2003-04-30

# A Numerical Study of Globalizations of Newton-GMRES Methods

Joseph P. Simonis
*Worcester Polytechnic Institute*

**A Numerical Study of Globalizations of Newton-GMRES Methods**

by

Joseph P. Simonis

A Thesis

Submitted to the Faculty

of

WORCESTER POLYTECHNIC INSTITUTE

in Partial Fulfillment of the Requirements for the

Degree of Master of Science

in

Applied Mathematics

by

---

May 2003

APPROVED:

---

Dr. Homer Walker, Thesis Advisor

---

Dr. Bogdan Vernescu, Department Head

# Abstract

Newton's method is at the core of many algorithms used for solving nonlinear equations. A globalized Newton method is an implementation of Newton's method augmented with "globalization procedures" intended to enhance the likelihood of convergence to a solution from an arbitrary initial guess. A Newton-GMRES method is an implementation of Newton's method in which the iterative linear algebra method GMRES is used to solve approximately the linear system that characterizes the Newton step. A globalized Newton-GMRES method combines both globalization procedures and the GMRES scheme to develop robust and efficient algorithms for solving nonlinear equations. The aim of this project is to describe the development of some globalized Newton-GMRES methods and to compare their performances on a few benchmark fluid flow problems.

# Acknowledgments

I was first introduced to the topic of Numerical Methods for Nonlinear Equations by my advisor Prof. Walker during the summer of 2001 when he offered me a summer research position. I am grateful for his help, he was always there to answer my questions and to discuss my work. I am also very grateful for the internship he found for me at Sandia National Laboratories. It presented me with a wonderful topic for this thesis and a chance to work with wonderful people..

I also wish to thank the researchers at Sandia National Laboratories: John Shadid, Roger Pawlowski, Tamara Kolda and Paul Lin. Over the past year that I have known them they have tutored me in the subjects of fluid dynamics, mechanical engineering, computer science and mathematics. John, Roger, and Tamara all worked very hard to see that I had working codes that formulated and solved the test problems. Paul Lin was always there to ensure that I had almost painless methods of running numerous instances of the codes on the parallel machines.

Finally, I would like to thank Sandia National Laboratories for the funding throughout the school year to work on this project.

# Contents

# Chapter 1

# Introduction

There is a rapidly growing demand in almost all fields of applied sciences for ways of accurately and efficiently solving very large nonlinear systems of equations. As processing speeds and memory capabilities of computers increase, the desire to solve larger and larger systems arises. It is very important to have solvers which are both robust and efficient.

The following study looks at a class of solvers known as globalized Newton-GMRES methods. These are methods stemming from the classical Newton method and are specifically designed for use on large-scale problems. A globalized Newton method is an implementation of the Newton method augmented with "globalization procedures" intended to enhance the likelihood of convergence to a solution from an arbitrary initial guess. A Newton-GMRES method is an implementation of Newton's method in which the iterative linear algebra method GMRES [14] is used to solve approximately the linear system that characterizes the Newton step. A Newton-GMRES method is just one of many Newton iterative methods. These are methods that use an iterative linear solver within the Newton algorithm. Newton iterative methods are in turn special cases of inexact Newton methods [3], in which an approximation of

the Newton step is used at each iteration.

The following chapters are aimed at describing the development of robust and efficient globalized Newton-GMRES methods. The definitions of inexact Newton methods, Newton iterative methods, and globalization methods will be given; then a few feasible globalized algorithms will be developed and compared.

Three challenging fluid dynamics problems were chosen to assess the effectiveness of the solvers. The first problem is a thermal convection problem consisting of fluid confined to a differentially heated rectangular domain. The second is a backward-facing step problem. In this, fluid comes in at one end of a duct and flows rapidly over a "stair", causing recirculation in the low pressure area. The final problem is the lid-driven cavity problem. In this, fluid flows across the top of a cavity filled with the same fluid. This flow causes a large circulation within the domain, and when the velocity is high enough, smaller circulations also appear within the corners of the cavity. In our experiments with these problems, the goal is to determine the relative robustness and efficiency of the algorithms. We determine robustness by the number and difficulty of the problems a globalized method successfully solves. Efficiency is based on mean values of solver expenses, e.g., time or linear solver iterations.

The algorithms and problems were developed by teams of researchers at Sandia National Laboratories in Albuquerque, New Mexico The principal codes used were NOX (see [9]) and MPSalsa [15]. NOX is a software package designed to solve large-scale nonlinear problems. MPSalsa is a finite element computer program for reacting flow problems. The codes were run on large parallel machines there.

# Chapter 2

# Theoretical Development

This chapter deals with the theoretical development of globalized Newton iterative methods, specifically globalized Newton-GMRES methods. These methods use the GMRES method to solve the linear subproblem of Newton's method and use globalization methods to improve the likelihood of convergence from a poor starting guess.

The overall goal of a globalized Newton-GMRES method is obtaining a solution to the following problem:

$$\text{Given } F : \mathbf{R}^n \to \mathbf{R}^n,$$
$$\text{find } x_* \text{ such that } F(x_*) = 0.$$

The following two assumptions will be used in many of the theorems presented below.

**Assumption 2.0.1** *$F$ is continuously differentiable.*

**Assumption 2.0.2** *$F(x_*) = 0$ and $F'(x_*)$ is nonsingular.*

It is almost always impossible to solve this problem exactly. Thus iterative methods must be formulated to find an approximate numerical solution. For the algorithms presented here, the goal is to find an iterate $x_k$ approximating $x_*$ "well enough".

## 2.1 Newton's Method

The general iterative approach to solving the problem is to begin with a trial value $x_0$ and then to construct hopefully better and better approximations to the solution. A classical way of constructing better approximations is via Newton's method or the Newton-Raphson Method, which is most easily described in one dimension and then extended to $n$-dimensional space.

Assume a function $f : \mathbf{R} \to \mathbf{R}$ and an initial guess $x_0$ are given. Graphically the problem can be formulated as: Find the intersection of the graph of $f$ and the x-axis. The basic idea of Newton's method is to find a better guess by replacing the curve by a suitable line whose intersection with the x-axis can be easily computed. For classical Newton's method, the line is the tangent line at $x_0$. Compute the intersection of the tangent line and the x-axis, and use this point as the next approximate solution. The tangent line is given by $y = f'(x_0)(x - x_0) + f(x_0)$, henceforth called the local linear model of the function. The point of intersection $x_1$ satisfies $0 = f'(x_0)(x_1 - x_0) + f(x_0)$, so the next approximate solution is $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$. Now iterate using $x_1$ as the new initial guess.

In $n$ dimensions, $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, $F(x) = \begin{pmatrix} F_1(x) \\ \vdots \\ F_n(x) \end{pmatrix}$ and $F'(x) = J(x) = \left(\frac{\partial F_i(x)}{\partial x_j}\right) \in \mathbf{R}^{n \times n}$. The local linear model of $F$ is $F(x) + J(x)s$. In this model, $s$ is the unknown that yields the next step via $x_1 = x_0 + s$, with $s$ found by solving $F(x) + J(x)s = 0$. A formal algorithm can now be written.

**Newton's Method**[6]:
Given an initial $x$.

Iterate:
    Decide whether to stop or continue.
    Solve $J(x)s = -F(x)$.
    Update $x \leftarrow x + s$.

As mentioned before, Newton's method is an iterative method. It produces a sequence of approximate solutions $x_1, x_2, \ldots$ from a starting guess $x_0$. Two questions that can be asked about the sequence produced are: 1) does it converge to a solution $x_*$ and if it does 2) how quickly does it converge to that solution? At this point some definitions are in order. We assume that $\|\cdot\|$ is a norm of interest on $\mathbf{R}^n$.

**Def 2.1.1 ([6])** *Let $x_* \in \mathbf{R}^n$, $x_k \in \mathbf{R}^n, k = 1, 2, \ldots$. Then the sequence $\{x_k\} = \{x_1, x_2, x_3, \ldots\}$ is said to* ***converge*** *to $x_*$ if $\lim_{k\to\infty} \|x_k - x_*\| = 0$. If in addition, there exists a constant $c \in [0, 1)$ and an integer $\hat{k} \geq 0$ such that for all $k \geq \hat{k}$, $\|x_{k+1} - x_*\| \leq c\|x_k - x_*\|$ then $\{x_k\}$ is said to be* ***q-linearly convergent*** *to $x_*$. If for some sequence $\{c_k\}$ that converges to 0, $\|x_{k+1} - x_*\| \leq c_k\|x_k - x_*\|$ for each $k$, then $\{x_k\}$ is said to* ***converge q-superlinearly*** *to $x_*$. If $\{x_k\}$ converges to $x_*$ and there exist constants $p > 1, c \geq 0$, such that $\{x_k\}$ converges to $x_*$ and for all $k \geq \hat{k}$, $\|x_{k+1} - x_*\| \leq c\|x_k - x_*\|^p$, then $\{x_k\}$ is said to converge to $x_*$ with* ***q-order at least p****. If $p = 2$ or $p = 3$, the convergence is said to be* ***q-quadratic*** *or* ***q-cubic****, respectively.*

In the above definitions the prefix 'q' stands for quotient, distinguishing these from 'r' (root) order convergence. The quotient orders are the stronger of the two; thus r-orders will not be discussed. For more information see Ortega and Rheinboldt[12]. The 'q' will be dropped from this point onward.

**Def 2.1.2 ([6])** *A function $g$ is* ***Lipschitz continuous*** *with constant $\gamma$ in a set*

*$X \in \mathbf{R}^n$, written $g \in Lip_\gamma(X)$, if for every $x$, $y \in X$, $\|g(x) - g(y)\| \leq \gamma\|x - y\|$.*

The following theorem answers the two questions posed of the sequence produced by Newton's method:

**Theorem 2.1.3 ([17])** *Suppose $F$ is Lipschitz continuously differentiable at $x_*$, and Assumptions 2.0.1 and 2.0.2 hold. Then for $x_0$ sufficiently near $x_*$, $\{x_k\}$ produced by Newton's method is well-defined and converges to $x_*$ with*

$$\|x_{k+1} - x_*\| \leq c\|x_k - x_*\|^2$$

*for a constant $c$ independent of $k$.*

Under the assumptions of the theorem, the sequence does converge locally to $x_*$ and it ultimately does so quadratically. Unfortunately this convergence only holds for $x_0$ within a neighborhood of the solution. For initial guesses far from the solution, a new strategy must be devised. Globalizations of Newton's method are methods in which some strategy for getting 'near' the solution from an arbitrary initial guess is implemented, and then once 'close' enough Newton's method is used to rapidly converge to the solution. These methods are discussed in Section 2.3.

## 2.2 Inexact Newton Methods

In Newton's method the goal is to update $x_k$ to $x_{k+1}$ by taking the step $s_k$ that solves $J(x_k)s_k = -F(x_k)$. Solving this system may be computationally expensive, on the order of $O(n^3)$ arithmetic operations for direct linear algebra methods. In large-scale problems, where $n$ is $10^4$, $10^5$ or greater, direct solutions are often infeasible. An alternative is to use iterative linear algebra methods to form an approximation of the

solution. Using an iterative linear solver to obtain approximate Newton steps results in a *Newton iterative method*, or a *truncated Newton method*.

For Newton iterative methods, one of the questions naturally asked is: How accurately must the iterative method solve the linear systems in order that the Newton iterates converge? The key idea is to get convergence, but also to minimize the amount of computation necessary to obtain it. It seems desirable that the next approximate solution $x_{k+1}$ make $\|F(x_{k+1})\|$ smaller than $\|F(x_k)\|$. If the local linear model, $F(x_k) + J(x_k)s_k$, is accurate, then a decrease in the model norm should correspond to a decrease in the function norm. However, a lot of effort may go into finding a very good approximation to the Newton step; when a poorer one would give at least as much reduction in $\|F\|$, then much of this effort may have been wasted. (In such a case, the algorithm is said to be 'oversolving' the problem.) Thus the iterative method should stop when a step which reduces the norm of the local linear model of $F$ by an appropriate amount is found. This restriction then leads to an *inexact Newton method*. An inexact Newton method [3] is any method each step of which reduces the norm of the local linear model of $F$.

**Inexact Newton Method**[3]:
Given an initial $x$.
Iterate:
Decide whether to stop or continue.
Find **some** $\eta \in [0, 1)$ and $s$ that satisfy
$\|F(x) + J(x)s\| \leq \eta \|F(x)\|$.
Update $x \leftarrow x + s$.

At each step of a Newton iterative implementation, one chooses $\eta \in [0, 1)$ and then calls an iterative linear algebra method to solve for a step meeting the given tolerance. In this context $\eta$ is called the *forcing term* because it forces a reduction of the local linear model norm. The following section outlines the details of choosing the $\eta$'s in such a way as to maintain the convergence properties of Newton's method.

### 2.2.1 Forcing Terms

The choice of the forcing terms in an inexact Newton method plays a critical role in the determination of whether or not the iterates converge to a solution and the rate of that convergence. The choice can have a very dramatic effect, demonstrated by the following three theorems found in [3]. Different restrictions on the sequence $\{\eta_k\}$ give great differences in the convergence of the inexact Newton iterates.

**Theorem 2.2.1** *Suppose Assumptions 2.0.1 and 2.0.2 hold and that* $\eta_k \leq \eta_{max} < t < 1$ *for all* $k$. *There exists* $\epsilon > 0$ *such that if* $\|x_0 - x_*\| \leq \epsilon$, *then the sequence of inexact Newton iterates* $\{x_k\}$ *converges to* $x_*$. *Moreover the convergence is linear in the sense that* $\|x_{k+1} - x_*\|_* \leq t\|x_k - x_*\|_*$, *where* $\|y\|_* = \|F'(x_*)y\|$ *for all* $y \in \mathbf{R}^n$.

**Theorem 2.2.2** *Suppose Assumptions 2.0.1 and 2.0.2 hold and that the inexact Newton iterates* $\{x_k\}$ *converge to* $x_*$. *Then* $x_k \to x_*$ *superlinearly iff* $\|F(x_k) + F'(x_k)s_k\| = o(\|F(x_k)\|)$ *as* $k \to \infty$.

For the final theorem, it is helpful to be reminded of a definition:

**Def 2.2.3** *A function* $g$ *is said to be Hölder continuous with exponent* $p \in (0, 1]$ *in a set* $X \in \mathbf{R}^n$ *if for some constant* $C$ *we have* $\|g(x) - g(y)\| \leq C\|x - y\|^p$ *for every* $x, y \in X$.

**Theorem 2.2.4** *Suppose Assumptions 2.0.1 and 2.0.2 hold and assume that the inexact Newton iterates* $\{x_k\}$ *converge to* $x_*$. *Then* $x_k \to x_*$ *with order at least* $1 + p$ *if* $F'$ *is Hölder continuous with exponent* $p$ *at* $x_*$ *and* $\eta_k = O(\|F(x_k)\|^p)$ *as* $k \to \infty$.

These three theorems show the importance of the choice of $\eta_k$ in the convergence rates near the solution. If $\eta_k$ is only chosen such that $\eta_k \in [0, \eta_{max}]$ for some $\eta_{max} < 1$,

then the convergence is linear. Requiring $\lim_{k\to\infty} \eta_k = 0$ increases the convergence rate to superlinear, and finally imposing $\eta_k = O(\|F(x_k)\|)$ gains quadratic convergence (if p=1 in the final theorem).

Many implementations of inexact Newton methods set a constant forcing term, usually something small such as $10^{-2}$ or $10^{-4}$, forcing the linear solver to choose a step very close to the Newton step. Other implementations have used decreasing forcing terms in order to obtain increasingly rapid convergence near the solution. Examples include $\eta_k = \frac{1}{2^{k+1}}$ [2], which gives local superlinear convergence, and $\eta_k = min\{\|F(x_k)\|, \frac{1}{k+2}\}$ [4], which gives the convergence rate of Newton's method (typically quadratic). Upon first examination it might seem that very small or decreasing $\eta_k$'s should be employed, but it must be remembered if the current approximation of the solution is not near $x_*$, then 'the local linear model may disagree with $F$ considerably at a step that closely approximates the Newton step.'[17] In this case the linear solver may work very hard to find a step that gives little or no decrease in $\|F\|$. So away from $x_*$ a desirable $\eta_k$ will in some way be based on the agreement between $F$ and its local linear model.

Eisenstat and Walker suggest an adaptive forcing term as follows [8]:

$$\eta_0 \in [0,1), \eta_k = \frac{\left|\|F(x_k)\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|\right|}{\|F(x_{k-1})\|}, k = 1, 2, \ldots, \tag{2.1}$$

With this choice a large discrepancy between the linear model at the previous step and $F$ at the current point will yield a larger $\eta_k$. Thus on the next iteration the linear solver will not spend excessive processor time finding a close approximation to the Newton step. Conversely if the previous step's model value and the current function value are in close agreement, $\eta_k$ will be small and the next step will approximate the Newton step very closely. Thus the choice (2.1) seems likely to reduce oversolving and has been shown to do so in experiments in [8] and [16].

Reasonable questions to ask are: With this choice of $\eta_k$ does convergence occur once an $x_k$ is in a small neighborhood of $x_*$, and what is the order of the convergence if it occurs?

**Theorem 2.2.5 ([8])** *Suppose Assumptions 2.0.1 and 2.0.2 hold and the Jacobian is Lipschitz continuous at $x_*$. If $x_0$ is sufficiently near $x_*$ then $\{x_k\}$ produced by the inexact Newton method with $\{\eta_k\}$ given by (2.1) remains near $x_*$ and converges to $x_*$ with*

$$\|x_{k+1} - x_*\| \leq \beta \|x_k - x_*\| \|x_{k-1} - x_*\|, \quad k = 1, 2, \ldots,$$

*for a constant $\beta$ independent of $k$.*

Thus once an $x_k$ close to the solution is found, superlinear convergence is expected.

Although the forcing term (2.1) works well for avoiding oversolving it does have its drawbacks. For example, assume that far away from $x_*$ there is an $x_k$ where $F(x_k)$ and the local linear model just happen to agree really well. Then $\eta_k$ will be very small and the linear solver may oversolve. To help avoid this, safeguards are implemented to ensure that the sequence $\{\eta_k\}$ does not decrease too rapidly. 'The rationale is that if large forcing terms are appropriate at some point, then subsequent forcing terms should not be allowed to become much smaller until this has been justified over several iterations.'[8] Also, far away from the solution it is possible for $\eta_k$ to become greater than 1. Thus a safeguard to ensure $\eta_k \in [0, 1)$ is necessary. A safeguard proposed by Eisenstat and Walker in [8] and implemented in the numerical studies discussed below is to modify $\eta_k \leftarrow max\{\eta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\}$ whenever $\eta_{k-1}^{(1+\sqrt{5})/2} > .1$. We also require that $\eta_k \in [0.0001, .9]$.

## 2.3 Global Inexact Newton Methods

Previously it was shown that under certain conditions the iterates produced by Newton's method converge *quadratically* to a solution $x_*$, and that those of the inexact Newton method with $\eta_k$ given by (2.1) can obtain *superlinear* convergence. Both cases assume an initial guess $x_0$ sufficiently close to $x_*$. Nothing has been said about how to get 'close' to the solution when the initial guess is far from the solution. Here the inexact Newton method will be augmented with additional conditions on the choices of iterates $\{x_k\}$ to enhance the likelihood of convergence to $x_*$. The new method is known as a *global inexact Newton method*. This section presents the general global inexact Newton method along with two specific step acceptance criteria. Section 3.2 outlines several *globalization procedures* that modify the steps as necessary to meet these criteria. It is important to note that no strategy will determine a sequence that converges to a solution for every problem; rather the globalization techniques are used only to **enhance** the likelihood of convergence to some solution of the problem.

As before, it seems desirable to find a sequence of iterates such that $\|F(x_{k+1})\| < \|F(x_k)\|$. However, this condition alone is not sufficient to ensure convergence. Therefore the global inexact Newton method includes a sufficient decrease condition on the iterate norms $\|F(x_k)\|$.

**Global Inexact Newton Method**[7]:
Given an initial $x$ and $t \in (0, 1)$.
Iterate:
Decide whether to stop or continue.
Find **some** $\eta \in [0, 1)$ and $s$ that satisfy
$\|F(x) + J(x)s\| \leq \eta\|F(x)\|$
and
$\|F(x + s)\| \leq [1 - t(1 - \eta)]\|F(x)\|$
Update $x \leftarrow x + s$.

The two theorems below address the existence of steps meeting these criteria and

the convergence of sequences produced by such steps.

**Theorem 2.3.1 ([7])** *Let $x$ and $t \in (0,1)$ be given and assume that there exists an $\bar{s}$ that satisfies $\|F(x) + J(x)\bar{s}\| < \|F(x)\|$. Then there exists $\eta_{min} \in [0,1)$ such that, for any $\eta \in [\eta_{min}, 1)$, there is an $s$ satisfying $\|F(x) + J(x)s\| \leq \eta\|F(x)\|$ and $\|F(x+s)\| \leq [1 - t(1-\eta)]\|F(x)\|$.*

**Theorem 2.3.2 ([7])** *Suppose $\{x_k\}$ is produced by the global inexact Newton method. If $\sum_{k=0}^{\infty}(1-\eta_k) = \infty$, then $F(x_k) \to 0$. If, in addition, $x_*$ is a limit point of $\{x_k\}$ such that $J(x_*)$ is nonsingular, then $F(x_*) = 0$ and $x_k \to x_*$.*

The following sections present two of the most widely used criteria for determining an acceptable step, the Ared/Pred condition and the Goldstein-Armijo conditions. Both are shown to be special cases of the global inexact Newton method acceptability condition. These criteria in no way guarantee that a sequence of iterates converges to $x_*$ such that $F(x_*) = 0$; rather, it may diverge or converge to a non-zero local minimum of $\|F\|$ instead. In the numerical studies described later, the Ared/Pred choice was almost always used; it will be noted when the Goldstein-Armijo conditions were implemented.

### 2.3.1 Ared/Pred

This criterion accepts a step based on the ratio of the actual reduction in the function norm to the reduction predicted by the local linear model. If a lot of effort has been put into finding a step in which there is sufficient decrease in the local linear model norm, then it is quite reasonable to require the chosen step to give a sufficient decrease in $\|F\|$ relative to that of the model norm. The following notation is used [7]: Given

$x \in \mathbf{R}^n$ and a step $s \in \mathbf{R}^n$, define

- $ared \equiv \|F(x)\| - \|F(x+s)\|$, the actual reduction of $\|F\|$;
- $pred \equiv \|F(x)\| - \|F(x) + J(x)s\|$, the predicted reduction of $\|F\|$.

For exact Newton's method an acceptable decrease at a step $k$ is obtained when $ared \geq t \cdot pred$ for some fixed $t \in (0,1)$. If $s$ and $\eta$ have been chosen to satisfy $\|F(x) + J(x)s\| \leq \eta\|F(x)\|$ as well as this condition, then they also satisfy the sufficient decrease condition in the global inexact Newton method. Indeed, in this case,

$$\begin{array}{rrcl} & ared & \geq & t \cdot pred \\ \Rightarrow & \|F(x)\| - \|F(x+s)\| & \geq & t(\|F(x)\| - \|F(x) + J(x)s\|) \\ \Rightarrow & (1-t)\|F(x)\| & \geq & \|F(x+s)\| - t\|F(x) + J(x)s\| \\ & & \geq & \|F(x+s)\| - t\eta\|F(x)\| \\ \Rightarrow & (1 - t(1-\eta))\|F(x)\| & \geq & \|F(x+s)\| \end{array}$$

It should also be noted that a step satisfies $\|F(x) + J(x)s\| \leq \eta\|F(x)\|$ and the last inequality if and only if $pred \geq (1-\eta)\|F(x_k)\|$ and $ared \geq t(1-\eta)\|F(x_k)\|$. Thus the sufficient decrease condition in the global inexact Newton method can be viewed as an extension of the condition $ared \geq t \cdot pred$ to the inexact Newton context. In the following, it will be referred to as the Ared/Pred condition.

### 2.3.2 Goldstein-Armijo

The second set of criteria, the Goldstein-Armijo conditions, was developed for optimization problems, where the goal is solving $min_{x \in \mathbf{R}^n} f(x)$ by seeking an $x_*$ such that $\nabla f(x_*) = 0$. The first of these two conditions is similar to the Ared/Pred condition in that the next iterate should give a sufficient decrease relative to that predicted by the model. The criterion is called the **$\alpha$-condition** and with $f(x) \equiv \|F(x)\|_2^2$ is written as,

$$\|F(x_k + s)\|_2^2 \leq \|F(x_k)\|_2^2 + 2\alpha F(x_k)^T J(x_k)s, \quad \alpha \in (0,1) \tag{2.2}$$

The second condition, the $\beta$**-condition**, is also known as the curvature condition. Its purpose is to ensure that the steps are not too short to make adequate progress towards a solution:

$$F(x_k + s)^T J(x_k + s)s \geq \beta F(x_k)^T J(x_k)s, \quad \beta \in (\alpha, 1) \tag{2.3}$$

For an inexact Newton step, the Goldstein-Armijo $\alpha$-condition, like the Ared/Pred condition, implies the sufficient decrease condition of the global inexact Newton method [7]. Indeed, given $s$ and $\eta$ satisfying $\|F(x) + J(x)s\| \leq \eta\|F(x)\|$, the final term of the $\alpha$-condition can be rewritten as follows:

$$\begin{aligned} 2\alpha F^T Js &= 2\alpha F^T[-F + F + Js] \\ &= -2\alpha\|F\|_2^2 + 2\alpha F^T(F + Js) \\ &\leq -2\alpha(1-\eta)\|F\|_2^2 \end{aligned}$$

where the last inequality comes from the Cauchy-Schwarz inequality:

$$\begin{aligned} |F^T(F + Js)| &\leq \|F\|_2\|F + Js\|_2 \\ &\leq \eta\|F\|_2^2 \end{aligned}$$

Hence the $\alpha$-condition is now

$$\|F(x + s)\|_2^2 \leq [1 - 2\alpha(1-\eta)]\|F(x)\|_2^2$$

Since the left-hand side is nonnegative, it must be that $2\alpha(1-\eta) \leq 1$; since $\sqrt{1-\epsilon} \leq 1 - \epsilon/2$ whenever $|\epsilon| \leq 1$, then

$$\|F(x + s)\|_2 \leq [1 - \alpha(1-\eta)]\|F\|_2$$

which is exactly the sufficient decrease condition in the global inexact Newton method with $\alpha = t$.

# Chapter 3

# Specific Algorithms

The previous chapter defined a global inexact Newton method. Globalized Newton-GMRES methods are global inexact Newton methods which use the iterative linear solver GMRES to find an initial inexact Newton step. Section 3.1 is a detailed examination of the GMRES method. Section 3.2 describes four *globalization* methods, i.e., methods of deriving a satisfactory step when the initial inexact Newton step does not meet the global step acceptance criterion.

## 3.1 GMRES

At the heart of an inexact Newton method is the linear problem $J(x)s = -F(x)$. Our algorithm will use GMRES to find an approximate solution. The following motivation for and formulation of GMRES is a merging of the descriptions from [5] and [17].

GMRES is a particular *Krylov subspace method*. These methods constitute a class of algorithms designed to solve the linear problem: Find $x \in \mathbf{R}^n$ such that $Ax = b$, $A \in \mathbf{R}^{n \times n}$, $b \in \mathbf{R}^n$. A Krylov subspace method begins with an initial $x_0$ and at the

$k^{th}$ step, determines an iterate $x_k$ through a correction in the $k^{th}$ *Krylov subspace*

$$\mathcal{K}_k \equiv span\{r_0, Ar_0, \ldots, A^{n-1}r_0\} \tag{3.1}$$

where $r_0 \equiv b - Ax_0$ is the initial residual. A strong attraction of these methods is that implementations only require products $Av$ and sometimes $A^Tv$. Thus within the methods no direct access or manipulation of the entries of $A$ is required. GMRES uses only $Av$ products.

A key to deriving the GMRES algorithm is the Arnoldi process [1].

**Arnoldi Process:**
Given $r_0$.
Set $\rho_0 \equiv \|r_0\|_2$ and $q_1 \equiv r_0/\rho_0$.
For $k = 1, 2, \ldots$ do:
    Initialize $q_{k+1} = Aq_k$.
    For $i = 1, \ldots, k$ do:
        Set $h_{ik} = q_i^T q_{k+1}$.
        Update $q_{k+1} \leftarrow q_{k+1} - h_{ik}q_i$.
    Set $h_{k+1,k} = \|q_{k+1}\|_2$.
    Update $q_{k+1} \leftarrow q_{k+1}/h_{k+1,k}$.

Two important notes: the vectors $q_1, \ldots, q_k$ form an orthogonal basis for $\mathcal{K}_k$ and $AQ_k = Q_{k+1}H_k$, where $Q_k = [q_1, \ldots, q_k]$, and $H_k$ is a $(k+1) \times k$ upper Hessenberg matrix.

In GMRES, each iterate $x_k$ is chosen to minimize the residual norm over all corrections in $\mathcal{K}_k$. Using the Arnoldi process, the minimization problem can be reformulated into a low-dimensional least squares problem. Note that a correction vector $z_k \in \mathcal{K}_k$ can be written using the orthogonal basis $Q_k$ as $z_k = Q_k y_k$. Then:

$$\begin{aligned} \|r_k\|_2 &= \|b - Ax_k\|_2 \\ &= \|b - A(x_0 + z_k)\|_2 \\ &= \|r_0 - AQ_k y_k\|_2 \end{aligned}$$

$$
\begin{aligned}
&= \|r_0 - Q_{k+1} H_k y_k\|_2 \\
&= \|Q_{k+1}(\|r_0\|_2 e_1 - H_k y_k)\|_2 \\
&= \left\| \|r_0\|_2 e_1 - H_k y_k \right\|_2
\end{aligned}
$$

where $e_1 = (1, 0, \ldots, 0)^T \in \mathbf{R}^{k+1}$.

With the problem of minimizing $\|r\|_2$ transformed into a least squares problem for the entries of $y_k$, and the bonus of $H_k$ being an upper Hessenberg matrix, the $QR$ decomposition needed to solve for $y_k$ can be accomplished with $k$ *Givens rotations*. A Givens rotation $R(\theta) \equiv \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$ rotates a vector $x \in \mathbf{R}^2$ counterclockwise by $\theta$ degrees. If $\theta$ is chosen such that $\cos(\theta) = \frac{x_1}{\sqrt{x_1^2+x_2^2}}$ and $\sin(\theta) = \frac{-x_2}{\sqrt{x_1^2+x_2^2}}$ then $\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ 0 \end{bmatrix}$. In the decomposition of $H_k$, a Givens rotation is used to zero out the $(i+1)^{th}$ element of each column $i$ by applying it to the $i^{th}$ and $(i+1)^{th}$ rows, leaving an upper triangular matrix R.

The decomposition of $H_k$ gives $J_k \ldots J_1 H_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}$, where the $J_i$'s are the Givens rotations. Now set $w = J_k \ldots J_1 e_1 \|r_0\|_2$:

$$
\left\| e_1 \|r_0\|_2 - H_k y_k \right\|_2 = \left\| J_1^T \ldots J_k^T \left[ w - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y_k \right] \right\|_2 = \left\| w - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y_k \right\|_2 \qquad (3.2)
$$

where $R_k \in \mathbf{R}^{k \times k}$.

It follows that $R_k y_k = (w_1, \ldots, w_k)^T$ and the $(k+1)^{th}$ element of $w$ is the residual at the $k^{th}$ iteration, i.e., $\|r_k\|_2 = |w_{k+1}|$. Thus at each iteration it is a trivial matter to check the norm of the residual vector and subsequently stop the iterations when the residual norm has decreased below a specified tolerance.

As the iterations proceed the amount of storage required increases; it is on the order of $O(kn)$ for $k$ iterations. Also the number of multiplications increases with the introduction of each new vector. To overcome these steadily increasing costs, GMRES is usually restarted in practice. After a fixed number, $m$, of steps the current solution

$x_m$ is chosen to be the new initial vector, and GMRES is begun from scratch. The modified algorithm is GMRES($m$) [14].

**GMRES($m$):**
**Given:** $A, b, x, tol, itmax$.
**Initialize:** Set $r \equiv b - Ax$, $v_1 \equiv r/\|r\|_2$, $w \equiv \|r\|_2 e_1 \in (R)^{m+1}$.
**Iterate:** For $k = 1, \ldots, m$ do:
Initialize $v_{k+1} = Av_k$.
For $i = 1, \ldots, k$ do:
Set $h_{ik} = v_i^T v_{k+1}$.
Update $v_{k+1} \leftarrow v_{k+1} - h_{ik} v_i$.
Set $h_{k+1,k} = \|v_{k+1}\|_2$.
If $k > 1$, apply $J_{k-1} \ldots J_1$ to $(h_{1,k}, \ldots, h_{k,k}, h_{k+1,k}, 0, \ldots)^T \in \mathbf{R}^{m+1}$.
Determine $J_k$ such that

$$J_k \ldots J_1 \begin{pmatrix} h_{1,k} \\ \vdots \\ h_{k,k} \\ h_{k+1,k} \\ 0 \\ \vdots \end{pmatrix} \equiv \begin{pmatrix} r_{1,k} \\ \vdots \\ r_{k,k} \\ 0 \\ 0 \\ \vdots \end{pmatrix}.$$

If $k = 1$, form $R_1 \equiv (r_{11})$; else form $R_k \equiv \begin{pmatrix} R_{k-1} & r_{1,k} \\ & \vdots \\ 0 \ldots 0 & r_{k,k} \end{pmatrix}$.
Update $w \leftarrow J_k w$. If $|w_{k+1}| \leq tol$ or $k = m$, go to **Solve**; else
update $v_{k+1} \leftarrow v_{k+1}/h_{k+1,k}$.
**Solve:** Let $k$ be the final iteration number from **Iterate**.
Solve $R_k y = \bar{w}$ for $y$, where $\bar{w} \equiv (w_1, \ldots, w_k)^T$.
Update $x \leftarrow x + (v_1, \ldots, v_k)y$.
If $|w_{k+1}| \leq tol$, accept $x$; otherwise, return to **Initialize**.

## 3.2 Line Search and Trust Region Methods

A global inexact Newton method imposes one or more additional criteria on the steps of an inexact Newton method to enhance the likelihood that $x_k \to x_*$. Assume the

method has chosen a step approximating the Newton step, and furthermore, that the step does not satisfy the sufficient decrease condition imposed by either the Ared/Pred or Goldstein-Armijo conditions (whichever was chosen to be implemented). A globalization method is then invoked to find a satisfactory step. Two of the most widely used classes of these methods are *line search methods* and *trust region methods*. Line search methods use the direction of the inexact Newton step, and an acceptable step is sought by systematically changing the length of the step. A trust region method first chooses a step length within which the local linear model is 'trusted', and then seeks a step which minimizes $\|F(x) + J(x)s\|$ over all steps of length less than or equal to the trusted length.

### 3.2.1 Line Searches

Line search methods consider the 1-dimensional cross section of $\|F(x)\|$ that intersects the points $\|F(x_k)\|$ and $\|F(x_k + s_k)\|$. The goal is to then find a $\lambda_k \in (0, 1]$ such that $x_{k+1} = x_k + \lambda_k s_k$ is an acceptable step.

This approach is a feasible method. If an initial inexact Newton step has been found, then a sufficiently shortened step $s_k$ having the same direction will satisfy both an inexact Newton condition $\|F(x_k) + J(x_k)s_k\| \leq \eta_k \|F(x_k)\|$ and the corresponding sufficient decrease condition $\|F(x_k + s_k)\| \leq [1 - t(1 - \eta_k)]\|F(x_k)\|$; see [7, p. 410].

The next three sections present different methods of obtaining a $\lambda$ that yields a satisfactory next step. In the developments of all three algorithms, it is assumed that the norm is the Euclidean norm, although the developments can be easily adapted to allow any inner-product norm. Also, in implementations, usually some restriction is put on the size of $\lambda$, for example $\lambda \in (\lambda_{min}, \lambda_{max})$ for some $0 < \lambda_{min} < \lambda_{max} < 1$.

**Quadratic Interpolation**

Take a cross-section of $F$ along the line $x_k + \lambda s_k$, $\lambda \in [0, 1]$. For ease of notation the index $k$ will be dropped. The problem is to minimize $\|F(x + \lambda s)\|$, or similarly $\|F(x + \lambda s)\|^2$, with respect to $\lambda$. This 1-dimensional minimization problem can be very expensive to solve exactly, so the idea is to use interpolation to get an easily minimizable approximation to $\|F(x + \lambda s)\|^2$.

Finding a quadratic polynomial approximation to the function will require three interpolating values. Define $g(\lambda) = \|F(x + \lambda s)\|^2$. Then $g(0) = \|F(x)\|^2$ is already known and, since $s$ has been unsuccessfully tested for sufficient reduction, so is $g(1) = \|F(x + s)\|$. The third interpolation value is $g'(0) = 2F^T Js$. Using the three points $g(0)$, $g(1)$, and $g'(0)$ a quadratic model of $\|F(x+\lambda s)\|^2$ can be built, a minimizer can be found, and the new trial step can be checked. The quadratic model is

$$p(\lambda) = [g(1) - g(0) - g'(0)]\lambda^2 + g'(0)\lambda + g(0). \tag{3.3}$$

Then the derivatives are:

$$\begin{aligned} p'(\lambda) &= 2[g(1) - g(0) - g'(0)]\lambda + g'(0) \\ p''(\lambda) &= 2[g(1) - g(0) - g'(0)] \end{aligned}$$

If $p''(\lambda) \leq 0$ then the quadratic is concave down, so choose $\lambda = \lambda_{max}$. If $p''(\lambda) > 0$ then find the $\lambda$ such that $p'(\lambda) = 0$:

$$\begin{aligned} 0 = p'(\lambda) &= 2[g(1) - g(0) - g'(0)]\lambda + g'(0) \\ \Rightarrow \lambda &= \tfrac{-g'(0)}{2[g(1)-g(0)-g'(0)]} \end{aligned}$$

Correcting for $\lambda \in (\lambda_{min}, \lambda_{max})$ if necessary, one obtains $\lambda$, the fraction used to reduce the step length. Now updating $x \leftarrow \lambda s$ and $\eta \leftarrow 1 - \lambda(1 - \eta)$, check to see if the new step $s$ and $\eta$ satisfy the reduction criterion $\|F(x + s)\| \leq [1 - t(1 - \eta)]\|F\|$. If this is not satisfied, repeat the process.

### Cubic Interpolation

Cubic interpolation is almost like quadratic interpolation: instead of using three interpolation points, four points are used and a cubic polynomial is constructed. On the first step reduction there is no clear way to choose a fourth point, so just three are chosen, and a quadratic polynomial is used. On subsequent reductions, however, four points can be found. The two points $g(0)$ and $g'(0)$ are used as well as the values of $g$ at the two previous $\lambda$ values. For example the second reduction uses $g(0)$, $g'(0)$, $g(\lambda_1)$, and $g(1)$, the fourth backtracking step will use $g(0)$, $g'(0)$, $g(\lambda_3)$, and $g(\lambda_2)$, etc.

As done in [6] denote the two previous $\lambda$ values as $\lambda_{prev}$ and $\lambda_{2prev}$. The cubic polynomial approximation of the function $\|F(x+\lambda s)\|^2$ then becomes

$$p(\lambda) = a\lambda^3 + b\lambda^2 + g'(0)\lambda + g(0)$$

with

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\lambda_{prev} - \lambda_{2prev}} \begin{bmatrix} \frac{1}{\lambda_{prev}^2} & \frac{-1}{\lambda_{2prev}^2} \\ \frac{-\lambda_{2prev}}{\lambda_{prev}^2} & \frac{\lambda_{prev}}{\lambda_{2prev}^2} \end{bmatrix} \begin{bmatrix} g(\lambda_{prev}) - g(0) - g'(0)\lambda_{prev} \\ g(\lambda_{2prev}) - g(0) - g'(0)\lambda_{2prev} \end{bmatrix}$$

The local minimizer of the model is given by $\lambda_+ = \frac{-b+\sqrt{b^2-3ag'(0)}}{3a}$. As with the quadratic line search, let $s \leftarrow \lambda s$ and $\eta \leftarrow 1 - \lambda(1-\eta)$ and check if the new $s$ and $\eta$ satisfy the reduction criterion. If not, iterate.

### Moré-Thuente

The Moré-Thuente algorithm [11] for choosing an acceptable step is a two-step process. First an interval of $\lambda$-values is chosen in a way which guarantees that there exists a $\lambda_*$ within the interval such that $\lambda_* s$ satisfies the sufficient decrease and curvature

conditions of the algorithm, given in (3.4) and (3.5) below. Second a point is chosen within the interval and tested. If it is not satisfactory, the interval is contracted, yielding a smaller interval still containing a $\lambda_*$, and the process iterates:

Given $\lambda_0 \in [\lambda_{min}, \lambda_{max}]$.
Set $I_0 = [0, \infty]$.
For $k = 0, 1, \ldots$
    Choose a safeguarded $\lambda_k \in I_k \bigcap [\lambda_{min}, \lambda_{max}]$.
    Test for convergence.
    Update the interval $I_k$.

An important feature of the algorithm is its ability to produce $\lambda$ values greater than one. The algorithm can produce steps longer than the original step if there is evidence that $\|F\|$ is decreasing sufficiently rapidly at the initial step.
The following paragraphs describe in detail the algorithms used for the interval updates and test point choices.

First, some general notation: let $\phi(\lambda) \equiv \frac{1}{2}\|F(x+\lambda s)\|^2$ and $\psi(\lambda) \equiv \phi(\lambda) - \phi(0) - \mu\phi'(0)\lambda$. The Moré-Thuente algorithm is designed to find a $\lambda$ satisfying the following two criteria:

$$\phi(\lambda) \leq \phi(0) + \mu\phi'(0)\lambda \tag{3.4}$$

and

$$|\phi'(\lambda)| \leq \zeta|\phi'(0)| \tag{3.5}$$

for $\mu$ and $\zeta$ in $(0, 1)$. The first condition is equivalent to the $\alpha$-condition, and the second is stronger than the $\beta$-condition with $\zeta = \beta$:

$$|\phi'(\lambda)| \leq \zeta|\phi'(0)| \Rightarrow \phi'(\lambda) \geq \zeta\phi'(0) \Rightarrow F(x+s)^T J(x+s)s \geq \zeta F(x)^T J(x)s \tag{3.6}$$

Therefore, if the method succeeds, the step found will satisfy the inexact Newton condition and sufficient decrease condition of the global method, at least if the final $\lambda$ satisfies $\lambda \leq 1$ and the initial $\eta$ is updated by $\eta \leftarrow 1 - \lambda(1 - \eta)$.

Define the set $T(\mu) \equiv \{\lambda > 0 : \phi(\lambda) \leq \phi(0) + \mu\phi'(0)\lambda, |\phi'(\lambda)| \leq \mu|\phi'(0)|\}$. The design of the algorithm guarantees a $\lambda$ in $T(\mu)$; but without additional restraints, there is no guarantee that a $\lambda$ satisfying criteria (3.4) and (3.5) can be found.

The first goal is to find a satisfactory interval that contains a $\lambda_* \in T(\mu)$. There exist conditions on the endpoints of an interval $I$ that guarantee it has a nonnempty intersection with $T(\mu)$.

**Theorem 3.2.1 ([11])** *Let $I$ be a closed interval with endpoints $\lambda_l$ and $\lambda_u$. If the endpoints satisfy*

$$\psi(\lambda_l) \leq \psi(\lambda_u), \quad \psi(\lambda_l) \leq 0, \quad \psi'(\lambda_l)(\lambda_u - \lambda_l) < 0,$$

*then there is a $\lambda_*$ in $I$ with $\psi(\lambda_*) \leq \psi(\lambda_l)$ and $\psi'(\lambda_*) = 0$. In particular, $\lambda_* \in (T(\mu) \bigcap I)$.*

Assuming there is a maximum allowable step length, $\lambda_{max} > 0$, such that $\psi(\lambda_{max}) > \psi(0)$, the interval $[0, \lambda_{max}]$ satisfies the assumptions of the theorem. Trivially, $\psi(0) = 0$, and with the assumption on $\lambda_{max}$ the first two conditions are satisfied. Also, $\psi'(0) = \phi'(0) - \mu\phi'(0) = (1 - \mu)\phi'(0)$, but $\phi'(0) < 0$, so $\psi'(0)(\lambda_{max} - 0) < 0$.
For now, assume that at step $k$ of the algorithm, a trial $\lambda_k$ has been found in $I_k$ with endpoints $\lambda_l^k$ and $\lambda_u^k$, but $\lambda_k \notin T(\mu)$, i.e., is unacceptable; then the interval must be updated. The updating algorithm is a conditional update based on three possibilities.

**Updating Algorithm**[11]:
Given a trial value $\lambda_k$ in $I_k$, the endpoints $\lambda_l^{k+1}$ and $\lambda_u^{k+1}$ of the updated interval $I_{k+1}$ are determined as follows:
Case U1: If $\psi(\lambda_k) > \psi(\lambda_l^k)$, then $\lambda_l^{k+1} = \lambda_l^k$ and $\lambda_u^{k+1} = \lambda_k$.
Case U2: If $\psi(\lambda_k) \leq \psi(\lambda_l^k)$ and $\psi'(\lambda_k)(\lambda_l^k - \lambda_k) > 0$, then $\lambda_l^{k+1} = \lambda_k$ and $\lambda_u^{k+1} = \lambda_u^k$.
Case U3: If $\psi(\lambda_k) \leq \psi(\lambda_l^k)$ and $\psi'(\lambda_k)(\lambda_l^k - \lambda_k) < 0$, then $\lambda_l^{k+1} = \lambda_k$ and $\lambda_u^{k+1} = \lambda_l^k$.

If the endpoints of the original interval $I_k$ satisfy the conditions of the previous theorem, then so do the endpoints of the updated interval $I_{k+1}$. Therefore $[I_{k+1} \bigcap T(\mu)]$

remains nonempty. It is possible that case U2 holds for every $k$, in which case the algorithm should eventually end with $\lambda_k = \lambda_{max}$ for some $k$. In an implementation this is accomplished by choosing $\lambda_{k+1} \in [min\{\delta_{max}\lambda_k, \lambda_{max}\}, \lambda_{max}]$, with $\delta_{max} > 1$, when case U2 holds for $\lambda_k$.

**Theorem 3.2.2 ([11])** *The search algorithm produces a sequence $\lambda_k$ in $[\lambda_{min}, \lambda_{max}]$ such that after a finite number of trial values one of the following conditions holds:*
*The search terminates at $\lambda_{max}$, the sequence of trial values is increasing, and $\psi(\lambda_k) \leq 0$ and $\psi'(\lambda_k) < 0$ for each $k$.*
*The search terminates at $\lambda_{min}$, the sequence of trial values is decreasing, and $\psi(\lambda_k) > 0$ or $\psi'(\lambda_k) \geq 0$ for each $k$.*
*An interval $I_k \subset [\lambda_{min}, \lambda_{max}]$ is generated.*

Termination at either $\lambda_{min}$ or $\lambda_{max}$ can be ruled out if they are chosen properly; see ([11, p. 293]). Thus the algorithm can be made to terminate in a finite number of steps with a $\lambda_k \in T(\mu)$. But a $\lambda_k \in T(\mu)$ only satisfies $|\phi'(\lambda_k)| \leq \mu|\phi'(0)|$, not the desired curvature condition (3.5) unless $\zeta \leq \mu$.

A $\lambda$ satisfying the curvature condition for a general $\zeta$ does not always exist, but it is possible to show that if while searching for a $\lambda_* \in T(\mu)$ an iterate $\lambda_k$ satisfying $\psi(\lambda_k) \leq 0$ and $\phi'(\lambda_k) > 0$ is found, then there exists an interval containing a $\lambda$ satisfying both (3.4) and (3.5). Further it is possible to modify the search algorithm to find this $\lambda$: simply replace the $\psi$'s with $\phi$'s.

**Modified Updating Algorithm**[11]:
Given a trial value $\lambda_k$ in $I_k$, the endpoints $\lambda_l^{k+1}$ and $\lambda_u^{k+1}$ of the updated interval $I_{k+1}$ are determined as follows:
Case U1: If $\phi(\lambda_k) > \phi(\lambda_l^k)$, then $\lambda_l^{k+1} = \lambda_l^k$ and $\lambda_u^{k+1} = \lambda_k$.
Case U2: If $\phi(\lambda_k) \leq \phi(\lambda_l^k)$ and $\phi'(\lambda_k)(\lambda_l^k - \lambda_k) > 0$, then $\lambda_l^{k+1} = \lambda_k$ and $\lambda_u^{k+1} = \lambda_u^k$.
Case U3: If $\phi(\lambda_k) \leq \phi(\lambda_l^k)$ and $\phi'(\lambda_k)(\lambda_l^k - \lambda_k) < 0$, then $\lambda_l^{k+1} = \lambda_k$ and $\lambda_u^{k+1} = \lambda_l^k$.

This modified updating algorithm then is used in conjunction with the original updating algorithm. The original algorithm is used until an iterate satisfying $\psi(\lambda_k) \leq 0$ and $\phi'(\lambda_k) > 0$ is found and then the modified updating algorithm is used for all subsequent iterations. There is no guarantee that the first will find an iterate satisfying $\psi(\lambda_k) \leq 0$ and $\phi'(\lambda_k) > 0$. If a $\lambda_k$ satisfying these conditions cannot be found, then the line search fails.

The next detail of the implementation is choosing the $\lambda_{k+1}$ in the updated interval $I_{k+1}$. Like the interval updates, the choices made here are based on the function and its derivative values at the endpoints of the interval. There are four different cases that cover all eventualities. At the current iteration, it is assumed that the following are known: the endpoints of the current interval, $(\lambda_l^k, \lambda_u^k)$, the previous trial value $\lambda_{k-1}$, and the function values and derivatives at all three of those points. The function and its derivatives are based on either $\psi$ or $\phi$ depending upon which interval updating function is currently being used. Denote the function values as $f_l$, $f_u$, $f_{k-1}$ and the derivative values as $g_l$, $g_u$, $g_{k-1}$. Also three more points are needed in the calculations of the new $\lambda_k$: $\lambda_c$, the minimizer of the cubic that interpolates $f_l$, $f_{k-1}$, $g_l$, $g_{k-1}$; $\lambda_q$, the minimizer of the quadratic that interpolates $f_l$, $f_{k-1}$, $g_l$; and $\lambda_s$, the minimizer of the quadratic that interpolates $f_l$, $f_{k-1}$, $g_{k-1}$. The four cases are listed here; for details on why they are chosen, see [11].

- Case 1: $f_{k-1} > f_l$ then

$$\lambda_k = \begin{cases} \lambda_c, & if \quad |\lambda_c - \lambda_l| < |\lambda_q - \lambda_l| \\ \frac{1}{2}(\lambda_q + \lambda_c), & otherwise. \end{cases}$$

- Case 2: $f_{k-1} \leq f_l$ and $g_{k-1} g_l < 0$ then

$$\lambda_k = \begin{cases} \lambda_c, & if \quad |\lambda_c - \lambda_l| \geq |\lambda_s - \lambda_{k-1}| \\ \lambda_s, & otherwise. \end{cases}$$

- Case 3: $f_{k-1} \leq f_l$, $g_{k-1}g_l \geq 0$, and $|g_{k-1}| \leq |g_l|$ then

$$\lambda_k = \begin{cases} \lambda_c, & if \quad |\lambda_c - \lambda_{k-1}| < |\lambda_s - \lambda_{k-1}| \\ \lambda_s, & otherwise. \end{cases}$$

- Case 4: $f_{k-1} \leq f_l$, $g_{k-1}g_l \geq 0$, and $|g_{k-1}| > |g_l|$ then

$$\lambda_k = \text{the minimizer of the cubic that interpolates } f_u, f_{k-1}, g_u, \text{ and } g_{k-1}.$$

Barring some minor implementational details of bounds and roundoff error checks, this is the algorithm implemented for the numerical studies described later. When successful, the algorithm produces a suitable $\lambda$ that satisfies (3.4) and (3.5).

### 3.2.2 Trust Region

The overall goal of the trust region method is the same as that of the line search methods: If the chosen inexact Newton step is unacceptable, then a step of new length, which is acceptable, must be chosen. Here, however, instead of just searching for a new step length in the inexact Newton direction, a step length is chosen based on a level of 'trust' in the local linear model, and then the direction of the step is chosen.

The level of 'trust' in the local linear model is based on how well the local linear model has approximated the function in previous iterations. If the model has been an accurate representation of the function than a longer step is justifiable, but if the model and function have had large discrepancies in past iterations then a shorter step is warranted. The length of the steps for which the local model is 'trusted' is denoted by $\delta$ and called the *trust region radius*. If the step at a given radius is inadequate, the radius is adjusted and a new step is computed. Dennis and Schnabel in [6] suggest the following criteria for adjusting the radius based upon the agreement of the local linear model and the function.

- if $ared \geq 0.75 \cdot pred$ then $\delta_+ \leftarrow 2\delta$
- if $ared < 0.1 \cdot pred$ then $\delta_+ \leftarrow \frac{1}{2}\delta$
- Otherwise $\delta_+ \leftarrow \delta$

However in the implementation used for our numerical tests, the defaults used were as follows:

- if $ared \geq 0.75 \cdot pred$ then $\delta_+ \leftarrow 4\delta$
- if $ared < 0.1 \cdot pred$ then $\delta_+ \leftarrow \frac{1}{4}\delta$
- Otherwise $\delta_+ \leftarrow \delta$

With a $\delta$ chosen, the length of a step from the current point is given an upper bound. The idealized trust region method will then find a step $s$ of length $\leq \delta$ that minimizes the norm of the local linear model, i.e., $s \in argmin_{\|w\| \leq \delta} \|F(x) + J(x)w\|$.

**Lemma 3.2.3 ([17])** *If J(x) is nonsingular, then* $s \in argmin_{\|w\| \leq \delta} \|F(x) + J(x)w\|$ *is given by*

$$s = s(\mu) \equiv -[J(x)^T J(x) + \mu I]^{-1} J(x)^T F(x)$$

*for a unique* $\mu \geq 0$*, as follows:*

$$\begin{cases} \|s^N\|_2 \leq \delta \Rightarrow \mu = 0, \\ \|s^N\|_2 > \delta \Rightarrow \mu > 0, \ \textit{uniquely determined by} \|s(\mu)\|_2 = \delta. \end{cases}$$

where $s^N$ is the Newton step.

One trust region method is the dogleg method [13]. The idea of the dogleg method is to find an approximation to the curve $s(\mu)$ and find a step $s$ such that $\|s\|_2 = \delta$. The approximation of the curve is given by a polygonal arc connecting three points: the current point, the Newton point $s^N$, and finally the steepest descent direction point, defined as follows:

**Def 3.2.4 ([17])** *The steepest descent point* $s^{SD}$ *is the minimizer of* $l(s) \equiv \frac{1}{2}\|F(x)+J(x)s\|_2^2$ *in the steepest descent direction* $-\nabla l(0) = -J(x)^T F(x)$*. It is easy to verify that*

$$s^{SD} = \frac{\|-J(x)^T F(x)\|_2^2}{\|J(x)J(x)^T F(x)\|_2^2} J(x)^T F(x).$$

The dogleg curve (denoted $\Gamma_{DL}$ in [17]) is the piecewise linear curve from $x_c$ to $s^{SD}$ to $s^N$. In the line-search methods defined above, the curve on which a step is sought has the important property that $\|F(x) + J(x)\lambda s\|$ is a strictly decreasing function in the size of $\lambda$. Dennis and Schnabel point out in [6] that the dogleg curve similarly has this property. In addition $\|s\|$ is monotone strictly increasing along the curve. So there exists a unique $s \in \Gamma_{DL}$ such that $\|s\| = \delta$ and $s \in argmin_{w\in\Gamma_{DL},\|w\|\leq\delta}\|F(x)+J(x)w\|$ [6]. The complete algorithm for choosing the dogleg step is:

**Computing the Dogleg Step**[17]:

Assume $s^N = -J(x)^{-1}F(x)$ has already been computed.

1. If $\|s^N\|_2 \leq \delta$, then $s = s^N$.
2. If $\|s^N\|_2 < \delta$, then do:
   (a) Compute $s^{SD}$.
   (b) If $\|s^{SD}\|_2 \geq \delta$, then $s = \frac{\delta}{\|s^{SD}\|_2} s^{SD}$.
   (c) If $\|s^{SD}\|_2 < \delta$ then $s = s^{SD} + \tau(s^N - s^{SD})$, where $\tau$ is uniquely determined by $\|s^{SD} + \tau(s^N - s^{SD})\|_2 = \delta$.

The entire algorithm is:

Given the Newton step $s^N$ and initial trust region radius $\delta$.
Calculate the dogleg step $s$ along $\Gamma_{DL}$.
Check to see if the new step $s$ is satisfactory.
If not, update $\delta$ and repeat. Otherwise accept $s$.

For the inexact Newton case the Newton step is replaced with the inexact Newton step: $s^{IN} \approx -J(x)^{-1}F(x)$. In this case, the previously given properties of the dogleg

curve do not necessarily hold. In our numerical tests, the inexact Newton step is chosen to approximate the Newton step 'well enough' that the properties of the dogleg curve are likely to hold. This is accomplished by using a very small constant forcing term.

# Chapter 4

# Testing Environment

The coding of all of the above algorithms and the associated numerical test problems into a parallel computing environment would be a very large undertaking, well beyond the scope of this thesis. Therefore, previously written codes that implement the numerical algorithms and formulate the flow problems were combined to produce the results of the numerical study. All codes were developed at Sandia National Laboratories in Albuquerque New Mexico. The testing environment was a multi-processor machine housed at Sandia National Labs.

## 4.1 NOX

NOX is short for 'Nonlinear Object-Oriented Solutions' and is one of the software packages written for the Trilinos project. From [9]:

> The Trilinos Project is an effort to develop and implement robust parallel algorithms using modern object-oriented software design, while still leveraging the value of established numerical libraries such as PETSc, Aztec,

> the BLAS and LAPACK. It emphasizes abstract interfaces for maximum flexibility of component interchanging, and provides a full-featured set of concrete classes that implement all abstract interfaces.... An over-riding emphasis of the Trilinos Project is to develop robust solution algorithms for scientific and engineering applications on parallel computers, and make these algorithms accessible to application developers in the most effective way.

The NOX package has been developed to bring nonlinear solution methods into the Trilinos framework. It includes implementations of the line search and trust region methods discussed previously as well as other methods such as tensor methods. Within each method exist numerous parameters that can be modified for particular problems. For example, the trust region method has options for setting the expansion and contraction ratios of the trust region radius. The flexibility of the code and algorithm adjustments allow the solvers to efficiently solve a large spectrum of difficult nonlinear problems.

The two leading developers of the code are Tamara G. Kolda and Roger P. Pawlowski at Sandia National Laboratories, though many others have contributed. A short list of other developers includes Russell Hooper, Eric T. Phipps, Andrew G. Salinger and Brett W. Bader. All are employees of Sandia National Labs except Brett Bader, who was a SNL summer researcher in 2002.

## 4.2 MPSALSA

MPSalsa is a finite element computer program for reacting flow problems. Like NOX, it was developed at Sandia National Laboratories in Albuquerque, New Mexico. The

principal developers were John N. Shadid, Harry K. Moffat, Scott A. Hutchinson, Gary L. Hennigan, Karen D. Devine, and Andrew G. Salinger. Many more people also contributed to the code and development of the program. The theoretical conception and development of the code are not central to this work; therefore, only the description of the code from the abstract of the Sandia report [15] will be given:

> MPSalsa is designed to solve laminar, low Mach number, two- or three-dimensional incompressible and variable density reacting fluid flows on massively parallel computers, using a Petrov-Galerkin finite element formulation. The code has the capability to solve coupled fluid flow, heat transport, multicomponent species transport, and finite-rate chemical reactions, and to solve coupled multiple Poisson or advection-diffusion-reaction equations. The program employs the CHEMKIN library to provide a rigorous treatment of multicomponent ideal gas kinetics and transport. Chemical reactions occurring in the gas phase and on surfaces are treated by calls to CHEMKIN and SURFACE CHEMKIN, respectively. The code employs unstructured meshes, using the EXODUS II finite element database suit of programs for its input and output files. MPSalsa solves both transient and steady flows by using fully implicit time integration, an inexact Newton method and iterative solvers based on preconditioned Krylov methods as implemented in the Aztec solver library.

The importance of the MPSalsa code, in relation to this work, is the large number of difficult problems it makes available to the nonlinear solver NOX. Using problems formulated and implemented in the MPSalsa software, it is possible to thoroughly test the NOX algorithms, and ultimately compare the performance of numerous globalized inexact Newton methods on some of the more computationally challenging large-scale

problems. In the following paragraphs some of the equations used in MPSalsa are given. All of these are used in the later test problems.

**Momentum Transport Equation:** The following two equations express the conservation of momentum laws.

$$\rho \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \mathbf{T} - \rho \mathbf{g} = 0 \tag{4.1}$$

$$\mathbf{T} = -P\mathbf{I} + \mu\{\nabla \mathbf{u} + \nabla \mathbf{u}^T\} \tag{4.2}$$

Here, $\mu$ is the velocity vector, $\rho$ is the mass density of the mixture, $\mathbf{T}$ is the stress tensor for a Newtonian fluid, $\mathbf{I}$ is the unity tensor, $g$ is the gravity vector, $\mu$ is the dynamic viscosity and $P$ is the isotropic hydrodynamic pressure.

**Total Mass Conservation Equation:** Conservation of total mass within MP-Salsa is expressed by

$$\nabla \cdot (\rho \mathbf{u}) = 0 \tag{4.3}$$

Here, $\rho$ is either considered to be a constant or is calculated from the ideal gas mixture equation of state. Thus, for an ideal gas, $\rho$ is a function of the constant thermodynamic pressure only.

**Energy Transport Equation(Temperature Formulation Simplified):**

$$\rho C_p \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{q} = 0 \tag{4.4}$$

$$\mathbf{q} = -\kappa \nabla T \tag{4.5}$$

Here, $T$ is the temperature, $C_p$ is the specific heat at constant pressure, and $\kappa$ is the thermal conductivity.

## 4.3 Parallel Machine[10]

The machine on which all numerical tests were performed is an IBM cluster located at Sandia National Laboratories in Albuquerque New Mexico. Details of the hardware

and software are given in the two subsections below.

### 4.3.1 Cluster Hardware

- One head node. Dual-processor 1 GHz Pentium III; 1 GB RAM; single 70 GB hard drive (66 GB after partitioning).
- Sixteen compute nodes: Each with a dual-processor 1 GHz Pentium III; 1 GB RAM; single 18 GB hard drive (16 GB after partitioning).
- Myrinet network between the 16 compute nodes for message passing.
- The cluster internal network consists of the head node being connected to a switch by 1GB Ethernet, with each compute node connected to that switch with 100 MB Ethernet.

### 4.3.2 Cluster Software

- Operating system: all nodes run Red Hat Linux 7.2.
- Standard GNU compilers: gcc/g++ (version 2.96) and g77.
- MPI message passing by MPICH 1.2.1 or LAM 6.5.6.
- mpicc/mpiCC and mpif77/mpif90 for MPICH-GM.
- mpicc/mpiCC and mpif77 for LAM-IP.
- Resource manager: Portable Batch System (PBS) (http://www.openpbs.org).
- Job scheduling for compute nodes is performed by the Maui scheduler (http://supercluster.org) rather than using PBS's own scheduler.
- The cluster software toolkit is xcat (http://x-cat.org).

# Chapter 5

# Benchmark Test Problems

The three test problems described below are standard benchmark problems used for verification of fluid flow codes and solution algorithms[16]. The initial guess for all runs of these problems is the all-zeroes vector.

## 5.1 Thermal Convection[16]

This problem is to determine the thermally-driven convection flow of a fluid 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 defined above on the unit square in $\mathbf{R}^2$ or unit cube in $\mathbf{R}^3$. In 2D the following Dirichlet and Neumann boundary conditions are imposed:

$$\begin{aligned}
T = T_{cold}, &\quad u = v = 0 \quad at\ x = 0, \\
T = T_{hot}, &\quad u = v = 0 \quad at\ x = 1, \\
\tfrac{\partial T}{\partial y} = 0, &\quad u = v = 0 \quad at\ y = 0, \\
\tfrac{\partial T}{\partial y} = 0, &\quad u = v = 0 \quad at\ y = 1.
\end{aligned}$$

When the equations and boundary conditions are suitably nondimensionalized one of the parameters to appear is the Rayleigh number, Ra. For this study the values of Ra used were $10^3, 10^4, 10^5, 10^6$. As the number increases the nonlinear effects of the convection terms increase and the solution becomes increasingly difficult to obtain. All solutions in 2D were computed on a 100x100 equally spaced mesh, which resulted in 40,804 unknowns for the discretized problem.

In 3D the boundary conditions are very similar:

$$\begin{aligned}
T = T_{cold}, &\quad u = v = w = 0 \quad at\ x = 0\ plane,\\
T = T_{hot}, &\quad u = v = w = 0 \quad at\ x = 1\ plane,\\
\tfrac{\partial T}{\partial y} = 0, &\quad u = v = w = 0 \quad at\ y = 0\ plane,\\
\tfrac{\partial T}{\partial y} = 0, &\quad u = v = w = 0 \quad at\ y = 1\ plane,\\
\tfrac{\partial T}{\partial z} = 0, &\quad u = v = w = 0 \quad at\ z = 0\ plane,\\
\tfrac{\partial T}{\partial z} = 0, &\quad u = v = w = 0 \quad at\ z = 1\ plane.
\end{aligned}$$

Here the same Ra numbers are used, the mesh is a 32x32x32 equally spaced grid which resulted in 179,685 unknowns for the discretized problem.

## 5.2 Backward Facing Step[16]

The second problem simulates a reentrant backward-facing step by introducing a fully developed parabolic velocity profile in the upper half of the inlet boundary and imposing zero velocity on the lower half. The problem requires the solution of the momentum transport equation and the total mass conservation equation with $\rho = 1$, or $\nabla \cdot \mathbf{u} = 0$. The nondimensionalized formulation contains the Reynolds number. As this parameter is increased the nonlinear components of the equation 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. The discretization is a 20x400 unequally spaced mesh which resulted in 25,263 unknowns. The boundary conditions are given by

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

For the studies performed here, the Reynolds number was given values of 100, 200,..., 700, 750 and 800.

## 5.3 Lid Driven Cavity[16]

The third problem studied is that of a confined fluid flow in a square box. The discretized equations for this problem are the same two used in the Backward Facing Step problem. In 2D the two sides and bottom are held fixed while the top is moving from left to right. The discretization of the domain is a 100x100 equally spaced grid. The following Dirichlet boundary conditions are applied:

$$\begin{aligned} u = v = 0 \quad & at\ x = 0, \\ u = v = 0 \quad & at\ x = 1, \\ u = v = 0 \quad & at\ y = 0, \\ u = U_0, v = 0 \quad & at\ y = 1. \end{aligned}$$

Again a suitable nondimensionalized formulation leads to the appearance of the Reynolds number. As this parameter is increased the nonlinear inertial terms in the momentum equation become more dominant and the solution becomes more difficult to obtain. For the tests performed in 2D the Reynolds number ranged from

1,000 to 10,000 in increments of 1,000. The discretization of the problem led to 30,603 unknowns.

In 3D, the problem is formulated on a 1x1x1 cube with a moving lid. The discretization is a 32x32x32 equally spaced grid. Again Dirichlet boundary conditions are applied:

$$\begin{aligned}
u = v = w = 0 \quad & at\ x = 0\ plane, \\
u = v = w = 0 \quad & at\ x = 1\ plane, \\
u = v = w = 0 \quad & at\ z = 0\ plane, \\
u = v = w = 0 \quad & at\ z = 1\ plane, \\
u = v = w = 0 \quad & at\ y = 0\ plane, \\
u = U_0, v = w = 0 \quad & at\ y = 1\ plane.
\end{aligned}$$

The Reynolds number ranged from 100 to 1,000 in increments of 100 and the discretization of the problem led to 143,748 unknowns.

# Chapter 6

# Results

## 6.1 Success vs. Failure

In numerical methods it is almost always impossible to find an $x$ such that $F(x) = 0$. The best a numerical solver can do is find an $x$ such that $\|F(x)\| \leq tol$. The $tol$ is set by the user and determines how 'close' a computed $x$ must be to the real solution to declare a successful termination.

In our numerical studies a 'success' was declared if an $x_k$ was found satisfying $\|F(x_k)\|_2 \leq 10^{-2}\|F(x_0)\|_2$ and $WRMS \leq 1$. The $WRMS$ is defined as the weighted root mean square norm, $WRMS = \|x_k - x_{k-1}\|_{wrms}$. "The $wrms$ norm is defined by: $\|\delta x^k\|_{wrms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{C\cdot(x_i^k - x_i^{k-1})}{RTOL|x_i^{k-1}| + ATOL_i}\right)^2}$, where $N$ is the total number of unknowns in the problem, $x_i^k$ denotes the $i$ -th component of the solution vector $x$ at nonlinear iteration $k$, $RTOL$ is the relative error tolerance (a scalar value), $ATOL_i$ is the absolute error tolerance and can be a scalar or a vector of the same size as the solution vector, and $C$ is a scalar value that is typically set to 1.0."[9] This second test checks to see that the change in the solution from one step to the next has not been too large. Failures occur when both of the above two criteria are not met

within a specified number of inexact Newton steps. The number of steps specified for individual problems is thought to be reasonable based on the number of solutions that were obtained within that limit.

The reasons for failure vary from one problem to the next. Some recurring reasons were: a singular Jacobian, convergence to local norm minima, and progress that was simply too slow. This is by no means an exhaustive list of reasons for failure, but it does account for the majority of failures encountered in this study.

## 6.2 Relevant Numbers

Each problem was run using nine different algorithms: quadratic interpolation, cubic interpolation, and Moré-Thuente backtracking, each with a constant forcing term and an adaptive forcing term; the trust-region dogleg method with a constant forcing term; and finally inexact Newton methods using the two choices of forcing term, but without globalization, i.e., taking the full inexact Newton step.

Each problem had the following information collected and summarized in the tables found in the data section.

S/F: S indicates a solution was successfully found; F indicates a failure to find a solution.

INS: The total number of inexact Newton steps carried out. This can be important if each inexact Newton step has a high computation cost associated with it, for example a preconditioner that must be computed.

F-Evals: The number of function evaluations performed. If the cost of computing $F$-values is very high for a specified problem, then an algorithm which requires many function evaluations should probably not be chosen.

LS: The number of inexact Newton steps for which the algorithm needed to call a line search method.

f-LS: The number of line searches which failed to find a suitable step. In these cases the original inexact Newton step was used. The rational is that it is possible the algorithm is at a 'bad' $x_k$, such as one near a local minimum, and hopefully taking a large step in a not-so-great direction will get the algorithm to a better spot to advance from.

BckTcks: The total number of step reductions used by the line search methods.

GMRES: The total number of linear solver iterations. In the experiments here, the linear solve is the most expensive part of the algorithm; therefore reducing the number of GMRES iterations may improve efficiency, even if it increases the nnumber of inexact Newton steps.

$\|F(x)\|$: The final 2-norm of the residual vector. A goal is to find an $x_k$ such that $\|F(x_k)\| \leq 10^{-2}\|F(x_0)\|_2$. Our algorithm also imposed a constraint of the form WRMS$\leq 1$.

Time: The total number of seconds elapsed in reaching a solution.

An 'NA' in any column of data indicates the specified value is not computed. For example, the dogleg method will never indicate the number of line searches performed.

## 6.3 Results and Conclusions

### 6.3.1 A Robustness Study

We conducted a study involving the benchmark problems with the goal of assessing the general robustness of the seven different global inexact Newton methods: cubic interpolation with adaptive forcing terms, cubic interpolation with constant forcing terms, quadratic interpolation with adaptive forcing terms, quadratic interpolation with constant forcing terms, Moré-Thuente with adaptive forcing terms, Moré-Thuente with constant forcing terms, and the dogleg method with constant forcing terms. All of the problems were also run without a globalization method, i.e., taking the full inexact Newton step. The full-step method was used with both adaptive and constant forcing terms.

The study compared the number of failures of each method on problems of varying degrees of difficulty. We wanted to determine whether any one method is consistently better at solving these problems than other methods. The difficulty of the problems were determined by the Reynolds and Rayleigh numbers (as appropriate) as follows:

| | *Easy* | *Hard* |
|---|---|---|
| $2D - Thermal\ Convection$ | $10^3,\ 10^4,\ 10^5$ | $10^6$ |
| $2D - Backward\ Facing\ Step$ | $100 - 500$ | $600, 700, 750, 800$ |
| $2D - Lid\ Driven\ Cavity$ | $1000 - 5000$ | $6000 - 10,000$ |
| $3D - Thermal\ Convection$ | $10^3,\ 10^4,\ 10^5$ | $10^6$ |
| $3D - Lid\ Driven\ Cavity$ | $100 - 500$ | $600 - 1,000$ |

The results of the study are shown in Tables 6.1 and 6.2. We found that the globalized methods are more robust than the non-globalized methods. The use of the adaptive forcing terms greatly increases the number of successful solves on some problems, and overall is not worse than the constant forcing term methods. Limiting our comparisons to adaptive forcing terms (or constant forcing terms) shows no significant differences

among the globalized methods; cubic interpolation, quadratic interpolation, Moré-Thuente, and the dogleg all perform about the same. A final note, no single method works on all of the problems all of the time. The complete results from all of the runs are listed in the appendix.

| Method | 2D Thermal Convection | | 2D Lid Driven Cavity | | 2D Backward Facing Step | |
|---|---|---|---|---|---|---|
| | Easier | Harder | Easier | Harder | Easier | Harder |
| Cubic | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 4 | 5 | 0 | 0 |
| Quadratic | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 4 | 5 | 0 | 1 |
| Moré-Thuente | 0 | 1 | 0 | 0 | 0 | 1 |
| | 0 | 1 | 4 | 5 | 0 | 0 |
| Trust Region | 0 | 0 | 2 | 5 | 1 | 0 |
| Full Step | 0 | 0 | 4 | 5 | 1 | 4 |
| | 0 | 1 | 5 | 5 | 3 | 4 |

Table 6.1: Distribution of failures: For each method, the upper and lower lines are the number of failures with the adaptive forcing term and the constant forcing term, respectively.

| Method | 3D Thermal Convection | | 3D Lid Driven Cavity | |
|---|---|---|---|---|
| | Easier | Harder | Easier | Harder |
| Cubic | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |
| Quadratic | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |
| Moré-Thuente | 0 | 1 | 0 | 0 |
| | 1 | 1 | 0 | 0 |
| Trust Region | 0 | 0 | 0 | 0 |
| Full Step | 0 | 1 | 0 | 0 |
| | 0 | 1 | 0 | 4 |

Table 6.2: Distribution of failures: For each method, the upper and lower lines are the number of failures with the adaptive forcing term and the constant forcing term, respectively.

### 6.3.2 An Efficiency Study

We follow the robustness study above with a study aimed at assessing the relative efficiency of the global methods. Because of numerous failures throughout the benchmark problems, a subset of the problems for which all globalized methods succeeded was chosen. This set includes Reynolds and Rayleigh numbers as follows:

$$\begin{aligned} 2D - \textit{Thermal Convection}: &\quad 10^3,\ 10^4,\ 10^5 \\ 2D - \textit{Backward Facing Step} &\quad 100 - 300, 500, 600 \\ 3D - \textit{Lid Driven Cavity} &\quad 100 - 1,000 \end{aligned}$$

From each run, the study looked at the mean numbers of inexact Newton steps, backtracks taken by the line searches, GMRES iterations, and run times (in seconds). All are geometric means except in the case of backtracks, in which they are arithmetic

means.

From Table 6.3 we see a clear tradeoff between the number of inexact Newton iterations and the linear solver iterations. The methods using an adaptive forcing term require many fewer GMRES iterations than do the methods using a small constant forcing term. Conversely, the constant forcing term methods require many fewer inexact Newton steps. Overall, the adaptive forcing term methods use less time and more backtracking iterations than do the constant forcing term methods. As in the robustness study, the differences between the methods within the subsets of adaptive or constant forcing term methods is almost negligible.

| | Inexact Newton Steps | Backtracks | GMRES | Time |
|---|---|---|---|---|
| Cubic | 15 | .9188 | 704 | 112 |
| | 7 | .1627 | 1072 | 137 |
| Quadratic | 15 | 1.1183 | 714 | 113 |
| | 7 | .1627 | 1072 | 136 |
| Moré-Thuente | 14 | 1.2166 | 768 | 122 |
| | 8 | 1.7333 | 1123 | 145 |
| Trust Region | 9 | NA | 1264 | 161 |

Table 6.3: Efficiency Study: For each method, the upper and lower lines represent the adaptive forcing term and the constant forcing term methods, respectively.

# Appendix

**Backward Facing Step 2D**

Cubic Choice1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 10 | 11 | 0 | 0 | 0 | 459 | $4.44e-14$ | 18.730446 |
| 200 | $S$ | 12 | 14 | 1 | 0 | 1 | 535 | $1.59e-13$ | 22.386361 |
| 300 | $S$ | 14 | 17 | 2 | 0 | 2 | 650 | $1.62e-17$ | 28.494028 |
| 400 | $S$ | 51 | 77 | 25 | 0 | 25 | 1348 | $2.17e-15$ | 58.985995 |
| 500 | $S$ | 60 | 90 | 29 | 0 | 29 | 1573 | $1.93e-12$ | 71.104727 |
| 600 | $S$ | 81 | 126 | 43 | 0 | 44 | 2459 | $9.91e-14$ | 102.899106 |
| 700 | $S$ | 124 | 207 | 82 | 0 | 82 | 4635 | $8.00e-16$ | 193.631929 |
| 750 | $F$ | 200 | 660 | 199 | 0 | 459 | 2379 | 0.000244 | 160.015110 |
| 800 | $S$ | 162 | 294 | 130 | 0 | 131 | 5905 | $1.33e-13$ | 238.542687 |

Cubic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | 0 | 0 | 0 | 595 | $1.09e-14$ | 24.677943 |
| 200 | $S$ | 9 | 10 | 0 | 0 | 0 | 1047 | $1.25e-15$ | 49.499055 |
| 300 | $S$ | 9 | 11 | 1 | 0 | 1 | 1095 | $5.05e-16$ | 53.260950 |
| 400 | $S$ | 11 | 16 | 4 | 0 | 4 | 1441 | $1.08e-13$ | 67.335533 |
| 500 | $S$ | 9 | 12 | 2 | 0 | 2 | 1145 | $2.73e-15$ | 53.705681 |
| 600 | $S$ | 11 | 16 | 4 | 0 | 4 | 1506 | $7.14e-15$ | 78.006895 |
| 700 | $S$ | 12 | 18 | 5 | 0 | 5 | 1745 | $1.67e-13$ | 88.272582 |
| 750 | $S$ | 29 | 61 | 22 | 0 | 31 | 4729 | $1.40e-15$ | 263.265165 |
| 800 | $S$ | 31 | 66 | 25 | 0 | 34 | 5227 | $3.74e-13$ | 285.394827 |

Quadratic Choice1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 10 | 11 | 0 | 0 | 0 | 459 | $4.44e-14$ | 18.675301 |
| 200 | $S$ | 12 | 14 | 1 | 0 | 1 | 535 | $1.59e-13$ | 22.254953 |
| 300 | $S$ | 14 | 17 | 2 | 0 | 2 | 650 | $1.62e-17$ | 30.670698 |
| 400 | $S$ | 51 | 77 | 25 | 0 | 25 | 1348 | $2.17e-15$ | 59.451767 |
| 500 | $S$ | 60 | 90 | 29 | 0 | 29 | 1573 | $1.93e-12$ | 68.978642 |
| 600 | $S$ | 98 | 161 | 61 | 0 | 62 | 3193 | $1.39e-15$ | 131.410889 |
| 700 | $S$ | 124 | 207 | 82 | 0 | 82 | 4635 | $8.00e-16$ | 184.417626 |
| 750 | $S$ | 142 | 240 | 93 | 0 | 97 | 5116 | $4.76e-15$ | 204.551093 |
| 800 | $S$ | 130 | 218 | 85 | 0 | 87 | 4837 | $4.04e-11$ | 193.543793 |

Quadratic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | 0 | 0 | 0 | 595 | $1.09e-14$ | 24.503630 |
| 200 | $S$ | 9 | 10 | 0 | 0 | 0 | 1047 | $1.25e-15$ | 46.338407 |
| 300 | $S$ | 9 | 11 | 1 | 0 | 1 | 1095 | $5.05e-16$ | 49.873409 |
| 400 | $S$ | 11 | 16 | 4 | 0 | 4 | 1441 | $1.08e-13$ | 68.497390 |
| 500 | $S$ | 9 | 12 | 2 | 0 | 2 | 1145 | $2.73e-15$ | 54.265813 |
| 600 | $S$ | 11 | 16 | 4 | 0 | 4 | 1506 | $7.14e-15$ | 72.175325 |
| 700 | $S$ | 12 | 18 | 5 | 0 | 5 | 1745 | $1.67e-13$ | 87.335903 |
| 750 | $F$ | 200 | 936 | 198 | 0 | 735 | 38510 | $5.51e-05$ | 2283.374432 |
| 800 | $S$ | 44 | 104 | 37 | 0 | 59 | 7634 | $1.17e-16$ | 421.327834 |

More Thuente Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 10 | 31 | 0 | 0 | 0 | 459 | $4.44e-14$ | 21.500709 |
| 200 | $S$ | 11 | 36 | 1 | 0 | 1 | 581 | $2.96e-17$ | 27.400564 |
| 300 | $S$ | 18 | 67 | 6 | 0 | 6 | 721 | $3.82e-14$ | 32.270921 |
| 400 | $S$ | 38 | 151 | 18 | 0 | 18 | 1067 | $1.58e-13$ | 54.444312 |
| 500 | $S$ | 65 | 270 | 35 | 0 | 37 | 1526 | $1.07e-11$ | 77.272371 |
| 600 | $S$ | 73 | 328 | 44 | 0 | 54 | 2010 | $8.41e-11$ | 96.881934 |
| 700 | $F$ | 200 | 1449 | 186 | 152 | 348 | 7662 | $7.03e-06$ | 348.985394 |
| 750 | $S$ | 108 | 468 | 69 | 1 | 71 | 4182 | $1.44e-13$ | 186.352307 |
| 800 | $S$ | 105 | 449 | 64 | 1 | 66 | 4100 | $1.08e-15$ | 180.836193 |

More Thuente Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 19 | 0 | 0 | 0 | 595 | $1.09e-14$ | 24.483775 |
| 200 | $S$ | 7 | 24 | 1 | 0 | 1 | 754 | $1.16e-13$ | 31.984996 |
| 300 | $S$ | 9 | 30 | 1 | 0 | 1 | 1090 | $1.08e-13$ | 48.452983 |
| 400 | $S$ | 8 | 29 | 2 | 0 | 2 | 981 | $8.02e-14$ | 44.111969 |
| 500 | $S$ | 9 | 32 | 2 | 0 | 2 | 1158 | $4.49e-16$ | 53.506665 |
| 600 | $S$ | 10 | 37 | 3 | 0 | 3 | 1362 | $1.72e-15$ | 64.784416 |
| 700 | $S$ | 11 | 42 | 4 | 0 | 4 | 1593 | $4.13e-15$ | 78.274686 |
| 750 | $S$ | 11 | 42 | 4 | 0 | 4 | 1584 | $7.85e-15$ | 77.421196 |
| 800 | $S$ | 12 | 51 | 6 | 0 | 7 | 1796 | $5.58e-14$ | 89.863523 |

Trust Region

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | $NA$ | $NA$ | $NA$ | 595 | $1.09e-14$ | 26.582939 |
| 200 | $S$ | 9 | 10 | $NA$ | $NA$ | $NA$ | 1047 | $1.25e-15$ | 46.818451 |
| 300 | $S$ | 17 | 25 | $NA$ | $NA$ | $NA$ | 2154 | $2.91e-14$ | 101.102956 |
| 400 | $F$ | 200 | 248 | $NA$ | $NA$ | $NA$ | 30478 | $8.80e-05$ | 1559.254081 |
| 500 | $S$ | 11 | 13 | $NA$ | $NA$ | $NA$ | 1447 | $1.36e-15$ | 69.123568 |
| 600 | $S$ | 11 | 13 | $NA$ | $NA$ | $NA$ | 1552 | $5.89e-15$ | 76.694969 |
| 700 | $S$ | 17 | 24 | $NA$ | $NA$ | $NA$ | 2564 | $1.15e-16$ | 131.410720 |
| 750 | $S$ | 17 | 23 | $NA$ | $NA$ | $NA$ | 2634 | $2.29e-13$ | 138.535062 |
| 800 | $S$ | 18 | 28 | $NA$ | $NA$ | $NA$ | 2734 | $2.33e-13$ | 141.542069 |

Full Step Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 10 | 11 | $NA$ | $NA$ | $NA$ | 459 | $4.44e-14$ | 20.083346 |
| 200 | $S$ | 14 | 15 | $NA$ | $NA$ | $NA$ | 561 | $8.12e-17$ | 24.361944 |
| 300 | $S$ | 19 | 20 | $NA$ | $NA$ | $NA$ | 616 | $6.46e-16$ | 29.328713 |
| 400 | $S$ | 38 | 39 | $NA$ | $NA$ | $NA$ | 822 | $1.16e-13$ | 40.786583 |
| 500 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 17158 | 0.658 | 856.973023 |
| 600 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 12391 | 86.1 | 610.909392 |
| 700 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 21379 | 2.08 | 1096.242105 |
| 750 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 13867 | 5.39 | 653.159951 |
| 800 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 14333 | 1.17 | 705.480811 |

Full Step Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | $NA$ | $NA$ | $NA$ | 595 | $1.09e-14$ | 26.587996 |
| 200 | $S$ | 9 | 10 | $NA$ | $NA$ | $NA$ | 1047 | $1.25e-15$ | 45.992849 |
| 300 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 33846 | 9.27 | 1840.297428 |
| 400 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 25806 | 84.5 | 1344.981927 |
| 500 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 36936 | 0.533 | 2031.390187 |
| 600 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 28595 | 1.95 | 1527.294388 |
| 700 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 34524 | 5.88 | 1844.124334 |
| 750 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 57827 | 0.221 | 3303.303649 |
| 800 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 2583 | 0.126 | 166.383267 |

**Thermal Convection 2D**

Cubic Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 6 | 7 | 0 | 0 | 0 | 1106 | $2.83e-15$ | 85.620622 |
| $10^4$ | $S$ | 9 | 10 | 0 | 0 | 0 | 1209 | $1.57e-14$ | 87.355162 |
| $10^5$ | $S$ | 14 | 16 | 1 | 0 | 1 | 801 | $2.42e-09$ | 57.508363 |
| $10^6$ | $S$ | 41 | 65 | 19 | 0 | 23 | 2259 | $3.21e-10$ | 160.858089 |

Cubic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 4 | 5 | 0 | 0 | 0 | 870 | $3.28e-12$ | 64.423048 |
| $10^4$ | $S$ | 6 | 7 | 0 | 0 | 0 | 1349 | $6.80e-11$ | 97.672187 |
| $10^5$ | $S$ | 8 | 10 | 1 | 0 | 1 | 2106 | $6.21e-14$ | 154.707285 |
| $10^6$ | $S$ | 11 | 24 | 6 | 0 | 12 | 3353 | $7.44e-12$ | 247.764391 |

Quadratic Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 6 | 7 | 0 | 0 | 0 | 1106 | $2.83e-15$ | 80.400617 |
| $10^4$ | $S$ | 9 | 10 | 0 | 0 | 0 | 1209 | $1.57e-14$ | 87.218471 |
| $10^5$ | $S$ | 14 | 16 | 1 | 0 | 1 | 801 | $2.42e-09$ | 57.538396 |
| $10^6$ | $S$ | 44 | 68 | 20 | 0 | 23 | 2595 | $1.55e-10$ | 183.936025 |

Quadratic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 4 | 5 | 0 | 0 | 0 | 870 | $3.28e-12$ | 65.005171 |
| $10^4$ | $S$ | 6 | 7 | 0 | 0 | 0 | 1349 | $6.8e-11$ | 98.042959 |
| $10^5$ | $S$ | 8 | 10 | 1 | 0 | 1 | 2106 | $6.21e-14$ | 152.987253 |
| $10^6$ | $S$ | 14 | 30 | 9 | 0 | 15 | 4228 | $4.15e-13$ | 306.222720 |

More Thuente Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 6 | 19 | 0 | 0 | 0 | 1106 | $2.83e-15$ | 83.486171 |
| $10^4$ | $S$ | 9 | 28 | 0 | 0 | 0 | 1209 | $1.57e-14$ | 90.595408 |
| $10^5$ | $S$ | 14 | 50 | 2 | 1 | 3 | 1154 | $1.6e-09$ | 89.942351 |
| $10^6$ | $F$ | 50 | 401 | 50 | 50 | 100 | 2878 | 0.0498 | 218.216901 |

More Thuente Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 4 | 13 | 0 | 0 | 0 | 870 | $3.28e-12$ | 65.620265 |
| $10^4$ | $S$ | 6 | 19 | 0 | 0 | 0 | 1349 | $6.8e-11$ | 101.083927 |
| $10^5$ | $S$ | 50 | 367 | 44 | 42 | 87 | 12976 | $1.41e-13$ | 994.836491 |
| $10^6$ | $F$ | 50 | 401 | 50 | 50 | 100 | 18194 | 0.0495 | 1360.247254 |

Trust Region

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 4 | 5 | $NA$ | $NA$ | $NA$ | 870 | $3.28e-12$ | 64.922235 |
| $10^4$ | $S$ | 7 | 9 | $NA$ | $NA$ | $NA$ | 1944 | $3.89e-15$ | 146.254160 |
| $10^5$ | $S$ | 11 | 15 | $NA$ | $NA$ | $NA$ | 3413 | $4.41e-12$ | 250.861188 |
| $10^6$ | $S$ | 36 | 57 | $NA$ | $NA$ | $NA$ | 12756 | $5.33e-13$ | 935.867827 |

Full Step Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 6 | 7 | 0 | 0 | 0 | 1106 | $2.83e-15$ | 82.157177 |
| $10^4$ | $S$ | 9 | 10 | 0 | 0 | 0 | 1209 | $1.57e-14$ | 88.578725 |
| $10^5$ | $S$ | 14 | 16 | 1 | 0 | 1 | 801 | $2.42e-09$ | 59.041938 |
| $10^6$ | $S$ | 41 | 65 | 19 | 0 | 23 | 2259 | $3.21e-10$ | 159.174970 |

Full Step Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 4 | 5 | $NA$ | $NA$ | $NA$ | 870 | $3.28e-12$ | 67.205929 |
| $10^4$ | $S$ | 6 | 7 | $NA$ | $NA$ | $NA$ | 1349 | $6.80e-11$ | 99.858243 |
| $10^5$ | $S$ | 11 | 12 | $NA$ | $NA$ | $NA$ | 2665 | $9.98e-11$ | 198.862105 |
| $10^6$ | $F$ | 50 | 51 | $NA$ | $Na$ | $NA$ | 8506 | $9.71e+04$ | 624.502921 |

**Lid Driven Cavity 2D**

Cubic Choice1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 24 | 27 | 2 | 0 | 2 | 799 | $6.77e-11$ | 45.544701 |
| 2000 | $S$ | 33 | 42 | 8 | 0 | 8 | 1632 | $4.61e-13$ | 84.856306 |
| 3000 | $S$ | 52 | 66 | 13 | 0 | 13 | 2138 | $8.31e-13$ | 111.914418 |
| 4000 | $S$ | 57 | 73 | 14 | 0 | 15 | 2230 | $1.74e-07$ | 112.602015 |
| 5000 | $S$ | 58 | 79 | 18 | 0 | 20 | 2808 | $6.23e-10$ | 152.400856 |
| 6000 | $S$ | 75 | 106 | 28 | 0 | 30 | 3471 | $3.45e-10$ | 176.270308 |
| 7000 | $S$ | 94 | 148 | 50 | 0 | 53 | 4984 | $6.47e-12$ | 249.669885 |
| 8000 | $S$ | 106 | 165 | 55 | 0 | 58 | 5535 | $6.47e-11$ | 299.044550 |
| 9000 | $S$ | 150 | 260 | 106 | 0 | 109 | 7695 | $5.73e-12$ | 394.156317 |
| 10000 | $S$ | 160 | 279 | 115 | 0 | 118 | 8581 | $6.89e-06$ | 450.478600 |

Cubic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 12 | 18 | 5 | 0 | 5 | 1621 | $1.79e-13$ | 91.037015 |
| 2000 | $F$ | 300 | 1861 | 246 | 22 | 1559 | 77599 | $5.51e+14$ | 5405.766932 |
| 3000 | $F$ | 300 | 1851 | 287 | 15 | 1550 | 127980 | $3.77e+13$ | 8845.640383 |
| 4000 | $F$ | 300 | 3521 | 293 | 65 | 3220 | 77090 | $2.02e+14$ | 5693.023875 |
| 5000 | $F$ | 300 | 4900 | 298 | 0 | 4599 | 170847 | 88 | 12536.333741 |
| 6000 | $F$ | 300 | 2624 | 299 | 0 | 2323 | 175767 | 294 | 12176.149498 |
| 7000 | $F$ | 300 | 2676 | 298 | 0 | 2375 | 172883 | 146 | 11955.534328 |
| 8000 | $F$ | 300 | 4273 | 299 | 0 | 3972 | 172634 | 196 | 12082.865161 |
| 9000 | $F$ | 300 | 2129 | 299 | 0 | 1828 | 177490 | 480 | 12324.763266 |
| 10000 | $F$ | 300 | 2158 | 299 | 0 | 1857 | 174191 | 499 | 12652.027231 |

Quadratic Choice1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 24 | 27 | 2 | 0 | 2 | 799 | $6.77e-11$ | 42.405545 |
| 2000 | $S$ | 33 | 42 | 8 | 0 | 8 | 1632 | $4.61e-13$ | 84.774296 |
| 3000 | $S$ | 52 | 66 | 13 | 0 | 13 | 2138 | $8.31e-13$ | 120.885515 |
| 4000 | $S$ | 52 | 66 | 12 | 0 | 13 | 2611 | $3.44e-12$ | 139.664326 |
| 5000 | $S$ | 60 | 78 | 16 | 0 | 17 | 2722 | $4.55e-06$ | 140.442642 |
| 6000 | $S$ | 76 | 110 | 32 | 0 | 33 | 3303 | $1.62e-09$ | 180.238783 |
| 7000 | $S$ | 101 | 164 | 61 | 0 | 62 | 5119 | $2.73e-09$ | 262.796143 |
| 8000 | $S$ | 140 | 248 | 106 | 0 | 107 | 6881 | $4.98e-09$ | 346.188046 |
| 9000 | $S$ | 143 | 242 | 97 | 0 | 98 | 7833 | $4.54e-10$ | 394.727409 |
| 10000 | $S$ | 163 | 284 | 119 | 0 | 120 | 8950 | $2.34e-08$ | 467.777669 |

Quadratic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 12 | 18 | 5 | 0 | 5 | 1621 | $1.79e-13$ | 95.523814 |
| 2000 | $F$ | 300 | 1738 | 298 | 0 | 1437 | 53931 | 15.6 | 3627.953197 |
| 3000 | $F$ | 300 | 1746 | 298 | 0 | 1445 | 110022 | 63.3 | 7389.694315 |
| 4000 | $F$ | 300 | 1712 | 298 | 0 | 1411 | 85748 | 42.8 | 5504.815610 |
| 5000 | $F$ | 300 | 1693 | 298 | 0 | 1392 | 170958 | 147 | 11732.354169 |
| 6000 | $F$ | 300 | 1979 | 299 | 0 | 1678 | 174141 | 520 | 11999.624440 |
| 7000 | $F$ | 300 | 1579 | 299 | 0 | 1278 | 174954 | 284 | 12649.365681 |
| 8000 | $F$ | 300 | 1605 | 299 | 0 | 1304 | 119613 | 244 | 8437.113389 |
| 9000 | $F$ | 300 | 1732 | 299 | 0 | 1431 | 176946 | 997 | 12875.594389 |
| 10000 | $F$ | 300 | 1878 | 299 | 0 | 1577 | 169707 | $2.39e+03$ | 11721.137505 |

More Thuente Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 18 | 63 | 4 | 0 | 4 | 923 | $1.72e-10$ | 55.868397 |
| 2000 | $S$ | 31 | 112 | 9 | 0 | 9 | 1341 | $8.87e-07$ | 76.574034 |
| 3000 | $S$ | 42 | 156 | 13 | 1 | 14 | 1863 | $4.96e-09$ | 107.952866 |
| 4000 | $S$ | 51 | 195 | 19 | 1 | 20 | 2377 | $2.27e-06$ | 135.197965 |
| 5000 | $S$ | 59 | 221 | 20 | 1 | 21 | 2939 | $2.01e-12$ | 169.963651 |
| 6000 | $S$ | 66 | 252 | 25 | 1 | 26 | 3374 | $1.47e-09$ | 194.691589 |
| 7000 | $S$ | 79 | 327 | 42 | 1 | 44 | 3609 | $2.04e-06$ | 205.061383 |
| 8000 | $S$ | 92 | 391 | 53 | 2 | 56 | 4717 | $1.08e-11$ | 271.396304 |
| 9000 | $S$ | 110 | 477 | 69 | 2 | 72 | 6149 | $5.96e-12$ | 359.284430 |
| 10000 | $S$ | 124 | 543 | 81 | 2 | 84 | 6550 | $2.99e-07$ | 371.076367 |

More Thuente Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 11 | 40 | 3 | 0 | 3 | 1475 | $1.13e-13$ | 87.467960 |
| 2000 | $F$ | 300 | 2376 | 298 | 293 | 591 | 50430 | 13 | 3447.789024 |
| 3000 | $F$ | 300 | 2368 | 298 | 289 | 589 | 76340 | 86.5 | 5374.839047 |
| 4000 | $F$ | 300 | 2384 | 299 | 295 | 594 | 61886 | 152 | 4527.014148 |
| 5000 | $F$ | 300 | 2379 | 299 | 292 | 593 | 77752 | 340 | 5479.676001 |
| 6000 | $F$ | 300 | 2375 | 299 | 290 | 592 | 110848 | 176 | 7764.665417 |
| 7000 | $F$ | 300 | 2390 | 299 | 297 | 596 | 110599 | 981 | 7655.571218 |
| 8000 | $F$ | 300 | 2392 | 299 | 297 | 597 | 110839 | $1.38e+03$ | 7665.834013 |
| 9000 | $F$ | 300 | 2392 | 299 | 297 | 597 | 124646 | $1.75e+03$ | 8820.438071 |
| 10000 | $F$ | 300 | 2373 | 300 | 286 | 593 | 178988 | $1.23e+03$ | 12835.061594 |

Trust Region

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 19 | 27 | $NA$ | $NA$ | $NA$ | 2671 | $1.73e-12$ | 155.229284 |
| 2000 | $S$ | 29 | 42 | $NA$ | $NA$ | $NA$ | 4325 | $5.24e-12$ | 257.387507 |
| 3000 | $S$ | 30 | 42 | $NA$ | $NA$ | $NA$ | 4552 | $8.33e-13$ | 280.585165 |
| 4000 | $F$ | 300 | 371 | $NA$ | $NA$ | $NA$ | 51585 | 37.4 | 3451.221284 |
| 5000 | $F$ | 300 | 448 | $NA$ | $NA$ | $NA$ | 60715 | 49.3 | 4100.548967 |
| 6000 | $F$ | 300 | 408 | $NA$ | $NA$ | $NA$ | 59459 | 60.8 | 4207.405368 |
| 7000 | $F$ | 300 | 346 | $NA$ | $NA$ | $NA$ | 85863 | 59.1 | 5844.815039 |
| 8000 | $F$ | 300 | 331 | $NA$ | $NA$ | $NA$ | 56610 | 70.2 | 5137.866875 |
| 9000 | $F$ | 300 | 372 | $NA$ | $NA$ | $NA$ | 57720 | 73.2 | 3866.840246 |
| 10000 | $F$ | 300 | 467 | $NA$ | $NA$ | $NA$ | 162693 | 71.6 | 11615.001159 |

Full Step Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $S$ | 20 | 21 | $NA$ | $NA$ | $NA$ | 787 | $2.02e-13$ | 45.827322 |
| 2000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 9454 | $1.01e+05$ | 649.749201 |
| 3000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 30362 | $5.62e+04$ | 1933.428328 |
| 4000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 11713 | $2.03e+05$ | 762.675952 |
| 5000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 13937 | $4.18e+05$ | 869.490376 |
| 6000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 11285 | $2.60e+05$ | 735.437533 |
| 7000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 17791 | $3.31e+05$ | 1089.994508 |
| 8000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 12830 | $3.15e+05$ | 787.788003 |
| 9000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 12399 | $4.16e+05$ | 751.368204 |
| 10000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 32499 | $1.76e+05$ | 2070.265778 |

Full Step Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 23416 | $3.83e+05$ | 1499.734888 |
| 2000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 10919 | $2.86e+05$ | 717.580405 |
| 3000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 14610 | $2.48e+05$ | 937.857191 |
| 4000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 12255 | $1.12e+05$ | 817.903355 |
| 5000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 16246 | $8.40e+04$ | 1085.189837 |
| 6000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 25644 | $8.88e+04$ | 1666.684245 |
| 7000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 29688 | $2.30e+05$ | 1936.334228 |
| 8000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 14166 | $4.88e+06$ | 958.168039 |
| 9000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 13504 | $1.06e+06$ | 855.090293 |
| 10000 | $F$ | 300 | 301 | $NA$ | $NA$ | $NA$ | 18901 | $1.64e+05$ | 1217.999328 |

**Thermal Convection 3D**

Cubic Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | 0 | 0 | 0 | 298 | $2.22e-12$ | 161.661027 |
| $10^4$ | $S$ | 8 | 10 | 1 | 0 | 1 | 469 | $1.92e-15$ | 257.054125 |
| $10^5$ | $S$ | 19 | 25 | 3 | 0 | 5 | 584 | $5.1e-15$ | 478.700628 |
| $10^6$ | $S$ | 58 | 97 | 27 | 0 | 38 | 3291 | $6.32e-14$ | 1818.225104 |

Cubic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | 0 | 0 | 0 | 504 | $1.23e-16$ | 214.912963 |
| $10^4$ | $S$ | 6 | 8 | 1 | 0 | 1 | 697 | $1.09e-15$ | 274.601650 |
| $10^5$ | $S$ | 10 | 17 | 4 | 0 | 6 | 1120 | $5.66e-15$ | 453.091963 |
| $10^6$ | $S$ | 20 | 57 | 14 | 0 | 36 | 2377 | $4.43e-14$ | 955.422234 |

Quadratic Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | 0 | 0 | 0 | 298 | $2.22e-12$ | 161.779187 |
| $10^4$ | $S$ | 8 | 10 | 1 | 0 | 1 | 469 | $1.92e-15$ | 252.332040 |
| $10^5$ | $S$ | 18 | 24 | 4 | 0 | 5 | 640 | $6.66e-15$ | 478.259933 |
| $10^6$ | $S$ | 58 | 93 | 28 | 0 | 34 | 3153 | $5.57e-14$ | 1779.562241 |

Quadratic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | 0 | 0 | 0 | 504 | $1.23e-16$ | 212.000412 |
| $10^4$ | $S$ | 6 | 8 | 1 | 0 | 1 | 697 | $1.09e-15$ | 278.757287 |
| $10^5$ | $S$ | 11 | 19 | 5 | 0 | 7 | 1248 | $7.57e-15$ | 499.187919 |
| $10^6$ | $S$ | 26 | 64 | 20 | 0 | 37 | 3156 | $4.32e-14$ | 1259.053510 |

More Thuente Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 16 | 0 | 0 | 0 | 298 | $2.22e-12$ | 167.026084 |
| $10^4$ | $S$ | 9 | 30 | 1 | 0 | 1 | 636 | $6.61e-16$ | 317.815613 |
| $10^5$ | $S$ | 21 | 79 | 6 | 1 | 7 | 939 | $8.18e-15$ | 620.417555 |
| $10^6$ | $F$ | 200 | 1601 | 200 | 200 | 400 | 6074 | 0.0775 | 5430.274450 |

More Thuente Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 16 | 0 | 0 | 0 | 504 | $1.23e-16$ | 212.738866 |
| $10^4$ | $S$ | 7 | 24 | 1 | 0 | 1 | 813 | $7.33e-16$ | 325.670966 |
| $10^5$ | $F$ | 200 | 1601 | 200 | 200 | 400 | 24243 | 0.0763 | 10001.531665 |
| $10^6$ | $F$ | 200 | 1601 | 200 | 200 | 400 | 26563 | 0.0763 | 10679.498846 |

Trust Region

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | $NA$ | $NA$ | $NA$ | 504 | $1.23e-16$ | 214.790449 |
| $10^4$ | $S$ | 8 | 9 | $NA$ | $NA$ | $NA$ | 871 | $6.8e-16$ | 353.840988 |
| $10^5$ | $S$ | 11 | 14 | $NA$ | $NA$ | $NA$ | 1334 | $8.19e-15$ | 528.589408 |
| $10^6$ | $S$ | 15 | 22 | $NA$ | $NA$ | $NA$ | 1832 | $5.89e-14$ | 718.635854 |

Full Step Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | $NA$ | $NA$ | $NA$ | 298 | $2.22e-12$ | 157.906350 |
| $10^4$ | $S$ | 12 | 13 | $NA$ | $NA$ | $NA$ | 446 | $1.13e-15$ | 319.371344 |
| $10^5$ | $S$ | 20 | 21 | $NA$ | $NA$ | $NA$ | 585 | $7.82e-15$ | 501.202213 |
| $10^6$ | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 1842 | $1.31e+104$ | 4086.060439 |

Full Step Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| $10^3$ | $S$ | 5 | 6 | $NA$ | $NA$ | $NA$ | 504 | $1.23e-16$ | 208.590341 |
| $10^4$ | $S$ | 8 | 9 | $NA$ | $NA$ | $NA$ | 871 | $6.8e-16$ | 350.150091 |
| $10^5$ | $S$ | 12 | 13 | $NA$ | $NA$ | $NA$ | 1236 | $8.1e-15$ | 515.169594 |
| $10^6$ | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 4561 | $1.37e+65$ | 4792.743731 |

**Lid Driven Cavity 3D**

Cubic Choice1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 7 | 8 | 0 | 0 | 0 | 266 | $2.63e-14$ | 104.692000 |
| 200 | $S$ | 12 | 13 | 0 | 0 | 0 | 393 | $7.27e-15$ | 169.832319 |
| 300 | $S$ | 10 | 11 | 0 | 0 | 0 | 455 | $1.23e-14$ | 160.762959 |
| 400 | $S$ | 13 | 14 | 0 | 0 | 0 | 495 | $1.23e-14$ | 195.418103 |
| 500 | $S$ | 13 | 14 | 0 | 0 | 0 | 515 | $1.53e-14$ | 206.223205 |
| 600 | $S$ | 16 | 17 | 0 | 0 | 0 | 413 | $1.33e-11$ | 206.086810 |
| 700 | $S$ | 17 | 20 | 2 | 0 | 2 | 819 | $8.14e-13$ | 283.679220 |
| 800 | $S$ | 17 | 19 | 1 | 0 | 1 | 720 | $3.97e-14$ | 272.367693 |
| 900 | $S$ | 22 | 27 | 4 | 0 | 4 | 1253 | $3.68e-14$ | 401.430639 |
| 1000 | $S$ | 24 | 28 | 3 | 0 | 3 | 756 | $5.06e-14$ | 340.594764 |

Cubic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | 0 | 0 | 0 | 495 | $2.59e-15$ | 131.654441 |
| 200 | $S$ | 7 | 8 | 0 | 0 | 0 | 625 | $7.12e-15$ | 164.200106 |
| 300 | $S$ | 8 | 9 | 0 | 0 | 0 | 751 | $1.10e-14$ | 194.042359 |
| 400 | $S$ | 9 | 10 | 0 | 0 | 0 | 864 | $1.17e-14$ | 224.545563 |
| 500 | $S$ | 9 | 10 | 0 | 0 | 0 | 927 | $2.87e-14$ | 235.666279 |
| 600 | $S$ | 11 | 12 | 0 | 0 | 0 | 1151 | $3.36e-14$ | 290.587334 |
| 700 | $S$ | 10 | 12 | 1 | 0 | 1 | 1176 | $4.03e-14$ | 292.697456 |
| 800 | $S$ | 11 | 13 | 1 | 0 | 1 | 1365 | $5.53e-14$ | 340.679790 |
| 900 | $S$ | 13 | 16 | 2 | 0 | 2 | 1648 | $5.67e-14$ | 404.259180 |
| 1000 | $S$ | 17 | 26 | 8 | 0 | 8 | 2160 | $5.07e-14$ | 529.605405 |

Quadratic Choice1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 7 | 8 | 0 | 0 | 0 | 266 | $2.63e-14$ | 105.548778 |
| 200 | $S$ | 12 | 13 | 0 | 0 | 0 | 393 | $7.27e-15$ | 168.653800 |
| 300 | $S$ | 10 | 11 | 0 | 0 | 0 | 455 | $1.23e-14$ | 162.668037 |
| 400 | $S$ | 13 | 14 | 0 | 0 | 0 | 495 | $1.23e-14$ | 196.143194 |
| 500 | $S$ | 13 | 14 | 0 | 0 | 0 | 515 | $1.53e-14$ | 203.371747 |
| 600 | $S$ | 16 | 17 | 0 | 0 | 0 | 413 | $1.33e-11$ | 207.986475 |
| 700 | $S$ | 17 | 20 | 2 | 0 | 2 | 819 | $8.14e-13$ | 280.494864 |
| 800 | $S$ | 17 | 19 | 1 | 0 | 1 | 720 | $3.97e-14$ | 274.225519 |
| 900 | $S$ | 22 | 27 | 4 | 0 | 4 | 1253 | $3.68e-14$ | 401.193352 |
| 1000 | $S$ | 24 | 28 | 3 | 0 | 3 | 756 | $5.06e-14$ | 354.664516 |

Quadratic Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | 0 | 0 | 0 | 495 | $2.59e-15$ | 132.703209 |
| 200 | $S$ | 7 | 8 | 0 | 0 | 0 | 625 | $7.12e-15$ | 168.146856 |
| 300 | $S$ | 8 | 9 | 0 | 0 | 0 | 751 | $1.10e-14$ | 196.932237 |
| 400 | $S$ | 9 | 10 | 0 | 0 | 0 | 864 | $1.17e-14$ | 224.682512 |
| 500 | $S$ | 9 | 10 | 0 | 0 | 0 | 927 | $2.87e-14$ | 235.633007 |
| 600 | $S$ | 11 | 12 | 0 | 0 | 0 | 1151 | $3.36e-14$ | 290.200892 |
| 700 | $S$ | 10 | 12 | 1 | 0 | 1 | 1176 | $4.03e-14$ | 288.763588 |
| 800 | $S$ | 11 | 13 | 1 | 0 | 1 | 1365 | $5.53e-14$ | 338.343319 |
| 900 | $S$ | 13 | 16 | 2 | 0 | 2 | 1648 | $5.67e-14$ | 409.653673 |
| 1000 | $S$ | 17 | 26 | 8 | 0 | 8 | 2160 | $5.07e-14$ | 528.286561 |

More Thuente Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 7 | 22 | 0 | 0 | 0 | 266 | $2.63e-14$ | 106.383072 |
| 200 | $S$ | 12 | 37 | 0 | 0 | 0 | 393 | $7.27e-15$ | 178.085724 |
| 300 | $S$ | 10 | 31 | 0 | 0 | 0 | 455 | $1.23e-14$ | 165.572808 |
| 400 | $S$ | 13 | 40 | 0 | 0 | 0 | 495 | $1.23e-14$ | 203.166992 |
| 500 | $S$ | 13 | 40 | 0 | 0 | 0 | 515 | $1.53e-14$ | 206.944519 |
| 600 | $S$ | 18 | 59 | 2 | 0 | 2 | 850 | $1.84e-14$ | 307.803561 |
| 700 | $S$ | 18 | 61 | 3 | 0 | 3 | 946 | $5.17e-13$ | 315.963266 |
| 800 | $S$ | 19 | 64 | 3 | 0 | 3 | 1176 | $4.80e-14$ | 374.291635 |
| 900 | $S$ | 19 | 64 | 3 | 0 | 3 | 1046 | $6.42e-14$ | 351.060000 |
| 1000 | $S$ | 16 | 53 | 2 | 0 | 2 | 813 | $6.42e-14$ | 291.630151 |

More Thuente Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 19 | 0 | 0 | 0 | 495 | $2.59e-15$ | 141.351836 |
| 200 | $S$ | 7 | 22 | 0 | 0 | 0 | 625 | $7.12e-15$ | 166.785230 |
| 300 | $S$ | 8 | 25 | 0 | 0 | 0 | 751 | $1.10e-14$ | 201.169303 |
| 400 | $S$ | 9 | 28 | 0 | 0 | 0 | 864 | $1.17e-14$ | 227.805476 |
| 500 | $S$ | 9 | 28 | 0 | 0 | 0 | 927 | $2.87e-14$ | 238.384421 |
| 600 | $S$ | 11 | 36 | 1 | 0 | 1 | 1211 | $2.40e-14$ | 314.293763 |
| 700 | $S$ | 10 | 33 | 1 | 0 | 1 | 1170 | $4.55e-14$ | 292.697230 |
| 800 | $S$ | 11 | 36 | 1 | 0 | 1 | 1320 | $4.37e-14$ | 332.888019 |
| 800 | $S$ | 12 | 41 | 2 | 0 | 2 | 1484 | $3.66e-14$ | 370.552730 |
| 1000 | $S$ | 13 | 46 | 3 | 0 | 3 | 1814 | $4.81e-14$ | 450.592816 |

Trust Region

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 7 | 8 | $NA$ | $NA$ | $NA$ | 586 | $2.47e-15$ | 156.567477 |
| 200 | $S$ | 8 | 9 | $NA$ | $NA$ | $NA$ | 721 | $6.15e-15$ | 188.789107 |
| 300 | $S$ | 9 | 10 | $NA$ | $NA$ | $NA$ | 849 | $8.50e-15$ | 218.150629 |
| 400 | $S$ | 9 | 10 | $NA$ | $NA$ | $NA$ | 879 | $1.67e-14$ | 223.849692 |
| 500 | $S$ | 11 | 12 | $NA$ | $NA$ | $NA$ | 1072 | $1.54e-14$ | 275.023223 |
| 600 | $S$ | 10 | 11 | $NA$ | $NA$ | $NA$ | 1082 | $3.07e-14$ | 271.410170 |
| 700 | $S$ | 11 | 12 | $NA$ | $NA$ | $NA$ | 1254 | $3.09e-14$ | 312.430216 |
| 800 | $S$ | 12 | 13 | $NA$ | $NA$ | $NA$ | 1512 | $4.12e-14$ | 367.438938 |
| 900 | $S$ | 18 | 22 | $NA$ | $NA$ | $NA$ | 2777 | $3.71e-14$ | 658.829983 |
| 1000 | $S$ | 14 | 16 | $NA$ | $NA$ | $NA$ | 2049 | $4.61e-14$ | 498.664243 |

Full Step Choice 1

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 7 | 8 | $NA$ | $NA$ | $NA$ | 266 | $2.63e-14$ | 103.176433 |
| 200 | $S$ | 12 | 13 | $NA$ | $NA$ | $NA$ | 393 | $7.27e-15$ | 173.186368 |
| 300 | $S$ | 10 | 11 | $NA$ | $NA$ | $NA$ | 455 | $1.23e-14$ | 162.967126 |
| 400 | $S$ | 13 | 14 | $NA$ | $NA$ | $NA$ | 495 | $1.23e-14$ | 196.046878 |
| 500 | $S$ | 13 | 14 | $NA$ | $NA$ | $NA$ | 515 | $1.53e-14$ | 204.283561 |
| 600 | $S$ | 16 | 17 | $NA$ | $NA$ | $NA$ | 413 | $1.33e-11$ | 206.968673 |
| 700 | $S$ | 18 | 19 | $NA$ | $NA$ | $NA$ | 736 | $3.92e-14$ | 278.062298 |
| 800 | $S$ | 27 | 28 | $NA$ | $NA$ | $NA$ | 629 | $3.52e-14$ | 339.087976 |
| 900 | $S$ | 32 | 33 | $NA$ | $NA$ | $NA$ | 775 | $3.66e-14$ | 410.559843 |
| 1000 | $S$ | 35 | 36 | $NA$ | $NA$ | $NA$ | 819 | $5.69e-14$ | 446.344469 |

Full Step Constant

| | $S/F$ | $INS$ | $F-Evals$ | $LS$ | $f-LS$ | $BckTcks$ | $GMRES$ | $\|F(x)\|$ | $Time$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | $S$ | 6 | 7 | $NA$ | $NA$ | $NA$ | 495 | $2.59e-15$ | 131.730091 |
| 200 | $S$ | 7 | 8 | $NA$ | $NA$ | $NA$ | 625 | $7.12e-15$ | 162.208564 |
| 300 | $S$ | 8 | 9 | $NA$ | $NA$ | $NA$ | 751 | $1.10e-14$ | 192.167292 |
| 400 | $S$ | 9 | 10 | $NA$ | $NA$ | $NA$ | 864 | $1.17e-14$ | 227.151910 |
| 500 | $S$ | 9 | 10 | $NA$ | $NA$ | $NA$ | 927 | $2.87e-14$ | 236.867409 |
| 600 | $S$ | 11 | 12 | $NA$ | $NA$ | $NA$ | 1151 | $3.36e-14$ | 297.637799 |
| 700 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 924 | 116 | 1908.105526 |
| 800 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 1048 | 154 | 1924.440846 |
| 900 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 792 | $3.48e+09$ | 1884.846037 |
| 1000 | $F$ | 200 | 201 | $NA$ | $NA$ | $NA$ | 1021 | 771 | 1934.055469 |

# Bibliography


[1] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

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

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

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

[5] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[6] 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.

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

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

[9] M. Heroux. Trilinos project home page. http://software.sandia.gov/trilinos, Retrieved April 2003.

[10] P. Lin. Using the 806 IBM cluster. 2002.

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

[12] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] H. F. Walker. Numerical methods for nonlinear equations. Technical Report MS-03-02-18, Worcester Polytechnical Institute, Worcester MA, 01609, May 2002.