	9	10	0	0	0	485	$1.87e-10$	27
*Re*_300	0	10	12	1	0	1	608	$1.68e-14$	32
*Re*_400	0	12	15	2	0	2	621	$6.32e-14$	38
*Re*_500	0	16	18	1	0	1	703	$3.69e-13$	40
*Re*_600	0	16	19	2	0	2	671	$1.56e-13$	39
*Re*_700	0	16	19	2	0	2	671	$1.29e-12$	42
*Re*_800	0	20	23	2	0	2	757	$1.59e-11$	44
*Re*_900	0	17	21	3	0	3	573	$1.76e-07$	32
*Re*_1000	0	24	27	2	0	2	799	$6.77e-11$	46

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	7	0	0	0	714	$3.2e-15$	40
*Re*_200	0	7	8	0	0	0	867	$7.28e-15$	49
*Re*_300	0	7	8	0	0	0	871	$1.01e-12$	49
*Re*_400	0	8	9	0	0	0	1016	$4.88e-14$	58
*Re*_500	0	9	11	1	0	1	1125	$1.78e-13$	69
*Re*_600	0	9	11	1	0	1	1126	$3.18e-12$	65
*Re*_700	0	14	22	6	0	7	1881	$1.53e-12$	110
*Re*_800	0	10	12	1	0	1	1322	$7.36e-14$	82
*Re*_900	0	10	12	1	0	1	1267	$6.9e-13$	73
*Re*_1000	0	12	18	5	0	5	1621	$1.79e-13$	103

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	8	25	0	0	0	597	$3.75e-15$	36
*Re*_200	0	9	28	0	0	0	485	$1.87e-10$	32
*Re*_300	0	9	30	1	0	1	560	$1.52e-10$	36
*Re*_400	0	11	38	2	0	2	642	$5.74e-14$	42
*Re*_500	0	12	41	2	0	2	612	$7.94e-13$	42
*Re*_600	0	12	39	1	0	1	596	$4.49e-12$	41
*Re*_700	0	14	45	1	0	1	594	$1.7e-09$	43
*Re*_800	0	14	47	2	0	2	485	$1.04e-06$	37
*Re*_900	0	17	56	2	0	2	819	$2.16e-13$	57
*Re*_1000	0	18	63	4	0	4	923	$1.72e-10$	61

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	19	0	0	0	714	$3.06e-15$	40
*Re*_200	0	7	22	0	0	0	867	$7.28e-15$	49
*Re*_300	0	7	22	0	0	0	871	$1.01e-12$	49
*Re*_400	0	8	27	1	0	1	1006	$6.08e-12$	57
*Re*_500	0	9	30	1	0	1	1118	$7.85e-14$	63
*Re*_600	0	9	30	1	0	1	1123	$1.21e-12$	64
*Re*_700	0	9	32	2	0	2	1124	$2.42e-12$	64
*Re*_800	0	9	32	2	0	2	1172	$4.89e-12$	68
*Re*_900	0	10	37	3	0	3	1292	$3.78e-13$	75
*Re*_1000	0	11	40	3	0	3	1475	$1.13e-13$	87

Dogleg Choice 1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	7	8	0	0	7	710	$1.15e-14$	37
*Re*_200	0	10	11	0	0	10	838	$7.62e-15$	44
*Re*_300	0	9	10	0	0	9	531	$5.08e-07$	26
*Re*_400	0	12	13	0	0	12	801	$2.04e-13$	43
*Re*_500	0	12	13	0	0	12	766	$1.41e-12$	41
*Re*_600	0	15	16	0	0	15	755	$2.83e-09$	38
*Re*_700	0	20	21	0	2	18	1295	$1.56e-13$	67
*Re*_800	0	19	21	0	2	17	997	$1.11e-09$	52
*Re*_900	0	18	19	0	0	18	722	$1.39e-07$	38
*Re*_1000	0	22	24	0	2	20	1298	$2.52e-12$	68

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	7	0	0	6	714	$3.06e-15$	39
*Re*_200	0	7	8	0	0	7	867	$7.28e-15$	48
*Re*_300	0	7	8	0	0	7	871	$1.01e-12$	48
*Re*_400	0	8	9	0	0	8	1016	$4.89e-14$	57
*Re*_500	0	9	11	0	1	8	1126	$8.53e-13$	63
*Re*_600	0	10	12	0	2	8	1231	$6.99e-14$	68
*Re*_700	0	17	20	0	8	9	2330	$5.78e-14$	133
*Re*_800	0	11	14	0	3	8	1451	$7.28e-14$	82
*Re*_900	0	13	18	0	7	6	1756	$2.31e-13$	100
*Re*_1000	0	19	27	0	13	6	2672	$1.38e-11$	155

Thermal Convection 3D

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	0	0	0	298	$2.22e-12$	161
$Ra\_1.0e04$	0	8	10	1	0	1	469	$1.92e-15$	257
$Ra\_1.0e05$	0	19	25	3	0	5	584	$5.1e-15$	478
$Ra\_1.0e06$	0	58	97	27	0	38	3291	$6.32e-14$	1818

Cubic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	0	0	0	504	$1.23e-16$	214
$Ra\_1.0e04$	0	6	8	1	0	1	697	$1.09e-15$	274
$Ra\_1.0e05$	0	10	17	4	0	6	1120	$5.66e-15$	453
$Ra\_1.0e06$	0	20	57	14	0	36	2377	$4.43e-14$	955

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	0	0	0	298	$2.22e-12$	161
$Ra\_1.0e04$	0	8	10	1	0	1	469	$1.92e-15$	252
$Ra\_1.0e05$	0	18	24	4	0	5	640	$6.66e-15$	478
$Ra\_1.0e06$	0	58	93	28	0	34	3153	$5.57e-14$	1779

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	0	0	0	504	$1.23e-16$	212
$Ra\_1.0e04$	0	6	8	1	0	1	697	$1.09e-15$	278
$Ra\_1.0e05$	0	11	19	5	0	7	1248	$7.57e-15$	499
$Ra\_1.0e06$	0	26	64	20	0	37	3156	$4.32e-14$	1259

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	16	0	0	0	298	$2.22e-12$	177
$Ra\_1.0e04$	0	9	30	1	0	1	636	$6.56e-16$	347
$Ra\_1.0e05$	0	18	67	5	0	6	702	$1.49e-08$	576
$Ra\_1.0e06$	0	48	213	23	0	34	2710	$5.74e-14$	1774

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	16	0	0	0	504	$1.23e-16$	211
$Ra\_1.0e04$	0	7	24	1	0	1	813	$7.12e-16$	323
$Ra\_1.0e05$	0	10	43	4	0	6	1117	$5.56e-15$	466
$Ra\_1.0e06$	0	21	134	15	0	35	2501	$4.19e-14$	1079

Dogleg Choice 1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e3$	0	6	7	0	0	6	486	$1.16e-16$	119
$Ra\_1.0e4$	0	14	17	0	4	10	854	$1.09e-15$	242
$Ra\_1.0e5$	0	25	31	0	9	16	1354	$5.64e-15$	404
$Ra\_1.0e6$	0	109	142	0	88	21	7081	$4.34e-14$	1872

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	0	0	5	504	$1.23e-16$	124
$Ra\_1.0e04$	0	7	9	0	2	5	817	$1.09e-15$	213
$Ra\_1.0e05$	0	10	13	0	5	5	1194	$6.45e-15$	266
$Ra\_1.0e06$	0	45	59	0	41	4	5965	$5.83e-14$	1331

Full Step Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	NA	NA	NA	298	$2.22e-12$	157
$Ra\_1.0e04$	0	12	13	NA	NA	NA	446	$1.13e-15$	319
$Ra\_1.0e05$	0	20	21	NA	NA	NA	585	$7.82e-15$	501
$Ra\_1.0e06$	−1	200	201	NA	NA	NA	1842	$1.31e+104$	4086

Full Step Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
$Ra\_1.0e03$	0	5	6	NA	NA	NA	504	$1.23e-16$	208
$Ra\_1.0e04$	0	8	9	NA	NA	NA	871	$6.8e-16$	350
$Ra\_1.0e05$	0	12	13	NA	NA	NA	1236	$8.1e-15$	515
$Ra\_1.0e06$	−1	200	201	NA	NA	NA	4561	$1.37e+65$	4792

Lid Driven Cavity 3D

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	7	8	0	0	0	266	$2.63e-14$	104
Re_200	0	12	13	0	0	0	393	$7.27e-15$	169
Re_300	0	10	11	0	0	0	455	$1.23e-14$	160
Re_400	0	13	14	0	0	0	495	$1.23e-14$	195
Re_500	0	13	14	0	0	0	515	$1.53e-14$	206
Re_600	0	16	17	0	0	0	413	$1.33e-11$	206
Re_700	0	17	20	2	0	2	819	$8.14e-13$	283
Re_800	0	17	19	1	0	1	720	$3.97e-14$	272
Re_900	0	22	27	4	0	4	1253	$3.68e-14$	401
Re_1000	0	24	28	3	0	3	756	$5.06e-14$	340

Cubic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	0	495	$2.59e-15$	131
Re_200	0	7	8	0	0	0	625	$7.12e-15$	164
Re_300	0	8	9	0	0	0	751	$1.1e-14$	194
Re_400	0	9	10	0	0	0	864	$1.17e-14$	224
Re_500	0	9	10	0	0	0	927	$2.87e-14$	235
Re_600	0	11	12	0	0	0	1151	$3.36e-14$	290
Re_700	0	10	12	1	0	1	1176	$4.03e-14$	292
Re_800	0	11	13	1	0	1	1365	$5.53e-14$	340
Re_900	0	13	16	2	0	2	1648	$5.67e-14$	404
Re_1000	0	17	26	8	0	8	2160	$5.07e-14$	529

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	7	8	0	0	0	266	$2.63e-14$	105
Re_200	0	12	13	0	0	0	393	$7.27e-15$	168
Re_300	0	10	11	0	0	0	455	$1.23e-14$	162
Re_400	0	13	14	0	0	0	495	$1.23e-14$	196
Re_500	0	13	14	0	0	0	515	$1.53e-14$	203
Re_600	0	16	17	0	0	0	413	$1.33e-11$	207
Re_700	0	17	20	2	0	2	819	$8.14e-13$	280
Re_800	0	17	19	1	0	1	720	$3.97e-14$	274
Re_900	0	22	27	4	0	4	1253	$3.68e-14$	401
Re_1000	0	24	28	3	0	3	756	$5.06e-14$	354

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	0	495	$2.59e-15$	132
Re_200	0	7	8	0	0	0	625	$7.12e-15$	168
Re_300	0	8	9	0	0	0	751	$1.1e-14$	196
Re_400	0	9	10	0	0	0	864	$1.17e-14$	224
Re_500	0	9	10	0	0	0	927	$2.87e-14$	235
Re_600	0	11	12	0	0	0	1151	$3.36e-14$	290
Re_700	0	10	12	1	0	1	1176	$4.03e-14$	288
Re_800	0	11	13	1	0	1	1365	$5.53e-14$	338
Re_900	0	13	16	2	0	2	1648	$5.67e-14$	409
Re_1000	0	17	26	8	0	8	2160	$5.07e-14$	528

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	7	22	0	0	0	266	$2.63e-14$	124
*Re*_200	0	12	37	0	0	0	393	$7.28e-15$	204
*Re*_300	0	10	31	0	0	0	455	$1.23e-14$	191
*Re*_400	0	13	40	0	0	0	495	$1.29e-14$	233
*Re*_500	0	13	40	0	0	0	517	$1.54e-14$	237
*Re*_600	0	18	59	2	0	2	850	$1.83e-14$	349
*Re*_700	0	18	61	3	0	3	946	$5.17e-13$	363
*Re*_800	0	19	64	3	0	3	1176	$4.81e-14$	417
*Re*_900	0	19	64	3	0	3	1046	$6.42e-14$	400
*Re*_1000	0	16	53	2	0	2	813	$6.41e-14$	332

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	19	0	0	0	495	$2.58e-15$	135
*Re*_200	0	7	22	0	0	0	625	$7.15e-15$	165
*Re*_300	0	8	25	0	0	0	751	$1.1e-14$	194
*Re*_400	0	9	28	0	0	0	864	$1.15e-14$	224
*Re*_500	0	9	28	0	0	0	927	$2.86e-14$	236
*Re*_600	0	11	36	1	0	1	1211	$2.43e-14$	304
*Re*_700	0	10	33	1	0	1	1170	$4.55e-14$	290
*Re*_800	0	11	36	1	0	1	1322	$4.37e-14$	325
*Re*_900	0	12	41	2	0	2	1484	$3.69e-14$	365
*Re*_1000	0	13	46	3	0	3	1815	$4.65e-14$	441

Dogleg Choice 1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	10	11	0	1	9	537	$1.76e-15$	105
*Re*_200	0	10	11	0	1	9	604	$5.58e-15$	112
*Re*_300	0	14	15	0	1	13	651	$1.1e-14$	138
*Re*_400	0	16	18	0	3	13	743	$9.01e-13$	157
*Re*_500	0	17	19	0	2	15	924	$2.04e-14$	180
*Re*_600	0	21	24	0	4	17	1019	$1.86e-14$	210
*Re*_700	0	20	22	0	2	18	971	$3.28e-14$	203
*Re*_800	0	20	21	0	4	16	1061	$4.13e-14$	211
*Re*_900	0	25	27	0	4	21	1250	$4.59e-14$	257
*Re*_1000	0	24	25	0	3	21	1029	$4.05e-14$	234

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_100	0	7	8	0	1	6	587	$2.2e-15$	98
Re_200	0	8	9	0	1	7	720	$4.73e-15$	117
Re_300	0	8	9	0	1	7	754	$1.27e-14$	119
Re_400	0	9	10	0	2	7	877	$1.4e-14$	137
Re_500	0	9	10	0	1	8	929	$2.34e-14$	141
Re_600	0	10	11	0	3	7	1099	$2.37e-14$	169
Re_700	0	10	11	0	3	7	1192	$4.52e-14$	200
Re_800	0	12	13	0	3	9	1462	$5.55e-14$	206
Re_900	0	12	13	0	2	10	1506	$3.64e-14$	247
Re_1000	0	23	27	0	15	8	4669	$5.27e-14$	901

Full Step Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	7	8	NA	NA	NA	266	$2.63e-14$	103
Re_200	0	12	13	NA	NA	NA	393	$7.27e-15$	173
Re_300	0	10	11	NA	NA	NA	455	$1.23e-14$	162
Re_400	0	13	14	NA	NA	NA	495	$1.23e-14$	196
Re_500	0	13	14	NA	NA	NA	515	$1.53e-14$	204
Re_600	0	16	17	NA	NA	NA	413	$1.33e-11$	206
Re_700	0	18	19	NA	NA	NA	736	$3.92e-14$	278
Re_800	0	27	28	NA	NA	NA	629	$3.52e-14$	339
Re_900	0	32	33	NA	NA	NA	775	$3.66e-14$	410
Re_1000	0	35	36	NA	NA	NA	819	$5.69e-14$	446

Full Step Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	NA	NA	NA	495	$2.59e-15$	131
Re_200	0	7	8	NA	NA	NA	625	$7.12e-15$	162
Re_300	0	8	9	NA	NA	NA	751	$1.1e-14$	192
Re_400	0	9	10	NA	NA	NA	864	$1.17e-14$	227
Re_500	0	9	10	NA	NA	NA	927	$2.87e-14$	236
Re_600	0	11	12	NA	NA	NA	1151	$3.36e-14$	297
Re_700	−1	200	201	NA	NA	NA	924	116	1908
Re_800	−1	200	201	NA	NA	NA	1048	154	1924
Re_900	−1	200	201	NA	NA	NA	792	$3.48e+09$	1884
Re_1000	−1	200	201	NA	NA	NA	1021	771	1934

Backward Facing Step 3D

Cubic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	12	13	0	0	0	2929	$1.99e-11$	415
Re_200	0	12	14	1	0	1	3467	$6.56e-10$	468
Re_300	0	14	16	1	0	1	4022	$6.4e-11$	539
Re_400	0	15	17	1	0	1	4269	$4.44e-12$	587
Re_500	0	15	17	1	0	1	4442	$3.66e-11$	597
Re_600	0	17	19	1	0	1	5286	$4.2e-12$	699
Re_700	0	17	20	2	0	2	4267	$1.11e-09$	586
Re_750	0	18	21	2	0	2	4401	$2.76e-10$	614
Re_800	0	18	21	2	0	2	4049	$5.92e-10$	580

Cubic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	0	3511	$7e-12$	527
Re_200	0	7	8	0	0	0	4096	$8.52e-12$	506
Re_300	0	8	9	0	0	0	4800	$3.87e-12$	616
Re_400	0	10	11	0	0	0	6000	$6.06e-12$	832
Re_500	0	14	16	1	0	1	8399	$1.64e-11$	1111
Re_600	0	34	87	24	0	52	20400	$1.89e-10$	2689
Re_700	0	10	13	2	0	2	6000	$2.18e-10$	827
Re_750	0	10	13	2	0	2	6000	$6.85e-11$	809
Re_800	0	11	14	2	0	2	6600	$2.83e-10$	863

Quadratic Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	12	13	0	0	0	2929	$1.99e-11$	409
Re_200	0	12	14	1	0	1	3467	$6.56e-10$	464
Re_300	0	14	16	1	0	1	4022	$6.4e-11$	554
Re_400	0	15	17	1	0	1	4269	$4.44e-12$	572
Re_500	0	15	17	1	0	1	4442	$3.66e-11$	595
Re_600	0	17	19	1	0	1	5286	$4.2e-12$	689
Re_700	0	17	20	2	0	2	4267	$1.11e-09$	588
Re_750	0	18	21	2	0	2	4401	$2.76e-10$	624
Re_800	0	18	21	2	0	2	4049	$5.92e-10$	577

Quadratic Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	7	0	0	0	3511	$7e-12$	413
*Re*_200	0	7	8	0	0	0	4096	$8.52e-12$	481
*Re*_300	0	8	9	0	0	0	4800	$3.87e-12$	560
*Re*_400	0	10	11	0	0	0	6000	$6.06e-12$	703
*Re*_500	0	14	16	1	0	1	8399	$1.64e-11$	989
*Re*_600	0	32	63	20	0	30	19200	$1.53e-11$	2279
*Re*_700	0	10	13	2	0	2	6000	$2.18e-10$	706
*Re*_750	0	10	13	2	0	2	6000	$6.85e-11$	701
*Re*_800	0	11	14	2	0	2	6600	$2.83e-10$	773

Moré–Thuente Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	12	13	0	0	0	2929	$1.99e-11$	489
*Re*_200	0	15	17	1	0	1	2678	$1.97e-09$	527
*Re*_300	0	11	13	1	0	1	3501	$2.37e-11$	537
*Re*_400	0	19	23	3	0	3	3200	$3.56e-10$	615
*Re*_500	0	14	17	2	0	2	3937	$2.87e-10$	614
*Re*_600	0	16	20	3	0	3	3901	$2.33e-10$	598
*Re*_700	0	23	32	8	0	8	3672	$1.46e-09$	745
*Re*_750	0	26	37	10	0	10	4072	$9.92e-11$	815
*Re*_800	0	26	38	9	1	10	4095	$9.95e-10$	854

Moré–Thuente Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
*Re*_100	0	6	7	0	0	0	3511	$7e-12$	508
*Re*_200	0	7	8	0	0	0	4096	$8.52e-12$	603
*Re*_300	0	8	9	0	0	0	4800	$3.87e-12$	712
*Re*_400	0	10	11	0	0	0	6000	$6.06e-12$	881
*Re*_500	0	11	13	1	0	1	6600	$3.23e-11$	975
*Re*_600	0	14	19	4	0	4	8400	$3.44e-13$	1235
*Re*_700	0	15	22	6	0	6	9000	$1.29e-10$	1330
*Re*_750	−1	200	256	41	6	49	9687	9.46	3595
*Re*_800	−1	200	327	117	3	123	75645	2.59	12083

Dogleg Choice1

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_100	0	9	10	0	0	9	2592	$2.7e-10$	376
Re_200	0	10	11	0	1	9	3183	$2.56e-11$	426
Re_300	0	12	13	0	5	7	4270	$6.98e-11$	562
Re_400	0	16	17	0	9	7	6391	$3.34e-11$	832
Re_500	0	22	23	0	15	7	11317	$6.65e-12$	1396
Re_600	0	29	30	0	21	8	15565	$6.39e-11$	1921
Re_700	0	37	38	0	29	8	20954	$3.37e-11$	2553
Re_750	0	41	42	0	33	8	22985	$1.42e-10$	2806
Re_800	0	44	45	0	38	6	25391	$8.97e-10$	3113

Dogleg Constant

	S/F	INS	FE	0:C	C:IN	IN	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	0	0	6	3511	$7.0e-12$	464
Re_200	0	7	8	0	2	5	4194	$2.94e-11$	549
Re_300	0	10	11	0	5	5	6000	$1.38e-11$	783
Re_400	0	15	16	0	9	6	9000	$3.54e-11$	1172
Re_500	0	20	21	0	15	5	12000	$3.62e-11$	1575
Re_600	0	27	28	0	21	6	16200	$1.23e-11$	2209
Re_700	0	35	36	0	29	6	21000	$1.69e-10$	2753
Re_750	0	39	40	0	33	6	23400	$9.99e-11$	3091
Re_800	0	45	46	0	38	7	27000	$5.62e-11$	3489

Full Step Choice 1

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	12	13	NA	NA	NA	2929	$1.99e-11$	414
Re_200	0	14	15	NA	NA	NA	2677	$1.1e-09$	405
Re_300	0	16	17	NA	NA	NA	3026	$4.05e-10$	465
Re_400	0	15	16	NA	NA	NA	3195	$1.76e-09$	467
Re_500	0	18	19	NA	NA	NA	4108	$2.48e-11$	598
Re_600	0	22	23	NA	NA	NA	3717	$3.16e-10$	587
Re_700	−1	200	201	NA	NA	NA	4579	3.29	2535
Re_750	−1	200	201	NA	NA	NA	3068	2.36	2738
Re_800	−1	200	201	NA	NA	NA	6203	3.04	2567

Full Step Constant

	S/F	INS	FE	LS	f−LS	Bkt	GMRES	$\|\|F\|\|$	Time
Re_100	0	6	7	NA	NA	NA	3511	$7e-12$	429
Re_200	0	7	8	NA	NA	NA	4096	$8.52e-12$	492
Re_300	0	8	9	NA	NA	NA	4800	$3.87e-12$	574
Re_400	0	10	11	NA	NA	NA	6000	$6.06e-12$	713
Re_500	−1	200	201	NA	NA	NA	17149	229	3627
Re_600	−1	200	201	NA	NA	NA	4899	36.2	2492
Re_700	−1	200	201	NA	NA	NA	7003	55.6	2660
Re_750	−1	200	201	NA	NA	NA	14166	23.8	3280
Re_800	−1	200	201	NA	NA	NA	13953	18.8	3361

DISTRIBUTION:

1 MS 0321
Bill Camp, 09200

1 MS 0318
Paul Yarrington, 09230

1 MS 0316
Sudip Dosanjh, 09233

1 MS 1110
David Womble, 09214

10 MS 0316
Roger Pawlowski, 09233

10 MS 1111
John N. Shadid, 09233

1 MS 9217
Tamara G. Kolda, 08950

1 MS 0316
Russell Hooper, 09233

1 MS 1111
Andrew Salinger, 09233

1 MS 1111
Eric Phipps, 09233

1 MS 1111
Brett Bader, 09233

1 MS 1110
Mike Heroux, 09214

1 MS 1110
James Willenbring, 09214

1 MS 0316
Scott A. Hutchinson, 09233

1 MS 0316
Eric R. Keiter, 09233

1 MS 0316
Robert J. Hoekstra, 09233

1 MS 0316
Joseph P. Castro, 09233

1 MS 1110
David Day, 09214

1 MS 0316
Curtis Ober, 09233

1 MS 0316
Thomas Smith, 09233

1 MS 0835
Alfred Lorber, 09141

1 MS 0316
William Spotz, 09233

1 MS 9217
Paul Boggs, 08950

1 MS 1110
Roscoe Bartlett, 09211

1 MS 0847
Bart van Bloemen Waanders, 09211

1 MS 0834
Patrick Notz, 09411

1 MS 0834
Matt Hopkins, 09411

1 MS 0834
P. Randall Schunk, 09411

1 MS 0835
Kendall Pierson, 09142

1 MS 0847
Garth Reese, 09142

1 MS 0826
James Stewart, 09143

1 MS 0826
Alan Williams, 09143

1 MS 1186
Thomas Brunner, 01674

1 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616

5 Homer F. Walker
WPI Mathematical Sciences
100 Institute Road
Worcester, MA 01609

5 Joseph P. Simonis
WPI Mathematical Sciences
100 Institute Road
Worcester, MA 01609

5 Carol S. Woodward
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Box 808, L-561
Livermore, CA 94551

5 David E. Keyes
Appl Phys & Appl Math
Columbia University
200 S. W. Mudd Bldg., MC 4701
500 W. 120th Street
New York, NY, 10027

1 Peter N. Brown
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Box 808, L-561
Livermore, CA 94551

1 Richard Byrd
University of Colorado at Boulder
Department of Computer Science
430 UCB
Boulder, CO 80309-0430

1 Xiao–Chuan Cai
University of Colorado at Boulder
Department of Computer Science
430 UCB
Boulder, CO 80309-0430

1 John E. Dennis, Jr.
8419 42nd Ave SW
Seattle, WA 98136

1 Stanley C. Eisenstat
Department of Computer Science
Yale University
P.O. Box 208285
New Haven, CT 06520-8285

1 Paul F. Fischer
Building 221, Room D-248
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439

1 Jorge J. Moré
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439-4844

1 William D. Gropp
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S Cass Ave
Argonne, IL 60439

1 C. Tim Kelley
Department of Mathematics , Box 8205
Center for Research in Scientific Computation
North Carolina State University
Raleigh, NC 27695-8205

1 Dana Knoll
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545

1 Lois Curfman McInnes
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4844

1 Jorge Nocedal
Northwestern University
Electrical and Computer Engineering Department
2145 Sheridan Road
Evanston, IL 60208-3118

1 Michael Pernice
Computer and Computational Sciences Division
Los Alamos National Laboratory
P.O. Box 1663, MS B256
Los Alamos, NM 87545

1 Linda Petzold
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5070

1 Yousef Saad
Department of Computer Science and Engineering
University of Minnesota
4-192 EE/CSci Building
200 Union Street SE
Minneapolis, MN 55455

1 Barry Smith
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S Cass Ave
Argonne, IL 60439

1 Steve Wright
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706

1 David P. Young
The Boeing Company
PO Box 24346, M/S 7L-21
Seattle, WA 98124-2207

1 Mary F. Wheeler
Texas Institute for Computational and Applied Mathematics
The University of Texas at Austin
SHC 318 / UT Mail Code C0200
Austin, Texas 78712