Solution of Nonlinear Least-Squares Proble

by

Christina Fraley

Department of Computer Science

Stanford University
Stanford, CA 94305

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-01
Exp. Date: Jun 30

)RT SECURITY CLASSIFICATION
unclassified

1b. RESTRICTIVE MARKINGS

)RITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release:
Distribution unlimited

.ASSIFICATION/DOWNGRADING SCHEDULE

)RMING ORGANIZATION REPORT NUMBER(S)
STAN-CS-87-1165

5. MONITORING ORGANIZATION REPORT NUMBER(S)

IE OF PERFORMING ORGANIZATION
Computer Science Dept.

6b. OFFICE SYMBOL (*If applicable*)

7a. NAME OF MONITORING ORGANIZATION

RESS (*City, State, and ZIP Code*)
Stanford University
Stanford, CA 94305

7b. ADDRESS (*City, State, and ZIP Code*)

IE OF FUNDING/SPONSORING ANIZATION
ONR

8b. OFFICE SYMBOL (*If applicable*)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-82-K-0335

RESS (*City, State, and ZIP Code*)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK U ACCESSI

E (*Include Security Classification*)
Solution of Nonlinear Least-Squares Problems

ONAL AUTHOR(S)
Christina Fraley

E OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
thesis	FROM _____ TO _____	1987 June	308

LEMENTARY NOTATION

COSATI CODES		
D	GROUP	SUB-GROUP

18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*

TRACT (*Continue on reverse if necessary and identify by block number*)

issertation addresses the nonlinear least-squares problem

$$\min_{x \in \mathbb{R}^n} \|f(x)\|_2^2,$$

is a vector in $\mathbb{R}^m$ whose components are smooth nonlinear functions. The problem arises in data fitting applications. Much research has focused on the development of specialized that attempt to exploit the structure of the nonlinear least-squares objective. We assume m are relatively small, so that limited storage and sparsity in the derivatives of f need not nto account in formulating algorithms.

rt discuss existing numerical algorithms for nonlinear least squares, nearly all of which involve inimization of quadratic functions. Methods for general unconstrained optimization, Gauss ethods, Levenberg-Marquardt methods, and special quasi-Newton methods are among the algorithms surveyed. Our emphasis is on those methods that fo the basis of widely-distributed software, and numerical results ar for a large set of test problems.

The main contribution of this research is to propose new algor make use of more general quadratic programming subproblems. O investigated that are based on convergence properties of sequential programming methods for constrained optimization, and on geom erations in nonlinear least squares. Numerical results are given, de that the new methods may be useful in practice.

TRIBUTION/AVAILABILITY OF ABSTRACT
NCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

ME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (*Include Area Code*)

22c OFFICE SYMBOL

RM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS P

SOLUTION
OF
NONLINEAR
LEAST-SQUARES PROBLEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

by
Christina Fraley
June 1987

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph Oliger
(Principal Adviser)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Walter Murray
(Operations Research)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Margaret H. Wright
(Operations Research)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Philip E. Gill
(Operations Research)

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Acknowledgements

I have many people to thank for helping to make my time at Stanford worthwhile. Joseph Oliger gave sound advice, and a vote of confidence throughout. Walter Murray, Margaret Wright, and Philip Gill suggested my thesis topic, and shared their insights and expertise. Many others contributed through their friendship and goodwill.

I am fortunate to have had generous financial support for this research in the form of a fellowship from the Xerox Corporation, with additional funding provided by Joseph Oliger under Office of Naval Research contract N00014–82–K–0335, as well as some summer support from Stanford Linear Accelerator Center and the Systems Optimization Laboratory under Army Research Office contract DAAG29–84–K–0156. I am also indebted to Stanford Linear Accelerator Center for the use of their computer facilities.

This manuscript was produced using TEX, the computer typesetting system developed by Donald Knuth at Stanford.

Solution of Nonlinear Least-Squares Problems

Christina Fraley, Ph. D.
Stanford University 1987

Abstract

This dissertation addresses the nonlinear least-squares problem

$$\min_{x \in \Re^n} \|f(x)\|_2^2,$$

where $f(x)$ is a vector in $\Re^m$ whose components are smooth nonlinear functions. The problem aris most often in data fitting applications. Much research has focused on the development of specialize algorithms that attempt to exploit the structure of the nonlinear least-squares objective. We assum that n and m are relatively small, so that limited storage and sparsity in the derivatives of f need n be taken into account in formulating algorithms.

We first discuss existing numerical algorithms for nonlinear least squares, nearly all of which invol iterative minimization of quadratic functions. Methods for general unconstrained optimization, Gaus Newton methods, Levenberg-Marquardt methods, and special quasi-Newton methods are among t algorithms surveyed. Our emphasis is on those methods that form the basis of widely-distribute software, and numerical results are given for a large set of test problems.

The main contribution of this research is to propose new algorithms that make use of more gener quadratic programming subproblems. Options are investigated that are based on convergence properti of sequential quadratic programming methods for constrained optimization, and on geometric consi erations in nonlinear least squares. Numerical results are given, demonstrating that the new metho may be useful in practice.

Table of Contents

Table of Contents

Table of Contents

List of Tables

List of Figures

1. Introduction

The dissertation addresses the problem of minimizing the l_2 norm of a multivariate function :

$$\min_{x \in \Re^n} \|f(x)\|_2^2, \qquad \text{NLSQ}$$

where $f(x)$ is a vector in $\Re^m$ whose components are real-valued nonlinear functions with continuous second partial derivatives. An alternative formulation of the problem is that of minimizing a sum of squares :

$$\min_{x \in \Re^n} \frac{1}{2} \sum_{i=1}^{m} \phi_i(x)^2,$$

where each ϕ_i is a real-valued function having continuous second partial derivatives. There is considerable interest in the nonlinear least-squares problem, because it arises in virtually all areas of quantitative research in data-fitting applications. A typical instance is the choice of parameters β within a nonlinear model φ so that the model agrees with measured quantities d_i as closely as possible :

$$\min_{\beta \in \Re^n} \sum_{i=1}^{m} (\varphi(\beta; \tau_i) - d_i)^2,$$

where τ_i are variables whose values are selected in advance. Much research has focused on the development of specialized algorithms that attempt to exploit the structure of the nonlinear least-squares objective. We assume that n and m are relatively small, so that limited storage and sparsity in the derivatives of f need not be taken into account in formulating algorithms.

In this dissertation, we first survey existing numerical methods for nonlinear least squares, and conduct extensive numerical tests using software that is widely available. Nearly all of the existing methods involve iterative minimization of quadratic functions. The main contribution of this research is to propose new algorithms that make use of more general quadratic programming subproblems. Options are investigated that are based on convergence properties of sequential quadratic programming methods for constrained optimization, and on geometric considerations in nonlinear least squares. Numerical results are given, demonstrating that the new methods may be useful in practice.

1.1 Overview

In the remainder of this introductory chapter, we summarize our definitions and notational conventions, as well as give general information as to how the numerical results were obtained, and how they are presented. Methods for general unconstrained optimization are reviewed in Chapter 2, because much of the motivation for these methods is relevant to algorithm development for the special case of nonlinear least squares. Numerical results are tabulated for some widely-distributed implementations of these unconstrained optimization methods applied to nonlinear least-squares problems. We expect that special-purpose algorithms for nonlinear least squares should compare favorably with the more general algorithms. Theoretical and computational aspects of linear least-squares problems are treated in Chapter 3. Linear least squares is an important and well-understood instance of NLSQ, and orthogonalization techniques related to those used to solve linear least-squares problems are applicable in many other situations in nonlinear programming, including quadratic programming, which plays a key role in the algorithms developed in this research. Chapter 4 is devoted to Gauss-Newton methods, the classical approach to nonlinear least squares, in which a linear least-squares problem is solved at every iteration. In some instances, Gauss-Newton methods are observed to perform very well, and in others, they perform very poorly. Most current algorithms are based to some extent on Gauss-Newton methods, in an attempt to exploit the good behavior, and overcome the drawbacks of the method. Examples that illustrate some of the difficulties involved are analyzed in detail, and numerical results are tabulated for two different implementations. Chapter 5 is a survey of existing numerical methods for nonlinear least squares, with emphasis on those for which software is readily available. As in Chapter 2, numerical results are presented for some widely-distributed implementations. A summary and discussion of the numerical results for Chapters 2, 4, and 5, is included at the end of the chapter. In Chapter 6, the final chapter, we motivate and describe the new sequential quadratic programming methods. Numerical results are presented and discussed, and we conclude with some suggestions for future work. Detailed information about the test problems is given in the Appendix.

1.2 Definitions and Notation

We shall use the following definitions and notational conventions :

- Generally subscripts on a function mean that the function is evaluated at the corresponding subscripted variable (for example, $f_k = f(x_k)$). An exception is made for the residual functions ϕ_i, where the subscript is the component index for the vector f.

- $^{\mathrm{T}}$ - As a superscript, $^{\mathrm{T}}$ denotes the transpose of a vector or matrix.
  If A is an $m \times n$ matrix, then A^{T} is the $n \times m$ matrix whose rows are the columns of A.

- f - The vector of nonlinear functions whose l_2 norm is to be minimized.
  The nonlinear least-squares problem is

$$\min_{x \in \Re^n} \frac{1}{2} f(x)^{\mathrm{T}} f(x), \tag{1.2.1}$$

  where the factor $\frac{1}{2}$ is introduced in order to avoid a factor of two in the derivatives.

- ϕ_i - The ith *residual* function, also the ith component of the vector f.

$$f(x) \equiv \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_m(x) \end{pmatrix}.$$

  An alternative formulation of the nonlinear least-squares problem is

$$\min_{x \in \Re^n} \frac{1}{2} \sum_{i=1}^{m} \phi_i(x)^2, \tag{1.2.2}$$

  where each $\phi_i(x)$ is a smooth function mapping $\Re^n$ to $\Re$.

- J - The Jacobian matrix of f.

$$J(x) \equiv \nabla f(x) = \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \cdots & \frac{\partial \phi_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \cdots & \frac{\partial \phi_m}{\partial x_n} \end{pmatrix}$$

- $\bar{g}$ - The gradient of the nonlinear least-squares objective.

$$\bar{g}(x) \equiv \nabla \left(\frac{1}{2} f(x)^{\mathrm{T}} f(x) \right) = J(x)^{\mathrm{T}} f(x)$$

- B - The part of the Hessian matrix of the nonlinear least-squares objective that involves second derivatives of the residual functions.

$$\nabla^2 \left(\frac{1}{2} f(x)^{\mathrm{T}} f(x) \right) = J(x)^{\mathrm{T}} J(x) + B(x),$$

where

$$B(x) \equiv \sum_{i=1}^{m} \phi_i(x) \nabla^2 \phi_i(x).$$

- $\mathcal{S}^{\perp}$ - If $\mathcal{S}$ is a subspace of $\Re^n$, then the set

$$\mathcal{S}^{\perp} \equiv \{ v \in \Re^n \mid v^{\mathrm{T}} u = 0 \ \text{ for every vector } \ u \in \mathcal{S} \}$$

consisting of all vectors orthogonal to those in $\mathcal{S}$ is also a subspace of $\Re^n$, called the *orthogonal complement* of $\mathcal{S}$ in $\Re^n$. $(\mathcal{S}^{\perp})^{\perp} = \mathcal{S}$.

- $\mathcal{R}(A)$ - The *range* of A.
  If A is an $m \times n$ matrix, then

$$\mathcal{R}(A) \equiv \{ b \mid Ax = b \ \text{ for some } \ x \in \Re^n \}$$

is a subspace of $\Re^m$.

- $\mathcal{N}(A)$ - The *null space* of A.
  If A is an $m \times n$ matrix, then

$$\mathcal{N}(A) \equiv \{ z \mid Az = 0 \}$$

is a subspace of $\Re^n$. $\mathcal{N}(A)$ is the orthogonal complement of $\mathcal{R}(A^T)$ in $\Re^n$.

- ϵ_M - relative machine precision
  If F is the set of floating-point numbers for a particular computer, and $fl(x)$ is the corresponding floating-point representation of a real number x, then

$$\epsilon_M = \max_{\epsilon \in F} \{ fl(1+\epsilon) = fl(1) \}.$$

(see, for example, Chapter 2 of Gill, Murray, and Wright, *Practical Optimization*, Academic Press [1981]).

1.3 Numerical Results : Sources and Presentation

The following is a list of software sources for programs that were used to obtain the results.

NAG	-	Numerical Algorithms Group, Inc.
NPL	-	National Physical Laboratory, England
PORT	-	PORT Mathematical Software Library, A. T. & T. Bell Laboratories, Inc.
ACM	-	Association for Computing Machinery
SOL	-	Systems Optimization Laboratory, Stanford University

All of the programs were run in FORTRAN using double precision on the IBM 3081 and IBM 3033 computers at Stanford Linear Accelerator Center, for which

relative machine precision $\epsilon_M = 2.22\ldots \times 10^{-16}$; $\sqrt{\epsilon_M} = 1.49\ldots \times 10^{-8}$.

In the tables, we include the quantity

$$\frac{\|f^*\|_2^2 - \|f_{best}\|_2^2}{1 + \|f_{best}\|_2^2}, \tag{1.3.1}$$

where f^* is the value of f at the point of termination, and $\|f_{best}\|_2$ is the best available estimate of the norm of the solution, in order to get some idea of the error in $\|f^*\|_2$. For those problems that have nonzero residuals, the value of $\|f_{best}\|_2$ is given to six figures of accuracy, rounded down.

The following abbreviations are used in the headings of the tables :

est. err.	-	error estimate (1.3.1)
conv.	-	termination conditions

The following abbreviations are used in the tables to describe conditions under which the algorithm terminates abnormally :

F LIM.	-	function evaluation limit reached
TIME	-	time limit exceeded
LOOP	-	subroutine appears to loop

A superscipt ^{0} following a problem number indicates a zero-residual problem.

A superscipt ^{L} following a problem number denotes a linear least-squares problem.

See the individual description of each method for additional notation used in the tables.

For information on the test problems, see the Appendix.

2. Unconstrained Optimization

2.1 Overview

This chapter reviews computational techniques for the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} \mathcal{F}(x). \tag{2.1.1}$$

These methods are of interest because nonlinear least squares is a particular instance of (2.1.1), so that special-purpose algorithms for sums of squares should compare favorably in performance with those developed for the more general case. Moreover, much of the motivation for the unconstrained optimization methods is also relevant to algorithm development for nonlinear least squares.

Many algorithms for unconstrained minimization that have proven successful in practice on small to medium-sized dense problems, where $\mathcal{F}$ is smooth, are based on a quadratic model. Subproblems are formulated and solved that minimize a quadratic objective function locally approximating $\mathcal{F}$. Only a brief discussion of these methods will be given; more extensive treatments can be found in Fletcher [1980], Gill, Murray, and Wright [1981], Dennis and Schnabel [1983], and Moré and Sorensen [1984]. There has also been some research concerning methods that are not directly related to Newton's method, including methods based on nonquadratic models (for example, Davidon [1980]; Sorensen [1980]; Schnabel [1983]; Grandinetti [1984]; Tassopoulos and Storey [1984]; Gourgeon and Nocedal [1985]) and continuation methods (for example, Allgower and Georg [1983]). As these techniques are still under development and have yet to be widely used, they will not be discussed here. To conclude the chapter, numerical results are given for comparison with the nonlinear least-squares methods of Chapters 4–6.

2.1.1 Notation and Derivative Information

In addition to the notation given in Section 1.1, we define

$$g(x) \equiv \nabla \mathcal{F}(x)$$

for the gradient of $\mathcal{F}$. We shall assume that it is possible to compute at least the first derivatives of $\mathcal{F}$.

2.2 Optimality Conditions

In this section we list optimality conditions that are straightforward to check computationally. Besides conditions for general smooth functions, quadratic functions are included as a special case because of their role in the algorithms described in this research. Proofs of the optimality conditions may be found in the general references given in Section 2.1.

(2.2-1) (necessary conditions for a minimum)
If x^* is a local minimum of a smooth function $\mathcal{F}$, then x^* is a *stationary point* of $\mathcal{F}$ (that is, $g(x^*) = 0$), and $\nabla^2\mathcal{F}(x^*)$ is positive semi-definite.

(2.2-2) (sufficient conditions for a minimum)
The vector x^* is a local minimum of $\mathcal{F}$ if it is a stationary point of $\mathcal{F}$ and $\nabla^2\mathcal{F}(x^*)$ is positive definite.
Moreover, under these conditions x^* is an *isolated* local minimum because it is the only minimum of $\mathcal{F}$ within an open neighborhood $\{x \mid \|x^* - x\| < \delta\}$ of x^*, for some $\delta > 0$.

(2.2-3) (minimum of a quadratic function)
The vector x^* is a minimum of a quadratic function

$$\mathcal{Q}(x) \equiv \zeta + q^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Qx$$

if and only if $\nabla\mathcal{Q}(x^*) = Qx^* + q = 0$, and Q is positive semi-definite.
Moreover, x^* is the unique minimum of $\mathcal{Q}$ if and only if $\nabla\mathcal{Q}(x^*) = 0$, and Q is positive definite.

2.3 Quadratic Modeling and Local Convergence

It is apparent from the Taylor series expansion

$$\mathcal{F}(x+p) = \mathcal{F}(x) + g(x)^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}\nabla^2\mathcal{F}(x)p + \mathcal{O}(\|p\|^3)$$

that a smooth function $\mathcal{F}$ can be approximated by a quadratic in some neighborhood of each point x in $\Re^n$. The size of the neighborhood in which the approximation is close

depends on x and the nature of $\mathcal{F}$. In Newton's method for unconstrained optimization, the quadratic part of the Taylor series

$$\mathcal{Q}_k(p) \equiv g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 \mathcal{F}_k p$$

is used as a local model for the change in $\mathcal{F}$ at x_k. When $\nabla^2 \mathcal{F}_k$ is positive definite, $\mathcal{Q}_k$ has the unique minimum

$$p_k^N = -\left(\nabla^2 \mathcal{F}_k\right)^{-1} g_k,$$

the Newton search direction at x_k. On the other hand, if $\nabla^2 \mathcal{F}_k$ has any negative eigenvalues, then $\mathcal{Q}_k$ can be made as small as desired by taking large enough steps along a direction of negative curvature. The remaining possibility is that $\nabla^2 \mathcal{F}_k$ is singular but has no negative eigenvalues. In this case, the minimum value of $\mathcal{Q}_k$ is achieved on a affine subspace of $\Re^n$.

The algorithms we shall discuss for unconstrained optimization are based on a quadratic model

$$\tilde{\mathcal{Q}}_k(p) \equiv g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H_k p \tag{2.3.1}$$

in which the matrix H_k is always positive definite, so that the model of the change $\mathcal{F}$ has a well-defined minimum,

$$p_k = -H_k^{-1} g_k,$$

that can be computed efficiently. Positive definiteness of H_k also means that the minimum p_k of $\tilde{\mathcal{Q}}_k$ is a descent direction for $\mathcal{F}$ at x_k, which is essential for linesearch methods (see Section 2.4.1).

An important feature of methods based on a quadratic model is their rate of convergence. *Linear convergence*, defined by the relationship

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \gamma; \quad 0 < \gamma < 1,$$

could be unacceptably slow for practical applications when γ is close to 1. It is therefore desirable to have *superlinear convergence*, which corresponds to the condition

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Newton's method is locally *quadratically convergent* to an isolated local minimum x^* of $\mathcal{F}$, that is,

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} = 0$$

for $x_{k+1} \equiv x_k + p_k^N$ when x_0 is sufficiently close to x^*, and $\nabla^2 \mathcal{F}(x^*)$ is positive definite (see, for example, Moré and Sorensen [1984]). For methods based on (2.3.1), the condition

$$\lim_{k \to \infty} \frac{\|(H_k - \nabla^2 \mathcal{F}(x^*))p_k\|}{\|p_k\|} = 0 \tag{2.3.2}$$

is equivalent to local superlinear convergence of the sequence $\{x_k + p_k\}$ to an isolated local minimum x^* of $\mathcal{F}$ (see Dennis and Moré [1974; 1977]). The relationship (2.3.2) implies that the step p_k approaches the Newton step in both magnitude and direction, although the sequence of matrices $\{H_k\}$ need not converge to $\nabla^2 \mathcal{F}(x^*)$.

The remainder of this chapter is concerned with modifications that are used to enforce convergence from an arbitrary starting point, and with the choice of H_k in (2.3.1). For more detailed information on rates of convergence, see the general references on unconstrained optimization listed in the introduction, and also the book by Ortega and Rheinboldt [1970].

2.4 Basic Strategies

Besides fast local convergence, it is also important that a method make good progress at points away from the solution. Strategies for unconstrained minimization starting from points that may not be close to a solution usually fall into one of two categories : *linesearch* methods and *trust-region* methods, which are described in this section.

2.4.1 Linesearch Approach

Linesearch methods obtain a new iterate in two essentially separate phases. First, a *descent direction* p_k is found for $\mathcal{F}$; that is, a vector p_k is computed for which

$$g_k^T p_k < 0. \tag{2.4.1}$$

Condition (2.4.1) is equivalent to requiring $\mathcal{F}$ to be strictly decreasing along p_k within some neighborhood of x_k. Various algorithms for defining p_k are discussed in Section 2.5. This section is mainly concerned with the second phase of a linesearch method, that of finding a steplength α_k satisfying

$$\mathcal{F}(x_k + \alpha_k p_k) < \mathcal{F}(x_k), \tag{2.4.2}$$

once a descent direction is obtained.

Because of (2.4.1), condition (2.4.2) can be satisfied by choosing a sufficiently small value of α_k, but the result may not be an appreciable reduction in $\mathcal{F}$. In fact, $\{\mathcal{F}(x_k)\}$ can converge to a point that is not a stationary point unless conditions stronger than (2.4.2) are imposed on α_k [see, for example, Dennis and Schnabel (1983), Chapter 6]. On the other hand, finding a minimum of $\mathcal{F}$ along p_k is an iterative process which could require many function and derivative evaluations. Steplength algorithms instead compute α_k that satisfies conditions sufficient to ensure convergence to a stationary point of $\mathcal{F}$ whenever the sequence $\{p_k\}$ is bounded away from orthogonality to the gradient.

The work of Goldstein [1965; 1967], Armijo [1966], Goldstein and Price [1967], and Wolfe [1969; 1971] (see also Ortega and Rheinboldt [1970]) established the fundamental principles underlying most steplength algorithms. A simple strategy for sufficient decrease is based on the condition

$$\mathcal{F}(x_k + \alpha_k p_k) - \mathcal{F}(x_k) \leq \mu \alpha_k g_k^{\mathrm{T}} p_k, \tag{2.4.3}$$

for $\mu \in [0, 1)$. An initial value (usually $\alpha_k = 1$) is tried first, and then a backtracking strategy is used to reduce it until an admissible step is found (see Ortega and Rheinboldt [1970], Gill, Murray, and Wright [1981], Chapter 4, or Dennis and Schnabel [1983], Chapter 6). The steplength strategy of Gill et al. [1979], combines (2.4.3) with the condition

$$|g(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| \leq -\eta g_k^{\mathrm{T}} p_k, \tag{2.4.4}$$

for $\eta \in [0, 1)$, which keeps the steplength bounded away from zero by forcing it to approximate a local minimum of $\mathcal{F}$ along p_k. A procedure for one-dimensional minimization is truncated, using (2.4.4) as the criterion for termination. This is accomplished by polynomial interpolation to the function

$$\Phi(\alpha) \equiv \mathcal{F}(x_k + \alpha p_k), \tag{2.4.5}$$

together with some safeguards to prevent iterates from being either too close together or too far apart. An exact minimization would be carried out for $\eta = 0$ in (2.4.4), while larger values of η increasingly relax this requirement. When $\mu < \eta$, an interval of steplengths satisfies both (2.4.3) and (2.4.4) (for a proof, see Moré and Sorensen [1984]); if μ is chosen sufficiently small, then (2.4.3) almost always holds when (2.4.4) does. When $\mu \geq \eta$, a

backtracking strategy may be used if (2.4.3) fails to hold for the steplength computed in the one-dimensional minimization. If $g_k^T p_k < 0$ and α_k satisfies (2.4.3) and (2.4.4), then

$$\lim_{k\to\infty} \frac{g_k^T p_k}{\|p_k\|_2} = 0,$$

which implies convergence to a stationary point of $\mathcal{F}$ provided $\{p_k\}$ remains uniformly bounded away from orthogonality to $\{g_k\}$ (see Dennis and Schnabel [1983], Chapter 6, and Moré and Sorensen [1984]). If $\mu \leq 0.5$, both conditions (2.4.3) and (2.4.4) are automatically satisfied by superlinearly or quadratically convergent algorithms with $\alpha_k = 1$ when x_k is sufficiently close to a local minimum (for a proof, see Dennis and Schnabel [1983], Chapter 6).

Although the theory allows considerable flexibility in choosing the interpolant to $\Phi(\alpha)$ and other parameters in the univariate minimization, and in the choice of μ and η in (2.4.3) and (2.4.4), in practice performance on difficult problems may be very sensitive to these factors. Moreover, safeguarding in univariate minimization requires specification of a finite interval of uncertainty in which the minimum is presumed to lie. If p_k is very large, it could happen that no satisfactory approximation to a minimum along that direction can be found, resulting in an excessively small step. For more detail on linesearch procedures, see the general references listed in the introduction to this chapter, and also the book by Ortega and Rheinboldt [1970]. An alternative approach is discussed in the next section.

2.4.2 Trust-Region Approach

Trust-region methods were first developed for nonlinear least squares [Levenberg (1944); Morrison (1960); Marquardt (1963)] (see Section 5.2), and later for general unconstrained minimization [Goldfeld, Quandt, and Trotter (1966)]. Motivation for trust-region methods comes from the following observation : if the step to the unconstrained minimum of the current local model for $\mathcal{F}(x+p) - \mathcal{F}(x)$ is relatively large, then it probably falls outside the region in which the model is applicable. The basic idea is to define a neighborhood of the current point over which an approximate minimization of a local model of the change in $\mathcal{F}$ will yield a suitable step to the next iterate.

The local model and constraints defining the neighborhood are chosen so that the subproblem has a well-defined minimum, and so that fast local convergence is possible with

the unconstrained model. Typically, the model at x_k is a quadratic function $g_k^{\mathrm{T}}p+\frac{1}{2}p^{\mathrm{T}}H_kp$, and an upper bound is imposed on a scaled l_2 norm of p, giving the subproblem

$$\min_{p\in\Re^n} g_k^{\mathrm{T}}p+\frac{1}{2}p^{\mathrm{T}}H_kp \tag{2.4.6}$$

$$\text{subject to} \quad \|D_kp\|_2\leq\delta_k.$$

For practical reasons, the scaling matrices D_k are usually diagonal (with positive diagonal entries). Solving (2.4.6) is equivalent to minimizing the quadratic function

$$g_k^{\mathrm{T}}p+\frac{1}{2}p^{\mathrm{T}}\left(H_k+\lambda_kD_k^{\mathrm{T}}D_k\right)p \tag{2.4.7}$$

for some $\lambda_k\geq 0$, where the matrix $H_k+\lambda_kD_k^{\mathrm{T}}D_k$ is at least positive semi-definite.

In practice, it has been found to be more satisfactory to control the value of δ_k directly rather than λ_k (see Moré [1983]). Increases and decreases in δ_k are usually based on comparing the actual reduction

$$\mathcal{F}(x_k+p_k)-\mathcal{F}(x_k)$$

to the reduction predicted by the model,

$$g_k^{\mathrm{T}}p_k+\frac{1}{2}p_k^{\mathrm{T}}H_kp_k;$$

the updating procedure for δ_k can be as simple as multiplying the current value by some prescribed factor, without compromising global convergence properties (see below). The preferred strategy for decreasing δ_k is more complicated. An approximate minimum τ_k of $\mathcal{F}(x_k+\tau p_k)$ is computed by safeguarded polynomial interpolation (as in linesearch methods — see Section 2.4.1), and $\tau_k\|D_kp_k\|_2$ is taken to be the new value of δ_k (see Fletcher [1980], Chapter 5, Dennis and Schnabel [1983], Chapter 6, and Moré [1983]). It may be necessary to decrease δ_k a number of times before a suitable reduction in $\mathcal{F}$ is achieved and the step to a new iterate can be taken.

Once δ_k is assigned a value, it remains to find p_k when the solution to (2.4.6) is not an unconstrained minimum. Moré and Sorensen [1983] obtain λ_k in (2.4.7) by truncating a numerical procedure for finding a zero of the function

$$\Psi(\lambda)\equiv\|p_k(\lambda)\|_2-\delta_k\equiv\left\|\left(H_k+\lambda D_k^{\mathrm{T}}D_k\right)^{-1}g_k\right\|_2-\delta_k, \tag{2.4.8}$$

based on the work of Hebden [1973] (see also Reinsch [1971]) and Gay [1981]. The algorithm of Gay [1983], implemented in the PORT Library [1984], approximates $p_k(\lambda)$ by a linear combination of the (scaled) steepest descent direction and the Newton direction. This technique was devised by Powell [1970] (see also Dennis and Mei [1979]), and is used because it achieves comparable performance to methods that attempt to approximate $\Psi(\lambda)$ closely, with considerably less computational effort.

Somewhat stronger convergence results have been proven for trust-region methods than are known for linesearch methods (see Section 2.4.1). Trust-region methods can be shown to converge to an isolated local minimum under fairly mild conditions when exact second derivatives are used, and otherwise to a stationary point (see Fletcher [1980], Chapter 5, Moré [1983], or Moré and Sorensen [1984]). Although global convergence properties are not affected, in practice the choice of δ_0 and the updating strategy for δ_k are important. As δ_k, and hence the norm of p, is made to approach zero, the minimizer of the quadratic becomes parallel to the steepest descent direction, $-g_k$. Small values of δ_k are accordingly safe, in the sense that they guarantee a decrease, but progress may be unacceptably slow if no provision is made for taking larger steps where possible.

For more detail and general discussion of trust-region methods, see Moré [1983] and Shultz, Schnabel, and Byrd [1985], as well as the general references on unconstrained optimization listed in the introduction. A variant in which a trust region is applied to a two-dimensional subspace at each step is described in Bulteau and Vial [1985].

2.4.3 Stationary Points and Directions of Negative Curvature

It is possible to decrease $\mathcal{F}$ at a stationary point x^* (see Section 2.2) if the Hessian matrix has one or more negative eigenvalues. The decrease is obtained by moving along a direction of negative curvature; in other words, a direction p for which $p^{\mathrm{T}}\nabla^2\mathcal{F}(x^*)p < 0$. Trust-region methods that use the quadratic model with exact Hessian information (see Section 2.5.1) will yield directions of negative curvature at stationary points when $\nabla^2\mathcal{F}(x^*)$ is indefinite, whereas the linesearch methods discussed above terminate when the gradient vanishes.

A fundamental problem is that of deciding the length of the step to be taken along a direction of negative curvature. This is very much related to the problem of setting a maximum step length in order to safeguard a linesearch method, or that of determining the

step bound in a trust-region method. First- and second-order information about the function at x^* indicates that an infinite step can be taken, since the quadratic part of the Taylor series at x^* is unbounded below when $\nabla^2 \mathcal{F}(x^*)$ is indefinite. Clearly this is not possible if $\mathcal{F}$ has a finite minimum.

Neither the question of choosing a direction of negative curvature, nor the problem of choosing a steplength along such directions has been adequately resolved, and thus in most methods directions of negative curvature are not explicitly sought at arbitrary points. For research on generating directions of negative curvature, and on their use in unconstrained optimization algorithms, see Gill and Murray [1974], Fletcher and Freeman [1977], McCormick [1977], Moré and Sorensen [1979], Goldfarb [1980], and Shultz, Schnabel, and Byrd [1985].

2.5 Defining the Quadratic Model

In this section we describe various ways of defining H_k in (2.3.1) so that condition (2.3.2) for superlinear convergence is satisfied.

2.5.1 Second-Derivative Methods

There are two basic frameworks for defining H_k in the quadratic model (2.5.2) when second derivative information is available: direct modification of the Hessian, and trust-region methods. Both can be viewed as procedures for producing a positive-definite quadratic model by modifying the exact Hessian $\nabla^2 \mathcal{F}_k$. A method that combines the two approaches is given in Chapter 5 (Section 5.5) of Dennis and Schnabel [1983].

The modified Newton method of Gill and Murray [1974a] is a linesearch method in which the definition of the search direction is based on the fact that if H_k is positive definite, it can be characterized by its *Cholesky factorization*

$$H_k = R_k^{\mathrm{T}} R_k, \tag{2.5.1}$$

where R_k is upper-triangular and nonsingular (see, for example, Stewart [1973], Chapter 3). Gill and Murray alter the Cholesky factorization procedure so that it can be continued in the event of indefiniteness or near-singularity. The modified factorization is applied to the Hessian matrix $\nabla^2 \mathcal{F}_k$, resulting in the Cholesky factorization of a matrix H_k with a

prescribed upper bound on its condition number. The matrix H_k may differ from $\nabla^2 \mathcal{F}_k$ only in the diagonal elements. When H_k is used to define the quadratic model (2.3.1), local convergence properties of of Newton's method are preserved, because $H_k = \nabla^2 \mathcal{F}_k$ whenever $\nabla^2 \mathcal{F}_k$ is sufficiently positive definite. An implementation is available in the NAG Library [1984] (subroutine E04LBF). See Greenstadt [1967], Murray [1972], and Higham [1986], as well as Gill, Murray, and Wright [1981], Chapter 4, for information on other direct modification methods.

In trust-region methods with exact Hessian information, a subproblem of the form

$$\min g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 \mathcal{F}_k p \tag{2.5.2}$$

$$\text{subject to} \quad \|D_k p\|_2 \leq \delta_k,$$

is solved for the step p_k to the next iterate. We recall from the overview of trust-regions in Section 2.4.2 that solving (2.5.2) is equivalent to solving

$$\min g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \left(\nabla^2 \mathcal{F}_k + \lambda_k D_k^{\mathrm{T}} D_k \right) p \tag{2.5.3}$$

for some non-negative value of λ_k, with $\nabla^2 \mathcal{F}_k + \lambda_k D_k^{\mathrm{T}} D_k$ positive semidefinite. In particular, λ_k will be positive whenever $\nabla^2 \mathcal{F}_k$ is indefinite, and also when $\nabla^2 \mathcal{F}_k$ is positive-definite if δ_k happens to be smaller than the scaled unconstrained minimum of the quadratic objective. In contrast to the modified Newton method described above, all of the eigenvalues of $\nabla^2 \mathcal{F}_k$ are changed when $\lambda_k > 0$ in (2.5.3). As long as the constraint in (2.5.2) is inactive near a local minimum, the local convergence behavior of Newton's method is preserved. A recent implementation of a trust-region method that uses second derivatives is available in the PORT Library [1984] (subroutine DMNH; see also Gay [1983]). For further information on trust-regions with exact Hessian information, see Fletcher [1980], Chapter 5, Gay [1981], Sorensen [1982], Moré [1983], Moré and Sorensen [1983], and Shultz, Schnabel, and Byrd [1985].

2.5.2 Quasi-Newton Methods

In *quasi-Newton* methods (also called *variable metric* or *secant* methods), a sequence of approximations $H_0, H_1, \ldots$, to the Hessian matrix of $\mathcal{F}$ is generated, with H_{k+1} depending on H_k as well as on gradient information at the current iterate. The approximate

Hessian is required to satisfy the *quasi-Newton* condition

$$H_{k+1}s_k = y_k, \tag{2.5.4}$$

$$s_k \equiv x_{k+1} - x_k; \quad y_k \equiv g_{k+1} - g_k,$$

motivated by the Taylor expansion of the gradient :

$$g_{k+1} = g_k + \nabla^2 \mathcal{F}_k s_k + \mathcal{O}(\|s_k\|^2). \tag{2.5.5}$$

The quantity $y_k^{\mathrm{T}} s_k$ approximates the curvature, $s_k^{\mathrm{T}} \nabla^2 \mathcal{F}_k s_k$, of $\mathcal{F}$ along s_k. Equation (2.5.6) does not uniquely define H_{k+1}, and much research has been directed toward developing criteria for completing the specification (see, for example, Dennis and Moré [1977], Nazareth [1984], Todd [1984], and Flachs [1986], as well as the general references listed in the introduction to this chapter). Conditions imposed on the approximate Hessian almost always include symmetry and positive definiteness.

It is generally agreed that the best overall performance is achieved by the BFGS update

$$H_{k+1} = H_k - \frac{H_k s_k (H_k s_k)^{\mathrm{T}}}{s_k^{\mathrm{T}} H_k s_k} + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}, \tag{2.5.6}$$

although precise reasons for its superiority are still not known (see Brodlie [1977], as well as the general references). Like most proposed updates, the BFGS update is a rank-two modification of the current approximate Hessian. The BFGS update preserves positive definiteness whenever $y_k^{\mathrm{T}} s_k > 0$, a condition that holds automatically in a linesearch method satisfying (2.4.4).

Originally, quasi-Newton updates were formulated in terms of H_k^{-1} rather than H_k, so that minimizing the quadratic (2.5.2) at each stage in a linesearch algorithm involved only a matrix multiplication ($\mathcal{O}(n^2)$ arithmetic operations) rather than solution of a linear system ($\mathcal{O}(n^3)$ arithmetic operations). Gill and Murray [1972] showed that rank-two quasi-Newton methods could be implemented in $\mathcal{O}(n^2)$ operations per iteration by applying an update to a Cholesky factor (see Section 2.5.1) of H_k. This has the additional advantage that it allows the numerical positive definiteness of H_k to be monitored from iteration to iteration. For more information on computational aspects of the update, see Dennis and Schnabel [1983], Chapter 9, and Gill et al. [1985].

The BFGS method belongs to a class of quasi-Newton methods that can be derived by minimizing the difference $(H_{k+1} - H_k)$ or $(H_{k+1}^{-1} - H_k^{-1})$, in various weighted norms, subject

to (2.5.6) [Dennis and Schnabel (1979)]. Other classes of methods attempt to minimize the condition number of H_k by selecting parameters in a class of updates at each step [Shanno (1970); Oren (1973, 1982); Davidon (1975); Oren and Spedicato (1976); Spedicato (1976); Schnabel (1978)]. Al-Baali and Fletcher (1985)] apply a scaling factor before updating that minimizes an approximate measure of the error in the inverse Hessian matrix. Performance tests indicate that these modified methods are not as successful as the BFGS method for general problems [Brodlie (1977); Shanno and Phua (1978b); Al-Baali and Fletcher (1985)].

Under the same assumptions as required for local quadratic convergence of Newton's method, quasi-Newton methods are locally superlinearly convergent, provided H_0 is sufficiently close to $\nabla^2 \mathcal{F}(x_0)$ [Broyden, Dennis, and Moré (1973)]. For quasi-Newton methods, superlinear convergence to x^* is equivalent to condition (2.3.2), so that the sequence $\{H_k\}$ of approximate Hessians need not converge to the exact Hessian at the solution. Convergence of $\{H_k\}$ is discussed in Ge and Powell [1983] and Stoer [1984].

Selection of the initial Hessian approximation H_0 can be critical to the success of a quasi-Newton method. Often the identity is chosen because it gives the steepest-descent direction on the first iteration, and it is positive definite. Computational tests have shown that improved performance can sometimes be achieved by scaling H_0 before performing the first update [Shanno and Phua (1978a); Dennis and Schnabel (1983), Chapter 9]. Another possibility is to use a finite-difference approximation to $\nabla^2 \mathcal{F}(x_0)$ for H_0, modified if necessary to ensure positive definiteness. Although the choice of H_0 can have a significant effect on performance, the question of how best to choose H_0 is still open. It is generally agreed that exact or approximate curvature information should be used to start the algorithm if it is available at a reasonable cost. In nonlinear least squares, the form of the Hessian matrix allows a special choice to be made for the initial approximation (see Section 5.5.1).

2.6 Numerical Results

In this section numerical results are presented for particular implementations of the methods discussed in this chapter. The tests were performed using the following software (described in more detail in the next three subsections) :

method	derivative information	global strategy	subroutine	source
modified Newton	second	linesearch	MNA/E04LBF	NPL/NAG
quasi-Newton (BFGS)	first	linesearch	NPSOL	SOL/NAG
modified Newton	second	trust region	DMNH/HUMSL	PORT/ACM
quasi-Newton (BFGS)	first	trust region	DMNG/SUMSL	PORT/ACM

In the tables, we include the quantity

$$\frac{\|f^*\|_2^2 - \|f_{best}\|_2^2}{1 + \|f_{best}\|_2^2}, \tag{2.6.1}$$

where f^* is the value of f at the point of termination, and $\|f_{best}\|_2$ is the best available estimate of the norm of the solution, in order to get some idea of the error in $\|f^*\|_2$. For those problems that have nonzero residuals, the value of $\|f_{best}\|_2$ is given to six figures of accuracy, rounded down.

For further details on the numerical tests, see Section 1.3, as well as the individual description of each method that follows. For information on the test problems, see the Appendix.

Since our main purpose in presenting these results is to compare them with those for specialized methods for nonlinear least squares given in Chapters 4 and 5, discussion is postponed until Section 5.7.

2.6.1 Second-Derivative (Modified-Newton) Linesearch Method (NPL/NAG MNA)

2.6.1.1 Software and Algorithm

The results were obtained using subroutine MNA from the National Physical Laboratory, available at Stanford Linear Accelerator Center. The algorithm implements a modified Newton method in which the search direction at each iteration is the solution to a subproblem of the form

$$\min_{p \in \mathbb{R}^n} \bar{g}_k^T p + \frac{1}{2} p^T H_k p,$$

and the exact Hessian matrix is replaced by modified Cholesky factors if it is either indefinite or computationally singular (see Gill and Murray [1974a] and Section 2.5.1). A step length along the search direction is then computed by a linesearch method [Gill and Murray (1974b)] that uses both function and gradient information to obtain sufficient decrease in the objective function. MNA requires exact second derivatives, and is similar to subroutine E04LBF from the NAG Library [1984], the principal difference being that the latter allows specification of fixed upper and lower bounds on the variables.

2.6.1.2 Parameters

Parameters were kept at their default values with the following exceptions :

MAXCAL	-	min $\{9999, 1000n\}$	function evaluation limit
XTOL	-	varied; see tables	accuracy in x
ETA	-	0.5	linesearch accuracy
STEPMX	-	usually 10^6 (default) †	maximum step for linesearch

† In some cases the default STEPMX $= 10^6$ was too large and overflow occurred during function evaluation in the linesearch. These cases are indicated in the table by giving the lower value of STEPMX that was subsequently used to obtain the results in the column labeled "max. step".

See NAG [1984] for details concerning the parameters.

2.6.1.3 Convergence Criteria

The following quantities will be used in describing the convergence criteria :

objective function	:	$\mathcal{F}_k \ (= \frac{1}{2} f_k^T f_k)$
objective gradient	:	$g_k = \nabla \mathcal{F}_k \ (= J_k^T f_k)$
search direction	:	p_k, the minimizer of the subproblem
steplength	:	α_k, determined by the linesearch

An iterate is determined to be optimal by MNA if the following four conditions hold :

$$\alpha_k \|p_k\|_2 < (\texttt{XTOL} + \sqrt{\epsilon_M})(1 + \|x_k\|_2) \tag{2.6.1}$$

and

$$\mathcal{F}_{k-1} - \mathcal{F}_k < (\texttt{XTOL}^2 + \epsilon_M)(1 + |\mathcal{F}_k|) \tag{2.6.2}$$

and

$$\|g_k\|_2 < (\texttt{XTOL} + \epsilon_M^{1/3})(1 + |\mathcal{F}_k|) \tag{2.6.3}$$

and

$$\nabla^2 \mathcal{F}_k \text{ is positive definite,} \tag{2.6.4}$$

or if

$$\|g_k\|_2 < 0.01\sqrt{\epsilon_M}. \tag{2.6.5}$$

A necessary condition for optimality is that the gradient vanish, and conditions (2.6.3) and (2.6.5) are intended to test whether this requirement is approximately satisfied at x_k. Conditions (2.6.1) and (2.6.2) are meant to ensure that the sequence $\{x_k\}$ has converged, while condition (2.6.4), together with condition (2.6.3), implies that sufficient conditions for a strict local minimum appear to hold at x_k. Condition (2.6.5) allows MNA to accept a point as a local mimimum if a more restrictive test than (2.6.1) on the necessary condition is met, but one or more of the other conditions for convergence do not hold. For a detailed discussion of convergence criteria similar to these, see Section 8.2 of Gill, Murray, and Wright [1981].

The following abbreviations are used in the tables to describe the conditions under which the algorithm terminates :

OPT.	-	optimal point found
*	-	current point cannot be improved †
F LIM.	-	function evaluation limit reached
TIME	-	time limit exceeded

† A '*' corresponds to the situation in which the algorithm terminates due to failure in the linesearch to find an acceptable step at the current iteration.

Numerical Results for MNA

	n	m	XTOL	max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-6}		15	11	1.41	10^{-11}	10^{-10}	10^{-21}	*
			10^{-10}		15	11	1.41	10^{-11}	10^{-10}	10^{-21}	*
2.^{0}	2	2	10^{-6}		8	6	6.40	10^{-12}	10^{-10}	10^{-24}	OPT.
			10^{-10}		8	6	6.40	10^{-12}	10^{-10}	10^{-24}	OPT.
3.^{0}	2	2	10^{-6}		129	41	9.11	10^{-11}	10^{-6}	10^{-22}	*
			10^{-10}		129	41	9.11	10^{-11}	10^{-6}	10^{-22}	*
4.^{0}	2	3	10^{-6}		1	1	10^{6}	10^{-16}	10^{-10}	10^{-32}	*
			10^{-10}		1	1	10^{6}	10^{-16}	10^{-10}	10^{-32}	*
5.^{0}	2	3	10^{-6}	10^{3}	47	9	3.04	10^{-10}	10^{-10}	10^{-20}	OPT.
			10^{-10}	10^{3}	47	9	3.04	10^{-10}	10^{-10}	10^{-20}	OPT.
6.	2	10	10^{-6}		12	10	.365	10^{1}	10^{-7}	10^{-6}	OPT.
			10^{-10}		12	10	.365	10^{1}	10^{-7}	10^{-6}	OPT.
7.^{0}	3	3	10^{-6}		12	9	1.00	10^{-16}	10^{-15}	10^{-31}	OPT.
			10^{-10}		12	9	1.00	10^{-16}	10^{-15}	10^{-31}	OPT.
8.	3	15	10^{-6}		11	11	2.60	10^{-1}	10^{-15}	10^{-8}	OPT.
			10^{-10}		11	11	2.60	10^{-1}	10^{-15}	10^{-8}	OPT.
9.	3	15	10^{-6}		3	3	1.08	10^{-4}	10^{-11}	10^{-14}	OPT.
			10^{-10}		3	3	1.08	10^{-4}	10^{-11}	10^{-14}	OPT.
10.	3	16	10^{-6}		274	180	10^{4}	10^{1}	10^{-1}	10^{-6}	*
			10^{-10}		274	180	10^{4}	10^{1}	10^{-1}	10^{-6}	*
11.^{0}	3	10	10^{-6}		532	332	55.9	10^{-11}	10^{-11}	10^{-22}	OPT.
			10^{-10}		532	332	55.9	10^{-11}	10^{-11}	10^{-22}	OPT.
12.^{0}	3	10	10^{-6}		41	20	10.1	10^{-13}	10^{-13}	10^{-26}	OPT.
			10^{-10}		41	20	10.1	10^{-13}	10^{-13}	10^{-26}	OPT.
13.^{0}	4	4	10^{-6}		23	23	10^{-4}	10^{-7}	10^{-10}	10^{-15}	OPT.
			10^{-10}		23	23	10^{-4}	10^{-7}	10^{-10}	10^{-15}	OPT.
14.^{0}	4	6	10^{-6}		50	20	2.00	10^{-11}	10^{-9}	10^{-21}	*
			10^{-10}		50	20	2.00	10^{-11}	10^{-9}	10^{-21}	*
15.	4	11	10^{-6}		20	8	.328	10^{-2}	10^{-13}	10^{-9}	OPT.
			10^{-10}		20	8	.328	10^{-2}	10^{-13}	10^{-9}	OPT.
16.	4	20	10^{-6}		9	9	17.6	10^{2}	10^{-10}	10^{-8}	OPT.
			10^{-10}		11	11	17.6	10^{2}	10^{-6}	10^{-8}	*
17.	5	33	10^{-6}		43	27	2.46	10^{-2}	10^{-10}	10^{-11}	OPT.
			10^{-10}		44	28	2.46	10^{-2}	10^{-10}	10^{-11}	*
18.^{0}	6	13	10^{-6}		31	15	12.3	10^{-15}	10^{-15}	10^{-29}	OPT.
			10^{-10}		31	15	12.3	10^{-15}	10^{-15}	10^{-29}	OPT.
19.	11	65	10^{-6}	10.0	7	4	9.38	10^{-1}	10^{-9}	10^{-8}	OPT.
			10^{-10}	10.0	8	5	9.38	10^{-1}	10^{-11}	10^{-8}	OPT.

Numerical Results for MNA

	n	m	XTOL	max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-6}		15	14	2.44	10^{-2}	10^{-14}	10^{-10}	OPT.
			10^{-10}		15	14	2.44	10^{-2}	10^{-14}	10^{-10}	OPT.
20b.	9	31	10^{-6}		15	14	6.06	10^{-3}	10^{-10}	10^{-13}	OPT.
			10^{-10}		15	14	6.06	10^{-3}	10^{-10}	10^{-13}	OPT.
20c.	12	31	10^{-6}		15	14	16.6	10^{-5}	10^{-10}	10^{-16}	OPT.
			10^{-10}		15	14	16.6	10^{-5}	10^{-10}	10^{-16}	OPT.
20d.	20	31	10^{-6}		(352)	(189)	10^{6}	10^{-3}	10^{-5}	10^{-5}	TIME
			10^{-10}		(251)	(135)	10^{6}	10^{-3}	10^{-5}	10^{-5}	TIME
21a.^{0}	10	10	10^{-6}		15	11	3.16	10^{-10}	10^{-9}	10^{-21}	*
			10^{-10}		15	11	3.16	10^{-10}	10^{-9}	10^{-21}	*
21b.^{0}	20	20	10^{-6}		15	11	4.47	10^{-10}	10^{-9}	10^{-20}	*
			10^{-10}		15	11	4.47	10^{-10}	10^{-9}	10^{-20}	*
22a.^{0}	12	12	10^{-6}		23	23	10^{-4}	10^{-7}	10^{-10}	10^{-14}	OPT.
			10^{-10}		23	23	10^{-4}	10^{-7}	10^{-10}	10^{-14}	OPT.
22b.^{0}	20	20	10^{-6}		24	24	10^{-4}	10^{-7}	10^{-11}	10^{-15}	OPT.
			10^{-10}		24	24	10^{-4}	10^{-7}	10^{-11}	10^{-15}	OPT.
23a.	4	5	10^{-6}		47	36	.500	10^{-3}	10^{-15}	10^{-10}	OPT.
			10^{-10}		47	36	.500	10^{-3}	10^{-15}	10^{-10}	OPT.
23b.	10	11	10^{-6}		46	37	.500	10^{-2}	10^{-11}	10^{-11}	OPT.
			10^{-10}		46	37	.500	10^{-2}	10^{-11}	10^{-11}	OPT.
24a.	4	8	10^{-6}		159	111	.759	10^{-3}	10^{-12}	10^{-11}	OPT.
			10^{-10}		159	111	.759	10^{-3}	10^{-12}	10^{-11}	OPT.
24b.	10	20	10^{-6}		133	94	.598	10^{-2}	10^{-11}	10^{-9}	OPT.
			10^{-10}		133	94	.598	10^{-2}	10^{-11}	10^{-9}	OPT.
25a.^{0}	10	12	10^{-6}		14	14	3.16	10^{-10}	10^{-8}	10^{-19}	*
			10^{-10}		14	14	3.16	10^{-10}	10^{-8}	10^{-19}	*
25b.^{0}	20	22	10^{-6}		17	17	4.47	10^{-8}	10^{-6}	10^{-15}	OPT.
			10^{-10}		18	18	4.47	10^{-8}	10^{-6}	10^{-15}	*
26a.^{0}	10	10	10^{-6}		22	11	.306	10^{-14}	10^{-15}	10^{-29}	OPT.
			10^{-10}		23	12	.306	10^{-14}	10^{-15}	10^{-29}	OPT.
26b.^{0}	20	20	10^{-6}		27	13	.189	10^{-12}	10^{-13}	10^{-25}	OPT.
			10^{-10}		27	13	.189	10^{-12}	10^{-13}	10^{-25}	OPT.
27a.^{0}	10	10	10^{-6}		22	12	3.18	10^{-12}	10^{-12}	10^{-24}	OPT.
			10^{-10}		22	12	3.18	10^{-12}	10^{-12}	10^{-24}	OPT.
27b.^{0}	20	20	10^{-6}		30	16	4.47	10^{-11}	10^{-11}	10^{-22}	OPT.
			10^{-10}		30	16	4.47	10^{-11}	10^{-11}	10^{-22}	OPT.
28a.^{0}	10	10	10^{-6}		4	4	.412	10^{-12}	10^{-13}	10^{-24}	OPT.
			10^{-10}		4	4	.412	10^{-12}	10^{-13}	10^{-24}	OPT.
28b.^{0}	20	20	10^{-6}		4	4	.571	10^{-13}	10^{-14}	10^{-25}	OPT.
			10^{-10}		4	4	.571	10^{-13}	10^{-14}	10^{-25}	OPT.

Numerical Results for MNA

	n	m	XTOL	max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
29a.^{0}	10	10	10^{-6}		4	4	.412	10^{-10}	10^{-10}	10^{-20}	OPT.
			10^{-10}		4	4	.412	10^{-10}	10^{-10}	10^{-20}	OPT.
29b.^{0}	20	20	10^{-6}		4	4	.571	10^{-10}	10^{-10}	10^{-20}	OPT.
			10^{-10}		4	4	.571	10^{-10}	10^{-10}	10^{-20}	OPT.
30a.^{0}	10	10	10^{-6}		7	7	2.05	10^{-16}	10^{-15}	10^{-31}	OPT.
			10^{-10}		7	7	2.05	10^{-16}	10^{-15}	10^{-31}	OPT.
30b.^{0}	20	20	10^{-6}		7	7	3.04	10^{-15}	10^{-14}	10^{-30}	OPT.
			10^{-10}		7	7	3.04	10^{-15}	10^{-14}	10^{-30}	OPT.
31a.^{0}	10	10	10^{-6}		9	9	1.80	10^{-13}	10^{-12}	10^{-26}	OPT.
			10^{-10}		9	9	1.80	10^{-13}	10^{-12}	10^{-26}	OPT.
31b.^{0}	20	20	10^{-6}		9	9	2.66	10^{-13}	10^{-12}	10^{-26}	OPT.
			10^{-10}		9	9	2.66	10^{-13}	10^{-12}	10^{-26}	OPT.
32.^{L}	10	20	10^{-6}		4	3	3.16	10^{0}	10^{-15}	10^{-16}	OPT.
			10^{-10}		4	3	3.16	10^{0}	10^{-15}	10^{-16}	OPT.
33.^{L}	10	20	10^{-6}		27	4	54.0	10^{0}	10^{-8}	10^{-6}	*
			10^{-10}		27	4	54.0	10^{0}	10^{-8}	10^{-6}	*
34.^{L}	10	20	10^{-6}		20	3	21.2	10^{0}	10^{-10}	10^{-6}	*
			10^{-10}		20	3	21.2	10^{0}	10^{-10}	10^{-6}	*
35a.	8	8	10^{-6}		46	15	1.65	10^{-1}	10^{-10}	10^{-9}	*
			10^{-10}		46	15	1.65	10^{-1}	10^{-10}	10^{-9}	*
35b.^{0}	9	9	10^{-6}		94	25	1.73	10^{-10}	10^{-10}	10^{-20}	OPT.
			10^{-10}		94	25	1.73	10^{-10}	10^{-10}	10^{-20}	OPT.
35c.	10	10	10^{-6}	10.0	59	17	1.76	10^{-1}	10^{-9}	10^{-9}	OPT.
			10^{-10}	10.0	60	18	1.76	10^{-1}	10^{-9}	10^{-9}	*
36a.^{0}	4	4	10^{-6}		(4001)	(2663)	50.4	10^{-9}	10^{-9}	10^{-19}	F LIM.
			10^{-10}		(4001)	(2663)	50.4	10^{-9}	10^{-9}	10^{-19}	F LIM.
36b.^{0}	9	9	10^{-6}		65	10	51.1	10^{-9}	10^{-10}	10^{-18}	OPT.
			10^{-10}		(1751)	(1135)	50.9	10^{-9}	10^{-9}	10^{-18}	TIME
36c.^{0}	9	9	10^{-6}		(9008)	(978)	1.74	10^{-6}	10^{-6}	10^{-11}	F LIM.
			10^{-10}		(9008)	(978)	1.74	10^{-6}	10^{-6}	10^{-11}	F LIM.
36d.^{0}	9	9	10^{-6}		(9001)	(1737)	225.	10^{-4}	10^{-4}	10^{-8}	F LIM.
			10^{-10}		(9001)	(1737)	225.	10^{-4}	10^{-4}	10^{-8}	F LIM.
37.	2	16	10^{-6}		6	5	8.85	10^{1}	10^{-10}	10^{-6}	OPT.
			10^{-10}		6	5	8.85	10^{1}	10^{-10}	10^{-6}	OPT.
38.	3	16	10^{-6}		12	7	26.1	10^{1}	10^{-13}	10^{-6}	OPT.
			10^{-10}		12	7	26.1	10^{1}	10^{-13}	10^{-6}	OPT.

Numerical Results for MNA

	n	m	XTOL	max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
39a.	2	3	10^{-6}		4	3	10^{-6}	10^{-1}	10^{-17}	10^{-7}	OPT.
			10^{-10}		4	3	10^{-6}	10^{-1}	10^{-17}	10^{-7}	OPT.
39b.	2	3	10^{-6}		4	3	10^{-7}	10^{-1}	10^{-11}	10^{-7}	OPT.
			10^{-10}		4	3	10^{-7}	10^{-1}	10^{-11}	10^{-7}	OPT.
39c.	2	3	10^{-6}		4	3	10^{-7}	10^{-1}	10^{-13}	10^{-7}	OPT.
			10^{-10}		4	3	10^{-7}	10^{-1}	10^{-13}	10^{-7}	OPT.
39d.	2	3	10^{-6}		6	5	10^{-7}	10^{-1}	10^{-17}	10^{-7}	OPT.
			10^{-10}		6	5	10^{-7}	10^{-1}	10^{-17}	10^{-7}	OPT.
39e.	2	3	10^{-6}		8	7	10^{-8}	10^{-1}	10^{-14}	10^{-7}	OPT.
			10^{-10}		8	7	10^{-8}	10^{-1}	10^{-14}	10^{-7}	OPT.
39f.	2	3	10^{-6}		11	10	10^{-9}	10^{-1}	10^{-8}	10^{-7}	*
			10^{-10}		11	10	10^{-9}	10^{-1}	10^{-8}	10^{-7}	*
39g.	2	3	10^{-6}		14	11	10^{-10}	10^{-1}	10^{-9}	10^{-7}	*
			10^{-10}		14	11	10^{-10}	10^{-1}	10^{-9}	10^{-7}	*
40a.	3	4	10^{-6}		4	3	10^{-6}	10^{0}	10^{-15}	10^{-7}	OPT.
			10^{-10}		4	3	10^{-6}	10^{0}	10^{-15}	10^{-7}	OPT.
40b.	3	4	10^{-6}		4	3	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
			10^{-10}		4	3	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
40c.	3	4	10^{-6}		5	4	10^{-7}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-10}		5	4	10^{-7}	10^{0}	10^{-12}	10^{-7}	OPT.
40d.	3	4	10^{-6}		6	5	10^{-7}	10^{0}	10^{-16}	10^{-7}	OPT.
			10^{-10}		6	5	10^{-7}	10^{0}	10^{-16}	10^{-7}	OPT.
40e.	3	4	10^{-6}		8	7	10^{-7}	10^{0}	10^{-8}	10^{-7}	*
			10^{-10}		8	7	10^{-7}	10^{0}	10^{-8}	10^{-7}	*
40f.	3	4	10^{-6}		10	9	10^{-8}	10^{0}	10^{-11}	10^{-7}	OPT.
			10^{-10}		10	9	10^{-8}	10^{0}	10^{-11}	10^{-7}	OPT.
40g.	3	4	10^{-6}		13	12	10^{-9}	10^{0}	10^{-6}	10^{-7}	*
			10^{-10}		13	12	10^{-9}	10^{0}	10^{-6}	10^{-7}	*
41a.	5	10	10^{-6}		4	3	10^{-6}	10^{0}	10^{-9}	10^{-7}	*
			10^{-10}		4	3	10^{-6}	10^{0}	10^{-9}	10^{-7}	*
41b.	5	10	10^{-6}		4	3	10^{-6}	10^{0}	10^{-13}	10^{-7}	OPT.
			10^{-10}		4	3	10^{-6}	10^{0}	10^{-13}	10^{-7}	OPT.
41c.	5	10	10^{-6}		8	7	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-10}		8	7	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
41d.	5	10	10^{-6}		9	8	10^{-6}	10^{0}	10^{-14}	10^{-7}	OPT.
			10^{-10}		9	8	10^{-6}	10^{0}	10^{-14}	10^{-7}	OPT.
41e.	5	10	10^{-6}		12	11	10^{-7}	10^{0}	10^{-10}	10^{-7}	*
			10^{-10}		12	11	10^{-7}	10^{0}	10^{-10}	10^{-7}	*
41f.	5	10	10^{-6}		14	13	10^{-7}	10^{0}	10^{-11}	10^{-7}	OPT.
			10^{-10}		14	13	10^{-7}	10^{0}	10^{-11}	10^{-7}	OPT.
41g.	5	10	10^{-6}		17	16	10^{-8}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-10}		17	16	10^{-8}	10^{0}	10^{-12}	10^{-7}	OPT.

Numerical Results for MNA

	n	m	XTOL	max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-6}	2.0	15	5	60.3	10^{-13}	10^{-10}	10^{-26}	OPT.
			10^{-10}	2.0	15	5	60.3	10^{-13}	10^{-10}	10^{-26}	OPT.
42b.^{0}	4	24	10^{-6}		16	7	60.3	10^{-13}	10^{-10}	10^{-25}	OPT.
			10^{-10}		16	7	60.3	10^{-13}	10^{-10}	10^{-25}	OPT.
42c.^{0}	4	24	10^{-6}		6	4	60.3	10^{-11}	10^{-9}	10^{-22}	*
			10^{-10}		6	4	60.3	10^{-11}	10^{-9}	10^{-22}	*
42d.^{0}	4	24	10^{-6}		6	4	60.3	10^{-13}	10^{-10}	10^{-26}	OPT.
			10^{-10}		6	4	60.3	10^{-13}	10^{-10}	10^{-26}	OPT.
43a.^{0}	5	16	10^{-6}	10.0	28	16	54.0	10^{-10}	10^{-9}	10^{-21}	*
			10^{-10}	10.0	28	16	54.0	10^{-10}	10^{-9}	10^{-21}	*
43b.^{0}	5	16	10^{-6}	10.0	24	12	54.0	10^{-12}	10^{-11}	10^{-24}	OPT.
			10^{-10}	10.0	24	12	54.0	10^{-12}	10^{-11}	10^{-24}	OPT.
43c.^{0}	5	16	10^{-6}	10.0	22	11	54.0	10^{-13}	10^{-11}	10^{-25}	OPT.
			10^{-10}	10.0	22	11	54.0	10^{-13}	10^{-11}	10^{-25}	OPT.
43d.^{0}	5	16	10^{-6}	10^2	41	18	54.0	10^{-14}	10^{-11}	10^{-27}	OPT.
			10^{-10}	10^2	41	18	54.0	10^{-14}	10^{-11}	10^{-27}	OPT.
43e.^{0}	5	16	10^{-6}		36	17	54.0	10^{-14}	10^{-11}	10^{-27}	OPT.
			10^{-10}		36	17	54.0	10^{-14}	10^{-11}	10^{-27}	OPT.
43f.^{0}	5	16	10^{-6}		87	36	54.0	10^{-10}	10^{-8}	10^{-21}	OPT.
			10^{-10}		87	36	54.0	10^{-10}	10^{-8}	10^{-21}	OPT.
44a.^{0}	6	6	10^{-6}		181	55	4.03	10^{-14}	10^{-13}	10^{-27}	OPT.
			10^{-10}		181	55	4.03	10^{-14}	10^{-13}	10^{-27}	OPT.
44b.^{0}	6	6	10^{-6}		49	14	3.52	10^{-15}	10^{-13}	10^{-30}	OPT.
			10^{-10}		49	14	3.52	10^{-15}	10^{-13}	10^{-30}	OPT.
44c.^{0}	6	6	10^{-6}		908	337	20.6	10^{-13}	10^{-9}	10^{-26}	OPT.
			10^{-10}		909	338	20.6	10^{-13}	10^{-9}	10^{-26}	*
44d.^{0}	6	6	10^{-6}		917	330	15.3	10^{-10}	10^{-7}	10^{-21}	OPT.
			10^{-10}		918	331	15.3	10^{-15}	10^{-11}	10^{-29}	OPT.
44e.^{0}	6	6	10^{-6}		501	156	9.27	10^{-11}	10^{-8}	10^{-21}	OPT.
			10^{-10}		502	157	9.27	10^{-11}	10^{-8}	10^{-21}	*
45a.^{0}	8	8	10^{-6}		170	49	4.06	10^{-16}	10^{-14}	10^{-32}	OPT.
			10^{-10}		170	49	4.06	10^{-16}	10^{-14}	10^{-32}	OPT.
45b.^{0}	8	8	10^{-6}		31	12	3.56	10^{-15}	10^{-13}	10^{-29}	*
			10^{-10}		31	12	3.56	10^{-15}	10^{-13}	10^{-29}	*
45c.^{0}	8	8	10^{-6}		1380	541	20.6	10^{-11}	10^{-7}	10^{-21}	OPT.
			10^{-10}		1381	542	20.6	10^{-14}	10^{-11}	10^{-28}	OPT.
45d.^{0}	8	8	10^{-6}		1431	520	15.3	10^{-11}	10^{-8}	10^{-22}	OPT.
			10^{-10}		1432	521	15.3	10^{-11}	10^{-8}	10^{-22}	*
45e.^{0}	8	8	10^{-6}		1512	510	9.31	10^{-10}	10^{-8}	10^{-20}	OPT.
			10^{-10}		1513	511	9.31	10^{-14}	10^{-12}	10^{-28}	OPT.

2.6.2 Quasi-Newton (BFGS) Linesearch Method (SOL/NAG NPSOL)

2.6.2.1 Software and Algorithm

The results were obtained using subroutine NPSOL from the Systems Optimization Laboratory (SOL), Stanford University, also available in the NAG Library. In NPSOL a search direction is determined at each iteration from a subproblem of the form

$$\min_{p \in \Re^n} \bar{g}_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H_k p,$$

where the Hessian matrix H_k is calculated using the BFGS method initialized with I (see Section 2.5.2). This is followed by a linesearch that uses both function and gradient information to obtain a steplength along the search direction [Gill et al. (1979)].

2.6.2.2 Parameters

Parameters were kept at their default values with the following exceptions † :

Linesearch Tolerance	-	0.5
Iteration Limit	-	9999
Optimality Tolerance	-	varied; see tables

† For unconstrained optimization with NPSOL, variable bounds were set to the default value of Infinite Step Size (10^{10}). In some cases overflow occurred during function evaluation in the linesearch. These cases are indicated in the tables by giving the value of bounds on the variables that was subsequently used to obtain the results in the column labeled "Var Bnd".

See Gill et al. [1986] for details concerning the parameters.

2.6.2.3 Convergence Criteria

The following quantities will be used in describing the convergence criteria :

objective function	:	$\mathcal{F}_k \ (= \frac{1}{2} f_k^{\mathrm{T}} f_k)$
objective gradient	:	$\bar{g}_k = \nabla \mathcal{F}_k \ (= J_k^{\mathrm{T}} f_k)$
optimality tolerance	:	ϵ_{opt}

The sequence of iterates generated by NPSOL is judged to have converged if the following two conditions hold :

$$\alpha_k \|p_k\|_2 \leq \sqrt{\epsilon_{opt}}(1 + \|x_k\|_2) \tag{2.6.6}$$

and

$$\|g_k\|_2 \leq \sqrt{\epsilon_{opt}}(1 + \max\{(1 + |\mathcal{F}_k|)\|g_k\|_2\}) \tag{2.6.7}$$

or if

$$\|g_k\|_2 \leq \epsilon_M^{0.8}(1 + \max\{(1 + |\mathcal{F}_k|), \|g_k\|_2\}). \tag{2.6.8}$$

Condition (2.6.6) is meant to ensure that the sequence $\{x_k\}$ has converged, while conditions (2.6.7) and (2.6.8) are intended to test whether the requirement that the gradient vanish is approximately satisfied at x_k. Condition (2.6.8) allows NPSOL to accept a point as a local mimimum if a more restrictive test on the necessary condition than (2.6.7) is satisfied, but condition (2.6.6) does not hold. For a detailed discussion of convergence criteria similar to these, see Section 8.2 of Gill, Murray, and Wright [1981].

The following abbreviations are used in the tables to describe the conditions under which the algorithm terminates : †

OPT.	-	optimal point found
*	-	current point cannot be improved
**	-	optimal solution found, but requested accuracy could not be achieved
F LIM.	-	function evaluation limit reached

† A '*' corresponds to the situation in which the algorithm terminates due to failure in the linesearch to find an acceptable step at the current iteration. A '**' occurs when condition (2.6.7) is satisfied but not condition (2.6.6); that is, conditions for optimality are met at the current point but the iterates have not yet converged.

Numerical Results for NPSOL

	n	m	Opt Tol	Var Bnd	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-10}		57	29	1.41	10^{-10}	10^{-9}	10^{-20}	OPT.
			10^{-14}		57	29	1.41	10^{-10}	10^{-9}	10^{-20}	OPT.
2.^{0}	2	2	10^{-10}		14	7	11.4	10^{1}	10^{-4}	10^{1}	OPT.
			10^{-14}		16	9	11.4	10^{1}	10^{-7}	10^{1}	OPT.
3.^{0}	2	2	10^{-10}		(2000)	(1000)	3.18	10^{-2}	10^{0}	10^{-3}	F LIM.
			10^{-14}		(2000)	(1000)	3.18	10^{-2}	10^{0}	10^{-3}	F LIM.
4.^{0}	2	3	10^{-10}		(2002)	(632)	10^{6}	10^{5}	10^{9}	10^{10}	F LIM.
			10^{-14}		(2002)	(632)	10^{6}	10^{5}	10^{9}	10^{10}	F LIM.
5.^{0}	2	3	10^{-10}		23	12	3.04	10^{-8}	10^{-8}	10^{-16}	OPT.
			10^{-14}		24	13	3.04	10^{-10}	10^{-10}	10^{-21}	OPT.
6.	2	10	10^{-10}	10^{4}	5	1	104.	10^{1}	10^{-31}	10^{1}	OPT.
			10^{-14}	10^{4}	5	1	104.	10^{1}	10^{-31}	10^{1}	OPT.
7.^{0}	3	3	10^{-10}		38	24	1.00	10^{-7}	10^{-6}	10^{-13}	OPT.
			10^{-14}		40	26	1.00	10^{-11}	10^{-10}	10^{-21}	OPT.
8.	3	15	10^{-10}		30	14	2.60	10^{-1}	10^{-9}	10^{-8}	OPT.
			10^{-14}		31	15	2.60	10^{-1}	10^{-11}	10^{-8}	OPT.
9.	3	15	10^{-10}		8	3	1.08	10^{-4}	10^{-13}	10^{-14}	OPT.
			10^{-14}		8	3	1.08	10^{-4}	10^{-13}	10^{-14}	OPT.
10.	3	16	10^{-10}		1	1	10^{3}	10^{4}	10^{10}	10^{7}	*
			10^{-14}		1	1	10^{3}	10^{4}	10^{10}	10^{7}	*
11.^{0}	3	10	10^{-10}		2	1	8.39	10^{-1}	0.00	10^{-2}	OPT.
			10^{-14}		2	1	8.39	10^{-1}	0.00	10^{-2}	OPT.
12.^{0}	3	10	10^{-10}		22	8	6.69	10^{-1}	10^{-1}	10^{-2}	*
			10^{-14}		22	8	6.69	10^{-1}	10^{-1}	10^{-2}	*
13.^{0}	4	4	10^{-10}		86	50	10^{-5}	10^{-9}	10^{-9}	10^{-17}	OPT.
			10^{-14}		86	50	10^{-5}	10^{-9}	10^{-9}	10^{-17}	OPT.
14.^{0}	4	6	10^{-10}		90	41	2.00	10^{-8}	10^{-7}	10^{-16}	OPT.
			10^{-14}		91	42	2.00	10^{-10}	10^{-9}	10^{-20}	OPT.
15.	4	11	10^{-10}		33	18	.328	10^{-2}	10^{-8}	10^{-9}	OPT.
			10^{-14}		35	20	.328	10^{-2}	10^{-12}	10^{-9}	OPT.
16.	4	20	10^{-10}		42	16	17.6	10^{2}	10^{-4}	10^{-8}	OPT.
			10^{-14}		43	17	17.6	10^{2}	10^{-5}	10^{-8}	OPT.
17.	5	33	10^{-10}	1.0	48	23	1.17	10^{-2}	10^{-3}	10^{-3}	OPT.
			10^{-14}	1.0	49	24	1.17	10^{-2}	10^{-3}	10^{-3}	OPT.
18.^{0}	6	13	10^{-10}		48	32	18.7	10^{-1}	10^{-10}	10^{-2}	OPT.
			10^{-14}		48	32	18.7	10^{-1}	10^{-10}	10^{-2}	OPT.
19.	11	65	10^{-10}		81	47	9.38	10^{-1}	10^{-6}	10^{-8}	OPT.
			10^{-14}		83	49	9.38	10^{-1}	10^{-8}	10^{-8}	OPT.

Numerical Results for NPSOL

	n	m	Opt Tol	Var Bnd	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-10}		53	25	2.44	10^{-2}	10^{-7}	10^{-10}	OPT.
			10^{-14}		54	26	2.44	10^{-2}	10^{-9}	10^{-10}	OPT.
20b.	9	31	10^{-10}		94	47	6.06	10^{-3}	10^{-7}	10^{-13}	OPT.
			10^{-14}		98	49	6.06	10^{-3}	10^{-10}	10^{-13}	OPT.
20c.	12	31	10^{-10}		86	43	1.28	10^{-4}	10^{-5}	10^{-8}	OPT.
			10^{-14}		251	123	1.33	10^{-4}	10^{-8}	10^{-8}	OPT.
20d.	20	31	10^{-10}		76	38	1.06	10^{-4}	10^{-5}	10^{-8}	OPT.
			10^{-14}		200	99	1.06	10^{-5}	10^{-8}	10^{-11}	OPT.
21a.^{0}	10	10	10^{-10}		138	63	3.16	10^{-8}	10^{-7}	10^{-15}	OPT.
			10^{-14}		139	64	3.16	10^{-9}	10^{-8}	10^{-17}	OPT.
21b.^{0}	20	20	10^{-10}		210	98	4.47	10^{-6}	10^{-5}	10^{-11}	OPT.
			10^{-14}		218	102	4.47	10^{-9}	10^{-7}	10^{-17}	OPT.
22a.^{0}	12	12	10^{-10}		133	75	10^{-3}	10^{-7}	10^{-7}	10^{-13}	OPT.
			10^{-14}		171	97	10^{-4}	10^{-8}	10^{-8}	10^{-16}	OPT.
22b.^{0}	20	20	10^{-10}		193	112	10^{-4}	10^{-7}	10^{-8}	10^{-15}	OPT.
			10^{-14}		232	135	10^{-4}	10^{-8}	10^{-8}	10^{-15}	OPT.
23a.	4	5	10^{-10}		73	37	.500	10^{-3}	10^{-8}	10^{-10}	OPT.
			10^{-14}		75	39	.500	10^{-3}	10^{-11}	10^{-10}	OPT.
23b.	10	11	10^{-10}		212	96	.500	10^{-2}	10^{-9}	10^{-11}	OPT.
			10^{-14}		225	102	.500	10^{-2}	10^{-12}	10^{-11}	OPT.
24a.	4	8	10^{-10}		21	12	.597	10^{-3}	10^{-6}	10^{-6}	OPT.
			10^{-14}		920	548	.759	10^{-3}	10^{-10}	10^{-11}	OPT.
24b.	10	20	10^{-10}		350	188	.598	10^{-2}	10^{-7}	10^{-9}	OPT.
			10^{-14}		368	200	.598	10^{-2}	10^{-10}	10^{-9}	OPT.
25a.^{0}	10	12	10^{-10}		26	20	3.16	10^{-12}	10^{-11}	10^{-24}	OPT.
			10^{-14}		26	20	3.16	10^{-12}	10^{-11}	10^{-24}	OPT.
25b.^{0}	20	22	10^{-10}		31	13	4.47	10^{-12}	10^{-11}	10^{-24}	OPT.
			10^{-14}		32	14	4.47	10^{-12}	10^{-11}	10^{-24}	**
26a.^{0}	10	10	10^{-10}		37	25	.328	10^{-2}	10^{-8}	10^{-5}	OPT.
			10^{-14}		39	27	.328	10^{-2}	10^{-10}	10^{-5}	OPT.
26b.^{0}	20	20	10^{-10}		86	49	.231	10^{-3}	10^{-6}	10^{-5}	OPT.
			10^{-14}		95	55	.231	10^{-3}	10^{-11}	10^{-5}	OPT.
27a.^{0}	10	10	10^{-10}		19	10	3.16	10^{-8}	10^{-7}	10^{-15}	OPT.
			10^{-14}		20	12	3.16	10^{-11}	10^{-10}	10^{-22}	OPT.
27b.^{0}	20	20	10^{-10}	10.0	18	10	4.47	10^{-6}	10^{-5}	10^{-11}	OPT.
			10^{-14}	10.0	21	12	4.47	10^{-11}	10^{-10}	10^{-21}	OPT.
28a.^{0}	10	10	10^{-10}		30	15	.412	10^{-7}	10^{-7}	10^{-14}	OPT.
			10^{-14}		33	17	.412	10^{-9}	10^{-9}	10^{-18}	OPT.
28b.^{0}	20	20	10^{-10}		54	30	.571	10^{-6}	10^{-7}	10^{-13}	OPT.
			10^{-14}		60	34	.571	10^{-10}	10^{-10}	10^{-19}	OPT.

Numerical Results for NPSOL

	n	m	Opt Tol	Var Bnd	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
29a.^{0}	10	10	10^{-10}		7	5	.412	10^{-8}	10^{-8}	10^{-15}	OPT.
			10^{-14}		8	6	.412	10^{-9}	10^{-9}	10^{-18}	OPT.
29b.^{0}	20	20	10^{-10}		7	5	.571	10^{-8}	10^{-8}	10^{-15}	OPT.
			10^{-14}		8	6	.571	10^{-9}	10^{-9}	10^{-18}	OPT.
30a.^{0}	10	10	10^{-10}		42	18	2.05	10^{-6}	10^{-5}	10^{-12}	OPT.
			10^{-14}		44	20	2.05	10^{-9}	10^{-8}	10^{-18}	OPT.
30b.^{0}	20	20	10^{-10}		61	28	3.04	10^{-7}	10^{-6}	10^{-14}	OPT.
			10^{-14}		62	29	3.04	10^{-8}	10^{-7}	10^{-16}	OPT.
31a.^{0}	10	10	10^{-10}		47	22	1.80	10^{-6}	10^{-6}	10^{-13}	OPT.
			10^{-14}		49	24	1.80	10^{-9}	10^{-8}	10^{-18}	OPT.
31b.^{0}	20	20	10^{-10}		76	30	2.66	10^{-7}	10^{-6}	10^{-13}	OPT.
			10^{-14}		78	32	2.66	10^{-9}	10^{-8}	10^{-17}	OPT.
32.^{L}	10	20	10^{-10}		2	1	3.16	10^{0}	10^{-15}	0.00	OPT.
			10^{-14}		2	1	3.16	10^{0}	10^{-15}	0.00	OPT.
33.^{L}	10	20	10^{-10}		4	2	1.46	10^{0}	10^{-11}	10^{-6}	OPT.
			10^{-14}		4	2	1.46	10^{0}	10^{-11}	10^{-6}	OPT.
34.^{L}	10	20	10^{-10}		4	2	1.78	10^{0}	10^{-11}	10^{-6}	OPT.
			10^{-14}		4	2	1.78	10^{0}	10^{-11}	10^{-6}	OPT.
35a.	8	8	10^{-10}		33	19	1.65	10^{-1}	10^{-5}	10^{-9}	OPT.
			10^{-14}		35	21	1.65	10^{-1}	10^{-7}	10^{-9}	OPT.
35b.^{0}	9	9	10^{-10}		29	16	1.73	10^{-6}	10^{-5}	10^{-11}	OPT.
			10^{-14}		33	19	1.73	10^{-8}	10^{-9}	10^{-16}	OPT.
35c.	10	10	10^{-10}		37	21	1.81	10^{-1}	10^{-7}	10^{-3}	OPT.
			10^{-14}		40	23	1.81	10^{-1}	10^{-7}	10^{-3}	OPT.
36a.^{0}	4	4	10^{-10}		786	477	9.22	10^{-5}	10^{-5}	10^{-9}	OPT.
			10^{-14}		2618	1437	10.1	10^{-5}	10^{-7}	10^{-9}	OPT.
36b.^{0}	9	9	10^{-10}		932	618	9.54	10^{-5}	10^{-5}	10^{-9}	OPT.
			10^{-14}		2153	1272	9.75	10^{-5}	10^{-7}	10^{-9}	OPT.
36c.^{0}	9	9	10^{-10}		3	1	1.73	0.00	0.00	0.00	OPT.
			10^{-14}		3	1	1.73	0.00	0.00	0.00	OPT.
36d.^{0}	9	9	10^{-10}		929	557	9.63	10^{-5}	10^{-5}	10^{-9}	OPT.
			10^{-14}		3265	1820	9.95	10^{-5}	10^{-8}	10^{-9}	OPT.
37.	2	16	10^{-10}		16	7	8.85	10^{1}	10^{-4}	10^{-6}	OPT.
			10^{-14}		18	9	8.85	10^{1}	10^{-7}	10^{-6}	OPT.
38.	3	16	10^{-10}	1.0	36	11	1.01	10^{1}	10^{2}	10^{1}	OPT.
			10^{-14}	1.0	37	12	1.01	10^{1}	10^{2}	10^{1}	OPT.

Numerical Results for NPSOL

	n	m	Opt Tol	Var Bnd	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
39a.	2	3	10^{-10}		8	5	10^{-6}	10^{-1}	10^{-14}	10^{-7}	OPT.
			10^{-14}		8	5	10^{-6}	10^{-1}	10^{-14}	10^{-7}	OPT.
39b.	2	3	10^{-10}		9	8	10^{-7}	10^{-1}	10^{-7}	10^{-7}	OPT.
			10^{-14}		10	9	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
39c.	2	3	10^{-10}		6	5	10^{-7}	10^{-1}	10^{-7}	10^{-7}	OPT.
			10^{-14}		8	7	10^{-7}	10^{-1}	10^{-16}	10^{-7}	OPT.
39d.	2	3	10^{-10}		11	6	10^{-7}	10^{-1}	10^{-7}	10^{-7}	OPT.
			10^{-14}		12	7	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
39e.	2	3	10^{-10}		20	11	10^{-8}	10^{-1}	10^{-6}	10^{-7}	OPT.
			10^{-14}		21	12	10^{-8}	10^{-1}	10^{-8}	10^{-7}	OPT.
39f.	2	3	10^{-10}		26	10	10^{-8}	10^{-1}	10^{-6}	10^{-7}	OPT.
			10^{-14}		27	11	10^{-9}	10^{-1}	10^{-8}	10^{-7}	OPT.
39g.	2	3	10^{-10}		28	13	10^{-9}	10^{-1}	10^{-6}	10^{-7}	OPT.
			10^{-14}		29	14	10^{-10}	10^{-1}	10^{-8}	10^{-7}	OPT.
40a.	3	4	10^{-10}		13	7	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-14}		14	8	10^{-6}	10^{0}	10^{-13}	10^{-7}	OPT.
40b.	3	4	10^{-10}		10	6	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
			10^{-14}		11	7	10^{-6}	10^{0}	10^{-13}	10^{-7}	OPT.
40c.	3	4	10^{-10}		9	7	10^{-7}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-14}		10	8	10^{-7}	10^{0}	10^{-11}	10^{-7}	OPT.
40d.	3	4	10^{-10}		15	8	10^{-7}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-14}		16	9	10^{-7}	10^{0}	10^{-12}	10^{-7}	OPT.
40e.	3	4	10^{-10}		23	12	10^{-6}	10^{0}	10^{-6}	10^{-7}	OPT.
			10^{-14}		25	14	10^{-7}	10^{0}	10^{-9}	10^{-7}	OPT.
40f.	3	4	10^{-10}		40	14	10^{-8}	10^{0}	10^{-6}	10^{-7}	OPT.
			10^{-14}		41	15	10^{-8}	10^{0}	10^{-8}	10^{-7}	OPT.
40g.	3	4	10^{-10}		45	17	10^{-9}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-14}		45	17	10^{-9}	10^{0}	10^{-8}	10^{-7}	OPT.
41a.	5	10	10^{-10}		12	7	10^{-6}	10^{0}	10^{-11}	10^{-7}	OPT.
			10^{-14}		12	7	10^{-6}	10^{0}	10^{-11}	10^{-7}	OPT.
41b.	5	10	10^{-10}		12	7	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-14}		13	8	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
41c.	5	10	10^{-10}		12	7	10^{-6}	10^{0}	10^{-7}	10^{-7}	OPT.
			10^{-14}		14	9	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
41d.	5	10	10^{-10}		17	9	10^{-6}	10^{0}	10^{-7}	10^{-7}	OPT.
			10^{-14}		19	11	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
41e.	5	10	10^{-10}		47	19	10^{-6}	10^{0}	10^{-5}	10^{-7}	OPT.
			10^{-14}		50	22	10^{-7}	10^{0}	10^{-8}	10^{-7}	OPT.
41f.	5	10	10^{-10}		73	30	10^{-7}	10^{0}	10^{-5}	10^{-7}	OPT.
			10^{-14}		76	33	10^{-7}	10^{0}	10^{-8}	10^{-7}	OPT.
41g.	5	10	10^{-10}		83	32	10^{-8}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-14}		84	33	10^{-8}	10^{0}	10^{-12}	10^{-7}	OPT.

Numerical Results for NPSOL

	n	m	Opt Tol	Var Bnd	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-10}	0.9	3	1	0.9	10^2	10^{-13}	10^5	OPT.
			10^{-14}	0.9	3	1	0.9	10^2	10^{-13}	10^5	OPT.
42b.^{0}	4	24	10^{-10}	0.9	3	1	0.9	10^2	10^{-13}	10^5	OPT.
			10^{-14}	0.9	3	1	0.9	10^2	10^{-13}	10^5	OPT.
42c.^{0}	4	24	10^{-10}	5.0	18	9	7.96	10^2	10^3	10^4	OPT.
			10^{-14}	5.0	19	10	7.96	10^2	10^3	10^4	OPT.
42d.^{0}	4	24	10^{-10}	5.0	21	12	7.96	10^2	10^3	10^4	OPT.
			10^{-14}	5.0	22	13	7.96	10^2	10^3	10^4	OPT.
43a.^{0}	5	16	10^{-10}	10.0	20	13	17.4	10^2	10^2	10^3	OPT.
			10^{-14}	10.0	21	14	17.4	10^2	10^2	10^3	OPT.
43b.^{0}	5	16	10^{-10}	10.0	18	11		17.4	10^2	10^2	10^3 OPT.
			10^{-14}	10.0	19	12		17.4	10^2	10^2	10^3 OPT.
43c.^{0}	5	16	10^{-10}	10.0	20	13	17.4	10^2	10^2	10^3	OPT.
			10^{-14}	10.0	22	15	17.4	10^2	10^2	10^3	OPT.
43d.^{0}	5	16	10^{-10}	10.0	20	12	17.4	10^2	10^2	10^3	OPT.
			10^{-14}	10.0	21	13	17.4	10^2	10^2	10^3	OPT.
43e.^{0}	5	16	10^{-10}	10.0	20	12	17.4	10^2	10^2	10^3	OPT.
			10^{-14}	10.0	21	13	17.4	10^2	10^2	10^3	OPT.
43f.^{0}	5	16	10^{-10}	10.0	18	11		17.4	10^2	10^2	10^3 OPT.
			10^{-14}	10.0	19	12		17.4	10^2	10^2	10^3 OPT.
44a.^{0}	6	6	10^{-10}		379	237	4.03	10^{-7}	10^{-5}	10^{-14}	OPT.
			10^{-14}		388	243	4.03	10^{-8}	10^{-9}	10^{-15}	OPT.
44b.^{0}	6	6	10^{-10}		25	15	3.52	10^{-6}	10^{-5}	10^{-12}	OPT.
			10^{-14}		27	17	3.52	10^{-9}	10^{-7}	10^{-17}	OPT.
44c.^{0}	6	6	10^{-10}		(6000)	(3010)	23.6	10^{-1}	10^0	10^{-1}	F LIM.
			10^{-14}		(6000)	(3010)	23.6	10^{-1}	10^0	10^{-1}	F LIM.
44d.^{0}	6	6	10^{-10}		(6000)	(2994)	15.9	10^{-2}	10^0	10^{-4}	F LIM.
			10^{-14}		(6000)	(2994)	15.9	10^{-2}	10^0	10^{-4}	F LIM.
44e.^{0}	6	6	10^{-10}		3430	1714	9.27	10^{-6}	10^{-6}	10^{-12}	OPT.
			10^{-14}		3586	1805	9.27	10^{-7}	10^{-7}	10^{-14}	OPT.
45a.^{0}	8	8	10^{-10}		307	197	4.06	10^{-7}	10^{-6}	10^{-14}	OPT.
			10^{-14}		312	200	4.06	10^{-12}	10^{-10}	10^{-23}	OPT.
45b.^{0}	8	8	10^{-10}		40	21	3.56	10^{-7}	10^{-6}	10^{-14}	OPT.
			10^{-14}		41	22	3.56	10^{-8}	10^{-7}	10^{-16}	OPT.
45c.^{0}	8	8	10^{-10}		(6001)	(2979)	20.3	10^{-1}	10^0	10^{-2}	F LIM.
			10^{-14}		(6001)	(2979)	20.3	10^{-1}	10^0	10^{-2}	F LIM.
45d.^{0}	8	8	10^{-10}		(6001)	(2989)	18.3	10^{-1}	10^0	10^{-2}	F LIM.
			10^{-14}		(6001)	(2989)	18.3	10^{-1}	10^0	10^{-2}	F LIM.
45e.^{0}	8	8	10^{-10}		2821	1398	9.31	10^{-6}	10^{-5}	10^{-12}	OPT.
			10^{-14}		3147	1583	9.31	10^{-9}	10^{-7}	10^{-17}	OPT.

2.6.3.3 Convergence Criteria

The following quantities will be used in describing the convergence criteria :

objective function	:	$\mathcal{F}_k \ (= \frac{1}{2} f_k^T f_k)$
objective gradient	:	$g_k = \nabla \mathcal{F}_k \ (= J_k^T f_k)$
current step	:	p_k, the minimizer of the subproblem
Newton step	:	$p_N \begin{cases} H_k^{-1} g & \text{if } H_k \text{ is positive definite;} \\ \textit{undefined} & \text{otherwise.} \end{cases}$
Newton reduction	:	$\rho_N = \begin{cases} -\mathcal{Q}_k(p_N) & \text{if } H_k \text{ is positive definite;} \\ 0 & \text{otherwise.} \end{cases}$
predicted reduction	:	$\rho_P = -\mathcal{Q}_k(p_k)$
actual reduction	:	$\rho_A = \mathcal{F}_k - \mathcal{F}(x_k + p_k)$
scaled distance	:	$\nu(x, y, D) = \dfrac{\max_{1 \le i \le n} \{\lvert (D(x-y))_i \rvert\}}{\max_{1 \le i \le n} \{\lvert (Dx)_i \rvert + \lvert (Dy)_i \rvert\}}$. †

† Here v_i denotes the ith component of the vector v. There is a provision for the user to replace the function ν; we used the default in all of the tests.

The convergence criteria used in DMNH and DMNG are as follows :

- *Absolute function convergence* occurs at x_k if

$$|\mathcal{F}_k| < \texttt{V(AFCTOL)}. \tag{2.6.9}$$

- *Relative function convergence* is intended to approximate the condition

$$\mathcal{F}_k - \mathcal{F}(x^*) \le \texttt{V(RFCTOL)}\, |\mathcal{F}_k| .$$

The test actually used is

$$\rho_N \le \texttt{V(RFCTOL)}\, |\mathcal{F}_k| . \tag{2.6.10}$$

- x *convergence* is intended to approximate the condition

$$\nu(x_k, x^*, D_k) \le \texttt{V(XCTOL)},$$

The test actually used is

$$p_k = p_N \ \text{ and } \ \nu(x_k, x_k + p_k, D_k) \le \texttt{V(XCTOL)}. \tag{2.6.11}$$

- *Singular convergence* is intended to approximate the condition

$$\mathcal{F}_k - \min \{\mathcal{F}(y) \mid \|D_k(y - x_k)\| \le \texttt{V(LMAXS)}\} < \texttt{V(SCTOL)}\, |\mathcal{F}_k| ,$$

where D_k is the diagonal scaling matrix at the kth iterate. The test for singular convergence is made only when none of the convergence criteria listed above holds. It is meant to indicate relative function convegence when the Hessian in the subproblem is singular.
The actual test is

$$\mathcal{F}_k - \min \{\mathcal{Q}_k(y) \mid \|D_k(y - x_k)\| \leq \texttt{V(LMAXS)}\} < \texttt{V(SCTOL)}\,|\mathcal{F}_k|\,. \qquad (2.6.12)$$

Under certain conditions, the test (2.6.12) is repeated for a step of length V(LMAXS).

- *False convergence* is returned if none of the other convergence criteria is satisfied and a trial step no larger than V(XFCTOL) is rejected. This usually indicates either an error in computing the objective gradient, a discontinuity (in $\mathcal{F}$ or g) near the current iterate, or that one or more of the convergence tolerances (V(RFCTOL), V(XCTOL), and V(AFCTOL)) are too small relative to the accuracy to which the objective is computed.

The test actually used is

$$\mathcal{F}_k - \mathcal{F}(x_k + p_k) \leq \texttt{V(TUNER1)}\rho_P \quad \text{and} \quad \nu(x_k, x_k + p_k, D_k) \leq \texttt{V(XFTOL)}, \qquad (2.6.13)$$

where the parameter V(TUNER1) is adjustable, although in these tests the default value 0.1 is used throughout.

Except for (2.6.9), tests for convergence are performed only when

$$\rho_A \leq 2\rho_P. \qquad (2.6.14)$$

See Dennis, Gay, and Welsch [1981a, 1981b], Gay [1983], and PORT [1984] for more discussion of the convergence criteria.

The following abbreviations are used in the tables to describe the conditions under which the algorithm terminates :

ABS. F	-	(2.6.9)
REL. F	-	(2.6.10) and (2.6.14)
X	-	(2.6.11) and (2.6.14)
X, F	-	(2.6.10) and (2.6.11) and (2.6.14)
SING.	-	(2.6.12) and (2.6.14)
FALSE	-	(2.6.13) and (2.6.14)
F LIM.	-	function evaluation limit reached
TIME	-	time limit exceeded
LOOP	-	subroutine appears to loop

The total number of Jacobian evaluations is either equal to the total number of iterations of the method, or it is one more than the number of iterations. The number in the column labeled "iters. / J evals." is followed by a "+" if an extra Jacobian evaluation was used in the computation.

Numerical Results for DMNH

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-8}		32	24+	1.41	10^{-14}	10^{-13}	10^{-28}	ABS. F
			10^{-12}		32	24+	1.41	10^{-14}	10^{-13}	10^{-28}	ABS. F
2.^{0}	2	2	10^{-8}		10	7+	11.4	10^{1}	10^{-11}	10^{1}	REL. F
			10^{-12}		10	7+	11.4	10^{1}	10^{-11}	10^{1}	REL. F
3.^{0}	2	2	10^{-8}		130	101+	9.11	10^{-9}	10^{-4}	10^{-17}	ABS. F
			10^{-12}		132	103+	9.11	10^{-16}	10^{-11}	10^{-31}	ABS. F
4.^{0}	2	3	10^{-8}		22	10+	10^{6}	10^{-11}	10^{-5}	10^{-22}	X
			10^{-12}		23	11+	10^{6}	10^{-22}	10^{-22}	10^{-44}	X
5.^{0}	2	3	10^{-8}		11	8+	3.04	10^{-10}	10^{-10}	10^{-20}	ABS. F
			10^{-12}		12	9+	3.04	$10^{0.00}$	$10^{0.00}$	$10^{0.00}$	ABS. F
6.	2	10	10^{-8}		11	10+	.365	10^{1}	10^{-9}	10^{-6}	REL. F
			10^{-12}		11	10+	.365	10^{1}	10^{-9}	10^{-6}	REL. F
7.^{0}	3	3	10^{-8}		16	12+	1.00	10^{-8}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		17	13+	1.00	10^{-16}	10^{-16}	10^{-33}	ABS. F
8.	3	15	10^{-8}		9	8+	2.60	10^{-1}	10^{-12}	10^{-8}	REL. F
			10^{-12}		10	9	2.60	10^{-1}	10^{-12}	10^{-8}	REL. F
9.	3	15	10^{-8}		4	3+	1.08	10^{-4}	10^{-16}	10^{-14}	X, REL. F
			10^{-12}		4	3+	1.08	10^{-4}	10^{-16}	10^{-14}	REL. F
10.	3	16	10^{-8}		387	244+	10^{4}	10^{1}	10^{-1}	10^{-6}	REL. F
			10^{-12}		388	245	10^{4}	10^{1}	10^{-1}	10^{-6}	REL. F
11.^{0}	3	10	10^{-8}		290	167+	55.9	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		292	169+	55.9	10^{-16}	10^{-16}	10^{-32}	ABS. F
12.^{0}	3	10	10^{-8}		24	19+	10.1	10^{-13}	10^{-13}	10^{-26}	ABS. F
			10^{-12}		24	19+	10.1	10^{-13}	10^{-13}	10^{-26}	ABS. F
13.^{0}	4	4	10^{-8}		27	26+	10^{-4}	10^{-8}	10^{-12}	10^{-16}	ABS. F
			10^{-12}		38	37+	10^{-6}	10^{-12}	10^{-17}	10^{-24}	ABS. F
14.^{0}	4	6	10^{-8}		42	32+	2.00	10^{-12}	10^{-11}	10^{-24}	ABS. F
			10^{-12}		49	38+	2.00	0.00	0.00	0.00	ABS. F
15.	4	11	10^{-8}		11	8+	.328	10^{-2}	10^{-13}	10^{-9}	REL. F
			10^{-12}		12	9	.328	10^{-2}	10^{-13}	10^{-9}	REL. F
16.	4	20	10^{-8}		11	9+	17.6	10^{2}	10^{-6}	10^{-8}	REL. F
			10^{-12}		13	11	17.6	10^{2}	10^{-11}	10^{-8}	X, REL. F
17.	5	33	10^{-8}	0.2	46	32+	2.46	10^{-2}	10^{-7}	10^{-11}	REL. F
			10^{-12}	0.2	47	33+	2.46	10^{-2}	10^{-12}	10^{-11}	REL. F
18.^{0}	6	13	10^{-8}		(6000)	(1824+)	283.	10^{-1}	10^{-5}	10^{-1}	F LIM.
			10^{-12}		(6000)	(1820+)	275.	10^{-1}	10^{-9}	10^{-1}	F LIM.
19.	11	65	10^{-8}		23	17+	9.38	10^{-1}	10^{-9}	10^{-8}	REL. F
			10^{-12}		24	18+	9.38	10^{-1}	10^{-10}	10^{-8}	REL. F

Numerical Results for DMNH

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-8}		15	14	2.44	10^{-2}	10^{-15}	10^{-10}	X, REL. F
			10^{-12}		15	14	2.44	10^{-2}	10^{-15}	10^{-10}	X, REL. F
20b.	9	31	10^{-8}		20	16+	6.06	10^{-3}	10^{-13}	10^{-13}	REL. F
			10^{-12}		22	18	6.06	10^{-3}	10^{-14}	10^{-13}	X, REL. F
20c.	12	31	10^{-8}		24	19	16.6	10^{-5}	10^{-13}	10^{-16}	X, REL. F
			10^{-12}		24	19	16.6	10^{-5}	10^{-13}	10^{-16}	X, REL. F
20d.	20	31	10^{-8}		50	26+	1.10	10^{-8}	10^{-13}	10^{-16}	ABS. F
			10^{-12}		(149)	(55+)	1.16	10^{-8}	10^{-13}	10^{-16}	LOOP
21a.^{0}	10	10	10^{-8}		25	22+	3.16	10^{-9}	10^{-8}	10^{-19}	ABS. F
			10^{-12}		26	23+	3.16	10^{-16}	10^{-14}	10^{-30}	ABS. F
21b.^{0}	20	20	10^{-8}		27	23+	4.47	10^{-14}	10^{-13}	10^{-28}	ABS. F
			10^{-12}		27	23+	4.47	10^{-14}	10^{-13}	10^{-28}	ABS. F
22a.^{0}	12	12	10^{-8}		28	27+	10^{-4}	10^{-8}	10^{-12}	10^{-16}	ABS. F
			10^{-12}		40	39+	10^{-6}	10^{-12}	10^{-18}	10^{-24}	ABS. F
22b.^{0}	20	20	10^{-8}		29	28+	10^{-4}	10^{-8}	10^{-12}	10^{-17}	ABS. F
			10^{-12}		40	39+	10^{-6}	10^{-12}	10^{-18}	10^{-24}	ABS. F
23a.	4	5	10^{-8}		42	36+	.500	10^{-3}	10^{-12}	10^{-10}	REL. F
			10^{-12}		43	37	.500	10^{-3}	10^{-12}	10^{-10}	REL. F
23b.	10	11	10^{-8}		44	37+	.500	10^{-2}	10^{-9}	10^{-11}	REL. F
			10^{-12}		45	38+	.500	10^{-2}	10^{-14}	10^{-11}	REL. F
24a.	4	8	10^{-8}		126	110+	.759	10^{-3}	10^{-7}	10^{-11}	REL. F
			10^{-12}		128	112+	.759	10^{-3}	10^{-13}	10^{-11}	REL. F
24b.	10	20	10^{-8}		158	106+	.598	10^{-2}	10^{-7}	10^{-9}	REL. F
			10^{-12}		162	110	.598	10^{-2}	10^{-16}	10^{-9}	X, REL. F
25a.^{0}	10	12	10^{-8}		15	14+	3.16	10^{-13}	10^{-12}	10^{-26}	ABS. F
			10^{-12}		15	14+	3.16	10^{-13}	10^{-12}	10^{-26}	ABS. F
25b.^{0}	20	22	10^{-8}		18	17+	4.47	10^{-10}	10^{-8}	10^{-20}	ABS. F
			10^{-12}		19	18	4.47	10^{-15}	10^{-13}	10^{-30}	X
26a.^{0}	10	10	10^{-8}		11	9+	.328	10^{-2}	10^{-11}	10^{-5}	REL. F
			10^{-12}		12	10	.328	10^{-2}	10^{-11}	10^{-5}	REL. F
26b.^{0}	20	20	10^{-8}		20	16	.228	10^{-3}	10^{-11}	10^{-6}	X, REL. F
			10^{-12}		20	16	.228	10^{-3}	10^{-11}	10^{-6}	X, REL. F
27a.^{0}	10	10	10^{-8}		9	7+	3.16	10^{-10}	10^{-10}	10^{-21}	ABS. F
			10^{-12}		10	8+	3.16	10^{-15}	10^{-14}	10^{-29}	ABS. F
27b.^{0}	20	20	10^{-8}		11	9+	4.47	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		12	11+	4.47	10^{-14}	10^{-12}	10^{-27}	ABS. F
28a.^{0}	10	10	10^{-8}		4	3+	.412	10^{-12}	10^{-13}	10^{-24}	ABS. F
			10^{-12}		4	3+	.412	10^{-12}	10^{-13}	10^{-24}	ABS. F
28b.^{0}	20	20	10^{-8}		4	3+	.571	10^{-13}	10^{-14}	10^{-25}	ABS. F
			10^{-12}		4	3+	.571	10^{-13}	10^{-14}	10^{-25}	ABS. F

Numerical Results for DMNH

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
29a.^{0}	10	10	10^{-8}		4	3+	.412	10^{-10}	10^{-10}	10^{-20}	ABS. F
			10^{-12}		5	4+	.412	10^{-16}	10^{-16}	10^{-32}	ABS. F
29b.^{0}	20	20	10^{-8}		4	3+	.571	10^{-10}	10^{-10}	10^{-20}	ABS. F
			10^{-12}		5	4+	.571	10^{-16}	10^{-16}	10^{-32}	ABS. F
30a.^{0}	10	10	10^{-8}		6	5+	2.05	10^{-8}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		7	6+	2.05	10^{-15}	10^{-15}	10^{-31}	ABS. F
30b.^{0}	20	20	10^{-8}		6	5+	3.04	10^{-8}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		7	6+	3.04	10^{-15}	10^{-15}	10^{-30}	ABS. F
31a.^{0}	10	10	10^{-8}		9	8+	1.80	10^{-13}	10^{-12}	10^{-26}	ABS. F
			10^{-12}		9	8+	1.80	10^{-13}	10^{-12}	10^{-26}	ABS. F
31b.^{0}	20	20	10^{-8}		9	8+	1.80	10^{-13}	10^{-12}	10^{-26}	ABS. F
			10^{-12}		9	8+	1.80	10^{-13}	10^{-12}	10^{-26}	ABS. F
32.^{L}	10	20	10^{-8}		6	4	3.16	10^{0}	10^{-32}	0.00	X, REL. F
			10^{-12}		6	4	3.16	10^{0}	10^{-32}	0.00	X, REL. F
33.^{L}	10	20	10^{-8}		5	4	1.46	10^{0}	10^{-11}	10^{-6}	SING.
			10^{-12}		5	4	1.46	10^{0}	10^{-11}	10^{-6}	SING.
34.^{L}	10	20	10^{-8}		6	5	1.78	10^{0}	10^{-11}	10^{-6}	SING.
			10^{-12}		6	5	1.78	10^{0}	10^{-11}	10^{-6}	SING.
35a.	8	8	10^{-8}		14	11	1.65	10^{-1}	10^{-9}	10^{-9}	X, REL. F
			10^{-12}		14	11	1.65	10^{-1}	10^{-9}	10^{-9}	REL. F
35b.^{0}	9	9	10^{-8}		17	12+	1.73	10^{-9}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		18	13+	1.73	10^{-15}	10^{-14}	10^{-29}	ABS. F
35c.	10	10	10^{-8}		19	11+	1.76	10^{-1}	10^{-9}	10^{-9}	REL. F
			10^{-12}		20	12	1.76	10^{-1}	10^{-9}	10^{-9}	REL. F
36a.^{0}	4	4	10^{-8}		(4000)	(2190)	17.8	10^{-6}	10^{-6}	10^{-12}	F LIM.
			10^{-12}		(4000)	(2190)	17.8	10^{-6}	10^{-6}	10^{-12}	F LIM.
36b.^{0}	9	9	10^{-8}		(1832)	(611)	9.15	10^{-5}	10^{-9}	10^{-9}	TIME
			10^{-12}		(1881)	(625)	9.24	10^{-5}	10^{-5}	10^{-9}	TIME
36c.^{0}	9	9	10^{-8}		31	30+	1.73	10^{-8}	10^{-10}	10^{-17}	ABS. F
			10^{-12}		35	33+	1.73	10^{-13}	10^{-16}	10^{-25}	ABS. F
36d.^{0}	9	9	10^{-8}		(1900)	(638)	9.38	10^{-5}	10^{-5}	10^{-9}	TIME
			10^{-12}		(1859)	(627)	9.32	10^{-5}	10^{-5}	10^{-9}	TIME
37.	2	16	10^{-8}		16	7+	9.05	10^{1}	10^{-7}	10^{-1}	REL. F
			10^{-12}		17	8	9.05	10^{1}	10^{-7}	10^{-1}	REL. F
38.	3	16	10^{-8}		14	8+	26.1	10^{1}	10^{-10}	10^{-6}	REL. F
			10^{-12}		14	8+	26.1	10^{1}	10^{-10}	10^{-6}	REL. F

Numerical Results for DMNH

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
39a.	2	3	10^{-8}		4	3+	10^{-6}	10^{-1}	10^{-17}	10^{-7}	REL. F
			10^{-12}		4	3+	10^{-6}	10^{-1}	10^{-17}	10^{-7}	REL. F
39b.	2	3	10^{-8}		4	3+	10^{-7}	10^{-1}	10^{-11}	10^{-7}	REL. F
			10^{-12}		4	3+	10^{-7}	10^{-1}	10^{-11}	10^{-7}	REL. F
39c.	2	3	10^{-8}		4	3+	10^{-7}	10^{-1}	10^{-13}	10^{-7}	REL. F
			10^{-12}		5	4	10^{-7}	10^{-1}	10^{-13}	10^{-7}	REL. F
39d.	2	3	10^{-8}		6	5+	10^{-7}	10^{-1}	10^{-8}	10^{-7}	REL. F
			10^{-12}		6	5+	10^{-7}	10^{-1}	10^{-8}	10^{-7}	REL. F
39e.	2	3	10^{-8}		8	7+	10^{-8}	10^{-1}	10^{-14}	10^{-7}	REL. F
			10^{-12}		8	7+	10^{-8}	10^{-1}	10^{-14}	10^{-7}	REL. F
39f.	2	3	10^{-8}		11	10	10^{-9}	10^{-1}	10^{-8}	10^{-7}	REL. F
			10^{-12}		11	10	10^{-9}	10^{-1}	10^{-8}	10^{-7}	REL. F
39g.	2	3	10^{-8}		13	12+	10^{-10}	10^{-1}	10^{-9}	10^{-7}	REL. F
			10^{-12}		14	13	10^{-10}	10^{-1}	10^{-9}	10^{-7}	REL. F
40a.	2	3	10^{-8}		4	3+	10^{-6}	10^{-1}	10^{-15}	10^{-7}	REL. F
			10^{-12}		4	3+	10^{-6}	10^{-1}	10^{-15}	10^{-7}	REL. F
40b.	3	4	10^{-8}		4	3+	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
			10^{-12}		5	4	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
40c.	3	4	10^{-8}		5	4+	10^{-7}	10^{0}	10^{-12}	10^{-7}	REL. F
			10^{-12}		5	4+	10^{-7}	10^{0}	10^{-12}	10^{-7}	REL. F
40d.	3	4	10^{-8}		5	4+	10^{-7}	10^{0}	10^{-8}	10^{-7}	REL. F
			10^{-12}		6	5+	10^{-7}	10^{0}	10^{-16}	10^{-7}	REL. F
40e.	3	4	10^{-8}		7	6+	10^{-7}	10^{0}	10^{-8}	10^{-7}	REL. F
			10^{-12}		7	6+	10^{-7}	10^{0}	10^{-8}	10^{-7}	REL. F
40f.	3	4	10^{-8}		10	9+	10^{-8}	10^{0}	10^{-11}	10^{-7}	REL. F
			10^{-12}		10	9+	10^{-8}	10^{0}	10^{-11}	10^{-7}	REL. F
40g.	3	4	10^{-8}		13	12+	10^{-9}	10^{0}	10^{-15}	10^{-7}	REL. F
			10^{-12}		13	12+	10^{-9}	10^{0}	10^{-15}	10^{-7}	REL. F
41a.	5	10	10^{-8}		4	3	10^{-6}	10^{0}	10^{-9}	10^{-7}	REL. F
			10^{-12}		4	3	10^{-6}	10^{0}	10^{-9}	10^{-7}	REL. F
41b.	5	10	10^{-8}		4	3+	10^{-6}	10^{0}	10^{-13}	10^{-7}	REL. F
			10^{-12}		4	3+	10^{-6}	10^{0}	10^{-13}	10^{-7}	REL. F
41c.	5	10	10^{-8}		8	7+	10^{-6}	10^{0}	10^{-12}	10^{-7}	REL. F
			10^{-12}		8	7+	10^{-6}	10^{0}	10^{-12}	10^{-7}	REL. F
41d.	5	10	10^{-8}		8	7+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
			10^{-12}		9	8+	10^{-6}	10^{0}	10^{-14}	10^{-7}	REL. F
41e.	5	10	10^{-8}		11	10+	10^{-7}	10^{0}	10^{-10}	10^{-7}	REL. F
			10^{-12}		12	11	10^{-7}	10^{0}	10^{-10}	10^{-7}	REL. F
41f.	5	10	10^{-8}		14	13	10^{-7}	10^{0}	10^{-11}	10^{-7}	REL. F
			10^{-12}		14	13	10^{-7}	10^{0}	10^{-11}	10^{-7}	REL. F
41g.	5	10	10^{-8}		17	16+	10^{-8}	10^{0}	10^{-12}	10^{-7}	REL. F
			10^{-12}		17	16+	10^{-8}	10^{0}	10^{-12}	10^{-7}	REL. F

Numerical Results for DMNH

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-8}		28	20+	60.8	10^{-12}	10^{-10}	10^{-25}	ABS. F
			10^{-12}		28	20+	60.8	10^{-12}	10^{-10}	10^{-25}	ABS. F
42b.^{0}	4	24	10^{-8}		35	26+	61.9	10^{-10}	10^{-8}	10^{-21}	ABS. F
			10^{-12}		36	27+	61.9	10^{-12}	10^{-10}	10^{-25}	X
42c.^{0}	4	24	10^{-8}		30	22+	60.3	10^{-12}	10^{-10}	10^{-23}	ABS. F
			10^{-12}		31	23+	60.3	10^{-13}	10^{-10}	10^{-26}	X
42d.^{0}	4	24	10^{-8}		30	21+	60.3	10^{-14}	10^{-11}	10^{-27}	X
			10^{-12}		30	21+	60.3	10^{-14}	10^{-11}	10^{-27}	ABS. F
43a.^{0}	5	16	10^{-8}		22	16+	54.0	10^{-14}	10^{-11}	10^{-27}	X
			10^{-12}		22	16+	54.0	10^{-14}	10^{-11}	10^{-27}	ABS. F
43b.^{0}	5	16	10^{-8}		26	20+	54.0	10^{-9}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		27	21+	54.0	10^{-14}	10^{-12}	10^{-28}	ABS. F
43c.^{0}	5	16	10^{-8}		21	17+	54.0	10^{-13}	10^{-11}	10^{-26}	X
			10^{-12}		21	17+	54.0	10^{-13}	10^{-11}	10^{-26}	ABS. F
43d.^{0}	5	16	10^{-8}	0.9	27	18+	54.0	10^{-9}	10^{-7}	10^{-18}	ABS. F
			10^{-12}	0.9	28	19+	54.0	10^{-14}	10^{-11}	10^{-27}	ABS. F
43e.^{0}	5	16	10^{-8}	0.9	28	20+	54.0	10^{-10}	10^{-8}	10^{-20}	ABS. F
			10^{-12}	0.9	29	21+	54.0	10^{-14}	10^{-12}	10^{-28}	ABS. F
43f.^{0}	5	16	10^{-8}		17	14+	54.0	10^{-9}	10^{-7}	10^{-18}	ABS. F
			10^{-12}		18	15+	54.0	10^{-14}	10^{-12}	10^{-28}	ABS. F
44a.^{0}	6	6	10^{-8}		179	150+	4.03	10^{-11}	10^{-9}	10^{-22}	ABS. F
			10^{-12}		180	151+	4.03	10^{-16}	10^{-15}	10^{-33}	ABS. F
44b.^{0}	6	6	10^{-8}		9	7+	3.52	10^{-10}	10^{-9}	10^{-19}	ABS. F
			10^{-12}		10	8+	3.52	10^{-15}	10^{-13}	10^{-30}	ABS. F
44c.^{0}	6	6	10^{-8}		194	179+	20.6	10^{-10}	10^{-7}	10^{-20}	ABS. F
			10^{-12}		195	180+	20.6	10^{-14}	10^{-10}	10^{-28}	ABS. F
44d.^{0}	6	6	10^{-8}		187	179+	15.3	10^{-9}	10^{-6}	10^{-17}	ABS. F
			10^{-12}		188	180+	15.3	10^{-14}	10^{-10}	10^{-27}	ABS. F
44e.^{0}	6	6	10^{-8}		219	210+	9.27	10^{-8}	10^{-6}	10^{-16}	ABS. F
			10^{-12}		220	211+	9.27	10^{-13}	10^{-10}	10^{-26}	ABS. F
45a.^{0}	8	8	10^{-8}		63	49+	4.06	10^{-8}	10^{-6}	10^{-16}	ABS. F
			10^{-12}		64	50+	4.06	10^{-14}	10^{-12}	10^{-27}	ABS. F
45b.^{0}	8	8	10^{-8}		15	11+	3.56	10^{-8}	10^{-7}	10^{-16}	ABS. F
			10^{-12}		16	12+	3.56	10^{-15}	10^{-14}	10^{-30}	ABS. F
45c.^{0}	8	8	10^{-8}		321	300+	20.6	10^{-9}	10^{-6}	10^{-18}	ABS. F
			10^{-12}		322	301+	20.6	10^{-14}	10^{-10}	10^{-28}	ABS. F
45d.^{0}	8	8	10^{-8}		328	292+	15.3	10^{-11}	10^{-7}	10^{-21}	ABS. F
			10^{-12}		329	293+	15.3	10^{-16}	10^{-13}	10^{-31}	ABS. F
45e.^{0}	8	8	10^{-8}		351	288+	9.31	10^{-11}	10^{-9}	10^{-23}	ABS. F
			10^{-12}		352	289+	9.31	10^{-15}	10^{-12}	10^{-29}	ABS. F

Numerical Results for DMNG

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-8}		40	35+	1.41	10^{-9}	10^{-8}	10^{-19}	ABS. F
			10^{-12}		42	37+	1.41	10^{-14}	10^{-14}	10^{-29}	X
2.^{0}	2	2	10^{-8}		12	10+	11.4	10^{1}	10^{-7}	10^{1}	REL. F
			10^{-12}		12	10+	11.4	10^{1}	10^{-7}	10^{1}	REL. F
3.^{0}	2	2	10^{-8}		217	160+	9.11	10^{-9}	10^{-4}	10^{-16}	ABS. F
			10^{-12}		220	162+	9.11	10^{-12}	10^{-7}	10^{-25}	ABS. F
4.^{0}	2	3	10^{-8}		66	11+	10^{6}	10^{-8}	10^{-2}	10^{-15}	X
			10^{-12}		67	12+	10^{6}	10^{-11}	10^{-8}	10^{-22}	X
5.^{0}	2	3	10^{-8}		16	14+	3.04	10^{-11}	10^{-10}	10^{-22}	ABS. F
			10^{-12}		17	15+	3.04	10^{-14}	10^{-13}	10^{-28}	ABS. F
6.	2	10	10^{-8}		33	20+	.365	10^{-1}	10^{-4}	10^{-6}	REL. F
			10^{-12}		34	21+	.365	10^{-1}	10^{-7}	10^{-6}	REL. F
7.^{0}	3	3	10^{-8}		28	23+	1.00	10^{-10}	10^{-8}	10^{-19}	ABS. F
			10^{-12}		30	25+	1.00	10^{-14}	10^{-13}	10^{-28}	ABS. F
8.	3	15	10^{-8}		19	17+	2.60	10^{-1}	10^{-7}	10^{-8}	REL. F
			10^{-12}		22	20	2.60	10^{-1}	10^{-12}	10^{-8}	REL. F
9.	3	15	10^{-8}		8	5+	1.08	10^{-4}	10^{-11}	10^{-14}	X, REL. F
			10^{-12}		12	9	1.08	10^{-4}	10^{-17}	10^{-14}	REL. F
10.	3	16	10^{-8}		465	325+	10^{4}	10^{1}	10^{0}	10^{-6}	REL. F
			10^{-12}		467	327	10^{4}	10^{1}	10^{-1}	10^{-6}	REL. F
11.^{0}	3	10	10^{-8}		4	3+	5.66	10^{-1}	10^{-6}	10^{-2}	REL. F
			10^{-12}		327	267+	55.9	10^{-13}	10^{-13}	10^{-26}	ABS. F
12.^{0}	3	10	10^{-8}		43	34+	10.1	10^{-9}	10^{-10}	10^{-18}	ABS. F
			10^{-12}		45	36+	10.1	10^{-14}	10^{-14}	10^{-28}	ABS. F
13.^{0}	4	4	10^{-8}		62	61+	10^{-4}	10^{-8}	10^{-9}	10^{-16}	ABS. F
			10^{-12}		89	88+	10^{-6}	10^{-12}	10^{-13}	10^{-24}	ABS. F
14.^{0}	4	6	10^{-8}		100	78+	2.00	10^{-8}	10^{-7}	10^{-16}	ABS. F
			10^{-12}		102	80+	2.00	10^{-12}	10^{-11}	10^{-24}	ABS. F
15.	4	11	10^{-8}		35	31+	.328	10^{-2}	10^{-8}	10^{-9}	REL. F
			10^{-12}		36	32+	.328	10^{-2}	10^{-11}	10^{-9}	REL. F
16.	4	20	10^{-8}		46	34+	17.6	10^{2}	10^{-2}	10^{-8}	REL. F
			10^{-12}		47	35+	17.6	10^{2}	10^{-5}	10^{-8}	REL. F
17.	5	33	10^{-8}		69	55+	2.46	10^{-2}	10^{-5}	10^{-11}	REL. F
			10^{-12}		72	58+	2.46	10^{-2}	10^{-9}	10^{-11}	REL. F
18.^{0}	6	13	10^{-8}		45	41+	18.7	10^{-1}	10^{-7}	10^{-2}	REL. F
			10^{-12}		47	43+	18.7	10^{-1}	10^{-10}	10^{-2}	REL. F
19.	11	65	10^{-8}		69	58+	9.38	10^{-1}	10^{-6}	10^{-8}	REL. F
			10^{-12}		72	61+	9.38	10^{-1}	10^{-8}	10^{-8}	REL. F

Numerical Results for DMNG

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-8}		35	32+	2.44	10^{-2}	10^{-6}	10^{-10}	REL. F
			10^{-12}		37	34+	2.44	10^{-2}	10^{-8}	10^{-10}	REL. F
20b.	9	31	10^{-8}		76	69+	6.06	10^{-3}	10^{-10}	10^{-13}	REL. F
			10^{-12}		79	74	6.06	10^{-3}	10^{-10}	10^{-13}	REL. F
20c.	12	31	10^{-8}		89	85+	1.28	10^{-4}	10^{-9}	10^{-8}	REL. F
			10^{-12}		148	143	2.87	10^{-5}	10^{-13}	10^{-9}	REL. F
20d.	20	31	10^{-8}		110	107+	1.06	10^{-6}	10^{-11}	10^{-12}	X
			10^{-12}		134	119	1.06	10^{-7}	10^{-12}	10^{-13}	FALSE
21a.^{0}	10	10	10^{-8}		120	98+	3.16	10^{-8}	10^{-7}	10^{-17}	ABS. F
			10^{-12}		125	103+	3.16	10^{-13}	10^{-12}	10^{-27}	X
21b.^{0}	20	20	10^{-8}		189	148+	4.47	10^{-9}	10^{-8}	10^{-17}	X
			10^{-12}		193	152+	4.47	10^{-13}	10^{-12}	10^{-25}	ABS. F
22a.^{0}	12	12	10^{-8}		143	131+	10^{-4}	10^{-8}	10^{-11}	10^{-16}	ABS. F
			10^{-12}		235	219+	10^{-6}	10^{-12}	10^{-17}	10^{-24}	ABS. F
22b.^{0}	20	20	10^{-8}		187	153+	10^{-4}	10^{-8}	10^{-8}	10^{-16}	ABS. F
			10^{-12}		344	311+	10^{-6}	10^{-12}	10^{-17}	10^{-24}	ABS. F
23a.	4	5	10^{-8}		77	57+	.500	10^{-3}	10^{-10}	10^{-10}	REL. F
			10^{-12}		78	58+	.500	10^{-3}	10^{-11}	10^{-10}	REL. F
23b.	10	11	10^{-8}		80	67+	.500	10^{-2}	10^{-9}	10^{-11}	REL. F
			10^{-12}		81	68+	.500	10^{-2}	10^{-11}	10^{-11}	REL. F
24a.	4	8	10^{-8}		364	270+	.769	10^{-3}	10^{-7}	10^{-9}	REL. F
			10^{-12}		472	355+	.759	10^{-3}	10^{-11}	10^{-11}	REL. F
24b.	10	20	10^{-8}		475	367+	.606	10^{-2}	10^{-5}	10^{-8}	REL. F
			10^{-12}		632	510+	.598	10^{-2}	10^{-9}	10^{-9}	REL. F
25a.^{0}	10	12	10^{-8}		20	19+	3.16	10^{-12}	10^{-10}	10^{-23}	ABS. F
			10^{-12}		21	20+	3.16	0.00	0.00	0.00	X
25b.^{0}	20	22	10^{-8}		25	24+	4.47	10^{-9}	10^{-7}	10^{-17}	ABS. F
			10^{-12}		26	25+	4.47	10^{-15}	10^{-13}	10^{-29}	ABS. F
26a.^{0}	10	10	10^{-8}		34	32+	.328	10^{-2}	10^{-8}	10^{-5}	REL. F
			10^{-12}		37	35+	.328	10^{-2}	10^{-10}	10^{-5}	REL. F
26b.^{0}	20	20	10^{-8}		62	59+	.231	10^{-3}	10^{-8}	10^{-5}	REL. F
			10^{-12}		65	62	.231	10^{-3}	10^{-10}	10^{-5}	REL. F
27a.^{0}	10	10	10^{-8}		13	11+	3.16	10^{-8}	10^{-7}	10^{-16}	ABS. F
			10^{-12}		16	14+	3.16	10^{-12}	10^{-12}	10^{-24}	ABS. F
27b.^{0}	20	20	10^{-8}		15	12+	4.47	10^{-8}	10^{-7}	10^{-16}	ABS. F
			10^{-12}		18	15+	4.47	10^{-13}	10^{-12}	10^{-27}	ABS. F
28a.^{0}	10	10	10^{-8}		31	25+	.412	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		34	28+	.412	10^{-14}	10^{-14}	10^{-28}	ABS. F
28b.^{0}	20	20	10^{-8}		60	48+	.571	10^{-8}	10^{-8}	10^{-16}	ABS. F
			10^{-12}		64	52+	.571	10^{-13}	10^{-13}	10^{-26}	ABS. F

Numerical Results for DMNG

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
29a.^{0}	10	10	10^{-8}		8	7+	.412	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		10	9+	.412	10^{-13}	10^{-13}	10^{-25}	ABS. F
29b.^{0}	20	20	10^{-8}		8	7+	.571	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		10	9+	.571	10^{-13}	10^{-13}	10^{-25}	ABS. F
30a.^{0}	10	10	10^{-8}		51	39+	2.05	10^{-8}	10^{-7}	10^{-15}	X
			10^{-12}		57	45+	2.05	10^{-13}	10^{-12}	10^{-25}	X
30b.^{0}	20	20	10^{-8}		65	45+	3.04	10^{-8}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		88	61+	3.04	10^{-12}	10^{-11}	10^{-24}	ABS. F
31a.^{0}	10	10	10^{-8}		46	26+	1.80	10^{-8}	10^{-8}	10^{-16}	ABS. F
			10^{-12}		60	35+	1.80	10^{-12}	10^{-12}	10^{-25}	ABS. F
31b.^{0}	20	20	10^{-8}		47	26+	2.66	10^{-8}	10^{-7}	10^{-16}	ABS. F
			10^{-12}		63	36	2.66	10^{-12}	10^{-11}	10^{-23}	X
32.^{L}	10	20	10^{-8}		6	4	3.16	10^{0}	10^{-16}	0.00	X, REL. F
			10^{-12}		6	4	3.16	10^{0}	10^{-16}	0.00	X, REL. F
33.^{L}	10	20	10^{-8}		4	2	1.46	10^{0}	10^{-8}	10^{-6}	X, REL. F
			10^{-12}		4	2	1.46	10^{0}	10^{-8}	10^{-6}	X, REL. F
34.^{L}	10	20	10^{-8}		5	3	1.78	10^{0}	10^{-12}	10^{-6}	X, REL. F
			10^{-12}		5	3	1.78	10^{0}	10^{-12}	10^{-6}	X, REL. F
35a.	8	8	10^{-8}		34	24	1.65	10^{-1}	10^{-5}	10^{-9}	REL. F
			10^{-12}		38	27+	1.65	10^{-1}	10^{-8}	10^{-9}	REL. F
35b.^{0}	9	9	10^{-8}		44	32+	1.73	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		46	34+	1.73	10^{-12}	10^{-12}	10^{-24}	ABS. F
35c.	10	10	10^{-8}		41	31+	1.81	10^{-1}	10^{-6}	10^{-3}	REL. F
			10^{-12}		45	36+	1.81	10^{-1}	10^{-8}	10^{-3}	REL. F
36a.^{0}	4	4	10^{-8}		(4000)	(2891)	17.0	10^{-6}	10^{-6}	10^{-11}	F LIM.
			10^{-12}		(4000)	(2891)	17.0	10^{-6}	10^{-6}	10^{-11}	F LIM.
36b.^{0}	9	9	10^{-8}		(9000)	(6426)	228.	10^{-6}	10^{-7}	10^{-12}	F LIM.
			10^{-12}		(9000)	(6426)	228.	10^{-6}	10^{-7}	10^{-12}	F LIM.
36c.^{0}	9	9	10^{-8}		69	64+	1.73	10^{-8}	10^{-9}	10^{-16}	ABS. F
			10^{-12}		101	96+	1.73	10^{-12}	10^{-13}	10^{-24}	ABS. F
36d.^{0}	9	9	10^{-8}		(9000)	(6486)	228.	10^{-6}	10^{-7}	10^{-12}	F LIM.
			10^{-12}		(9000)	(6486)	228.	10^{-6}	10^{-7}	10^{-12}	. LIM.
37.	2	16	10^{-8}		22	10+	9.05	10^{1}	10^{-6}	10^{-1}	REL. F
			10^{-12}		22	10+	9.05	10^{1}	10^{-6}	10^{-1}	REL. F
38.	3	16	10^{-8}		31	17+	26.1	10^{1}	10^{-4}	10^{-6}	REL. F
			10^{-12}		32	18+	26.1	10^{1}	10^{-5}	10^{-6}	REL. F

Numerical Results for DMNG

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
39a.	2	3	10^{-8}		9	8+	10^{-6}	10^{-1}	10^{-6}	10^{-7}	REL. F
			10^{-12}		11	10+	10^{-6}	10^{-1}	10^{-10}	10^{-7}	REL. F
39b.	2	3	10^{-8}		9	8+	10^{-7}	10^{-1}	10^{-7}	10^{-7}	REL. F
			10^{-12}		10	9+	10^{-7}	10^{-1}	10^{-9}	10^{-7}	REL. F
39c.	2	3	10^{-8}		6	5+	10^{-7}	10^{-1}	10^{-7}	10^{-7}	REL. F
			10^{-12}		7	6+	10^{-7}	10^{-1}	10^{-12}	10^{-7}	REL. F
39d.	2	3	10^{-8}		8	6+	10^{-8}	10^{-1}	10^{-8}	10^{-7}	REL. F
			10^{-12}		9	7+	10^{-7}	10^{-1}	10^{-9}	10^{-7}	REL. F
39e.	2	3	10^{-8}		11	8+	10^{-7}	10^{-1}	10^{-6}	10^{-7}	REL. F
			10^{-12}		12	9+	10^{-8}	10^{-1}	10^{-8}	10^{-7}	REL. F
39f.	2	3	10^{-8}		11	8+	10^{-9}	10^{-1}	10^{-8}	10^{-7}	REL. F
			10^{-12}		11	8+	10^{-9}	10^{-1}	10^{-8}	10^{-7}	REL. F
39g.	2	3	10^{-8}		17	13+	10^{-9}	10^{-1}	10^{-6}	10^{-7}	REL. F
			10^{-12}		18	14+	10^{-10}	10^{-1}	10^{-8}	10^{-7}	REL. F
40a.	3	4	10^{-8}		11	10+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
			10^{-12}		12	11+	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
40b.	3	4	10^{-8}		10	9+	10^{-6}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		12	11+	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
40c.	3	4	10^{-8}		9	7+	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
			10^{-12}		10	8+	10^{-7}	10^{0}	10^{-11}	10^{-7}	REL. F
40d.	3	4	10^{-8}		10	7+	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
			10^{-12}		11	8+	10^{-7}	10^{0}	10^{-9}	10^{-7}	REL. F
40e.	3	4	10^{-8}		11	8+	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		13	10+	10^{-7}	10^{0}	10^{-9}	10^{-7}	REL. F
40f.	3	4	10^{-8}		14	11+	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		16	13+	10^{-8}	10^{0}	10^{-8}	10^{-7}	REL. F
40g.	3	4	10^{-8}		18	14+	10^{-8}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		20	16+	10^{-9}	10^{0}	10^{-6}	10^{-7}	REL. F
41a.	5	10	10^{-8}		11	8+	10^{-6}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		13	10+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
41b.	5	10	10^{-8}		11	8+	10^{-6}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		13	10+	10^{-6}	10^{0}	10^{-9}	10^{-7}	REL. F
41c.	5	10	10^{-8}		13	9+	10^{-6}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		14	10+	10^{-6}	10^{0}	10^{-7}	10^{-7}	REL. F
41d.	5	10	10^{-8}		17	12+	10^{-5}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		20	15+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
41e.	5	10	10^{-8}		24	20+	10^{-6}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		26	22+	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
41f.	5	10	10^{-8}		27	22+	10^{-6}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		31	26+	10^{-7}	10^{0}	10^{-6}	10^{-7}	REL. F
41g.	5	10	10^{-8}		32	26+	10^{-6}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		35	29+	10^{-8}	10^{0}	10^{-8}	10^{-7}	REL. F

Numerical Results for DMNG

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-8}		53	41+	60.8	10^{-8}	10^{-6}	10^{-16}	x
			10^{-12}		56	43+	60.8	10^{-13}	10^{-11}	10^{-25}	x
42b.^{0}	4	24	10^{-8}	0.9	103	69+	61.9	10^{-10}	10^{-8}	10^{-20}	x
			10^{-12}	0.9	104	70+	61.9	10^{-13}	10^{-10}	10^{-25}	x
42c.^{0}	4	24	10^{-8}		76	52+	60.3	10^{-9}	10^{-6}	10^{-17}	x
			10^{-12}		78	54+	60.3	10^{-13}	10^{-11}	10^{-26}	x
42d.^{0}	4	24	10^{-8}		61	48+	60.3	10^{-7}	10^{-5}	10^{-14}	x
			10^{-12}		64	51+	60.3	10^{-12}	10^{-10}	10^{-24}	x
43a.^{0}	5	16	10^{-8}		49	34+	54.0	10^{-9}	10^{-7}	10^{-18}	x
			10^{-12}		51	36+	54.0	10^{-12}	10^{-10}	10^{-24}	x
43b.^{0}	5	16	10^{-8}		58	37+	54.0	10^{-9}	10^{-7}	10^{-18}	x
			10^{-12}		60	39+	54.0	10^{-14}	10^{-12}	10^{-27}	x
43c.^{0}	5	16	10^{-8}		41	29+	54.0	10^{-9}	10^{-6}	10^{-17}	x
			10^{-12}		44	32+	54.0	10^{-13}	10^{-12}	10^{-26}	x
43d.^{0}	5	16	10^{-8}		57	44+	54.0	10^{-9}	10^{-7}	10^{-17}	ABS. F
			10^{-12}		60	47+	54.0	10^{-13}	10^{-11}	10^{-26}	x
43e.^{0}	5	16	10^{-8}		51	41+	54.0	10^{-9}	10^{-8}	10^{-18}	x
			10^{-12}		53	43+	54.0	10^{-14}	10^{-11}	10^{-27}	x
43f.^{0}	5	16	10^{-8}		45	36+	54.0	10^{-8}	10^{-6}	10^{-16}	x
			10^{-12}		48	39+	54.0	10^{-13}	10^{-10}	10^{-25}	x
44a.^{0}	6	6	10^{-8}		441	341+	4.03	10^{-9}	10^{-8}	10^{-19}	ABS. F
			10^{-12}		444	344+	4.03	10^{-12}	10^{-11}	10^{-25}	ABS. F
44b.^{0}	6	6	10^{-8}		31	24+	3.52	10^{-9}	10^{-8}	10^{-18}	x
			10^{-12}		34	27+	3.52	10^{-14}	10^{-12}	10^{-26}	x
44c.^{0}	6	6	10^{-8}		3726	2748+	20.6	10^{-8}	10^{-5}	10^{-16}	ABS. F
			10^{-12}		3731	2753+	20.6	10^{-12}	10^{-10}	10^{-25}	ABS. F
44d.^{0}	6	6	10^{-8}		2752	1928+	15.3	10^{-1}	10^{0}	10^{-2}	REL. F
			10^{-12}		3865	2915+	15.3	10^{-12}	10^{-9}	10^{-24}	x
44e.^{0}	6	6	10^{-8}		2104	1550+	12.3	10^{-1}	10^{0}	10^{-2}	REL. F
			10^{-12}		2813	2098+	9.27	10^{-13}	10^{-10}	10^{-25}	ABS. F
45a.^{0}	8	8	10^{-8}		284	227+	4.06	10^{-8}	10^{-7}	10^{-16}	ABS. F
			10^{-12}		288	231+	4.06	10^{-12}	10^{-11}	10^{-25}	ABS. F
45b.^{0}	8	8	10^{-8}		36	28+	3.56	10^{-8}	10^{-7}	10^{-17}	ABS. F
			10^{-12}		40	32+	3.56	10^{-13}	10^{-12}	10^{-26}	x
45c.^{0}	8	8	10^{-8}		6197	4538+	20.6	10^{-8}	10^{-5}	10^{-16}	ABS. F
			10^{-12}		6200	4541+	20.6	10^{-12}	10^{-9}	10^{-24}	x
45d.^{0}	8	8	10^{-8}		7929	5976+	15.3	10^{-8}	10^{-5}	10^{-16}	ABS. F
			10^{-12}		7934	5981+	15.3	10^{-14}	10^{-11}	10^{-28}	ABS. F
45e.^{0}	8	8	10^{-8}		3341	2511+	9.31	10^{-9}	10^{-7}	10^{-17}	ABS. F
			10^{-12}		3346	2517+	9.31	10^{-13}	10^{-11}	10^{-25}	ABS. F

Dennis, J. E., D. M. Gay, and R. E. Welsch, "ALGORITHM 573 NL2SOL : An Adaptive Nonlinear Least-Squares Algorithm", ACM *Transactions on Mathematical Software*, Vol. 7 No. 3 (September 1981b) 369–383.

Dennis, J. E., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall (1983).

Flachs, J., "On the Generation of Updates for Quasi-Newton Methods", *Journal of Optimization Theory and Applications*, Vol. 48 No. 3 (March 1986) 379–417.

Fletcher, R., and T. L. Freeman, "A Modified Newton Method for Minimization", *Journal of Optimization Theory and Applications*, Vol. 23 No. 3 (November 1977) 357–372.

Fletcher, R., *Practical Methods of Optimization : Volume I, Unconstrained Optimization*, Wiley (1980).

Gay, D. M., "Computing Optimal Locally Constrained Steps", SIAM *Journal on Scientific and Statistical Computing*, Vol. 2 No. 2 (June 1981) 186–197.

Gay, D. M., "ALGORITHM 611 : Subroutines for Unconstrained Minimization Using a Model/Trust-Region Approach", ACM *Transactions on Mathematical Software*, Vol. 9 No. 4 (December 1983) 503–524.

Ge R.-P., and M. J. D. Powell, "The Convergence of Variable Metric Matrices in Unconstrained Optimization", *Mathematical Programming* 27 (1983) 123–143.

Gill. P. E., and W. Murray, "Quasi-Newton Methods for Unconstrained Optimization", *Journal of the Institute of Mathematics and its Applications* 9 (1972) 91–108.

Gill. P. E., and W. Murray, "Newton-type Methods for Unconstrained and Linearly Constrained Optimization", *Mathematical Programming* 7 (1974a) 311–350.

Gill, P. E., and W. Murray, "Safeguarded Steplength Algorithms for Optimization using Descent Methods", Technical Report NAC 37, National Physical Laboratory, England (1974b).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Two Steplength Algorithms for Numerical Optimization", Technical Report SOL 79-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University (1979).

Gill, P. E., W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press (1981).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Users Guide for NPSOL (Version 4.0) : A Fortran Package for Nonlinear Programming", Technical Report SOL 86-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University (January 1986).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Model Building and Practical Aspects of Nonlinear Programming", in in NATO ASI Series, Vol. **F15**, *Computational Mathematical Programming*, K. Schittkowski (ed.), Springer-Verlag (1985) 209–247.

Goldfarb, D., "Curvilinear Path Steplength Algorithms for Minimization which use Directions of Negative Curvature", *Mathematical Programming* **18** (1980) 31–40.

Goldfeld, S., R. Quandt, and H. Trotter, "Maximization by Quadratic Hill-Climbing", *Econometrica* **34** (1966) 541–551.

Goldstein, A. A., "On Steepest Descent", SIAM *Journal on Control* **3** (1965) 147–151.

Goldstein, A. A., *Constructive Real Analysis*, Harper and Row (1967).

Goldstein, A. A., and J. F. Price, "An Effective Algorithm for Minimization", *Numerische Mathematik* **10** (1967) 184–189.

Gourgeon, H., and J. Nocedal, "A Conic Algorithm for Optimization", SIAM *Journal on Scientific and Statistical Computing*, Vol. **6** No. 2 (April 1985) 253–267.

Grandinetti, L., "Some Investigations of a New Algorithm for Nonlinear Optimization Based on Conic Models", *Journal of Optimization Theory and Applications*, Vol. **43** No. 1 (May 1984) 1–22.

Greenstadt, J. L., "On the Relative Efficiencies of Gradient Methods", *Mathematics of Computation* **21** (1967) 360–367.

Hebden, M. D., "An Algorithm for Minimization using Exact Second Derivatives", Technical Report T. P. 515, Atomic Energy Research Establishment, Harwell.

Higham, N. J., "Computing the Polar Decomposition — with Applications", SIAM *Journal on Scientific and Statistical Computing*, Vol. **7** No. 4 (October 1986) 1160–1174.

Marquardt, D. W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", *Journal of the Society for Industrial and Applied Mathematics* (SIAM), Vol. **11** No. 2 (June 1963) 431–441.

McCormick, G. P., "A Modification of Armijo's Step-Size Rule for Negative Curvature", *Mathematical Programming* **13** (1977) 111–115.

Moré, J. J., and D. C. Sorensen, "On the Use of Directions of Negative Curvature in a Modified Newton Method", *Mathematical Programming* **16** (1979) 1–20.

Moré, J. J., "Recent Developments in Algorithms and Software for Trust Region Methods", in *Mathematical Programming—The State of the Art—Bonn 1982*, A. Bachem, M. Grötschel, and B. Korte (eds.), Springer-Verlag (1983) 258–287.

Moré, J. J., and D. C. Sorensen, "Computing a Trust Region Step", SIAM *Journal on Scientific and Statistical Computing*, Vol. 4 No. 3 (September 1983) 553–572.

Moré, J. J., and D. C. Sorensen, "Newton's Method", in *Studies in Numerical Analysis, MAA Studies in Mathematics*, Vol. 24, G. H. Golub (ed.), The Mathematical Association of America (1984).

Morrison, D. D., "Methods for Nonlinear Least Squares Problems and Convergence Proofs, Tracking Programs and Orbital Determinations", *Procedings of the Jet Propulsion Laboratory Seminar* (1960) 1–9.

Murray, W., "Second Derivative Methods", in *Numerical Methods for Unconstrained Optimization*, W. Murray (ed.), Academic Press (1972) 57–71.

NAG Fortran Library Manual, Vol. 3, Numerical Algorithms Group, Oxford U. K., and Downers Grove, IL U. S. A. (January 1984).

Nazareth, L., "An Alternative Variational Principle for Variable Metric Updating", *Mathematical Programming* 30 (1984) 99–104.

Oren, S. S., " Self-Scaling Variable Metric Algorithms Without Linesearch for Unconstrained Minimization", *Mathematics of Computation*, Vol. 27 No. 124 (October 1973).

Oren, S. S., and E. Spedicato, "Optimal Conditioning of Self-Scaling Variable Mettric Algorithms", *Mathematical Programming* 10 (1796) 70–90.

Oren, S. S., "Perspectives on Self-Scaling Variable Metric Algorithms", *Journal of Optimization Theory and Applications*, Vol. 37 No. 2 (June 1982) 137–148.

Ortega, J. M., and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press (1970).

PORT Mathematical Subroutine Library Manual, A. T. & T. Bell Laboratories, Inc., Murray Hill, NJ U. S. A. (May 1984).

Powell, M. J. D., "A New Algorithm for Unconstrained Optimization", in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter (eds.), Academic Press (1970) 31–65.

Powell, M. J. D., "Convergence Properties of a Class of Minimization Algorithms", in *Nonlinear Programming 2*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), Academic Press (1975) 1–27.

Reinsch, C. H., "Smoothing by Spline Functions II", *Numerische Mathematik* M 16 (1971) 451–454.

Schnabel, R. B., "Optimal Conditioning in the Convex Class of Rank Two Updates", *Mathematical Programming* 15 (1978) 247–260.

Schnabel, R. B., "Conic Methods for Unconstrained Minimization and Tensor Methods for Nonlinear Equations", in *Mathematical Programming—The State of the Art—Bonn 1982*, A. Bachem, M. Grötschel, and B. Korte (eds.), Springer-Verlag (1983) 417–438.

Shanno, D. F., and P. C. Kettler, "Optimal Conditioning of Quasi-Newton Methods", *Mathematics of Computation* 24 (1970) 657–667.

Shanno, D. F., and K. H. Phua, "Matrix Conditioning and Nonlinear Optimization", *Mathematical Programming* 14 (1978a) 145–160.

Shanno, D. F., and K. H. Phua, "Numerical Comparison of Several Variable Metric Algorithms", *Journal of Optimization Theory and Applications* 25 (1978b) 507–518.

Shultz, G. A., R. B. Schnabel, and R. H. Byrd, "A Family of Trust-Region-Based Algorithms for Unconstrained Minimization with Strong Global Convergence Properties", SIAM *Journal on Numerical Analysis*, Vol. 22 No. 1 (February 1985) 47–67.

Sorensen, D. C., "The Q-Superlinear Convergence of a Collinear Scaling Algorithm for Unconstrained Optimization", SIAM *Journal on Numerical Analysis*, Vol. 17 No. 1 (February 1980) 84–114.

Sorensen, D. C., "Newton's Method with a Model Trust Region Modification", SIAM *Journal on Numerical Analysis*, Vol. 19 No. 2 (April 1982) 409–426.

Sorensen, D. C., "Trust Region Methods for Unconstrained Minimization", in *Nonlinear Optimization 1981*, M. J. D. Powell (ed.), Academic Press (1982) 29–38.

Spedicato, E., "A Variable Metric Method for Function Minimization Derived from Invariancy to Nonlinear Scaling", *Journal of Optimization Theory and Applications* 20 (1976) 315–328.

Stewart, G. W., *Introduction to Matrix Computations*, Academic Press (1973).

Stoer, J., "The Convergence of Matrices Generated by Rank-2 Methods from the Restricted β-Class of Broyden", *Numerische Mathematik*, Vol. 44 No. 1 (1984) 37–53.

Tassopoulos, A., and C. Storey, "A Variable Metric Method using a Nonquadratic Model", *Journal of Optimization Theory and Applications*, Vol. 43 No. 3 (July 1984) 383–394.

Todd, M. J., "Quasi-Newton Updates in Abstract Vector Spaces", SIAM *Review*, Vol. 26 No. 3 (July 1984) 367–377.

Wolfe, P. "Convergence Conditions for Ascent Methods", SIAM *Review* 11 (1969) 226–235.

Wolfe, P. , "Convergence Conditions for Ascent Methods II : Some Corrections", SIAM *Review* 13 (1971) 185–188.

3. Linear Least Squares

3.1 Overview

The linear least-squares problem

LLSQ

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2, \tag{3.1.1}$$

approximates a vector b by a linear combination of the columns of a matrix A. A thorough understanding of linear least squares is essential in connection with nonlinear least squares for several reasons. First, LLSQ is an important and well-understood special case of nonlinear least squares. Second, the classical Gauss-Newton methods (Chapter 4), and Levenberg-Marquardt methods (Section 5.2) for nonlinear least squares, iteratively solve linear least-squares subproblems. Finally, orthogonalization techniques related to those used to solve linear least-squares problems are applicable in many other situations in nonlinear programming, including quadratic programming (see the references cited in Section 6.3), which plays a key role in the new algorithms proposed in Chapter 6 for sums of squares.

Some theoretical background for LLSQ is reviewed in the next section. In Section 3.3, we show how orthogonal factorizations can be used to analyze and solve LLSQ (assuming exact arithmetic), and describe the most important orthogonal factorizations : the QR factorization and the singular-value decomposition. Numerical procedures for LLSQ are treated in the final section of this chapter.

3.2 Theoretical Properties

In this section we list some theoretical properties of LLSQ for later reference. As these results are well known, they are stated without proof. See Stewart [1973], Lawson and Hanson [1974], and Golub and Van Loan [1983] for more detail.

(3.2-1) The vector x is a solution to LLSQ if and only if x is a solution to the *normal equations*

$$A^T A x = A^T b, \tag{3.2.1}$$

or, equivalently, x solves LLSQ if and only if $Ax - b \in \mathcal{N}(A^T)$.

(3.2-2) The vector x is a solution to LLSQ if and only if $Ax = b_R$, where b_R is the projection of b onto $\mathcal{R}(A)$, or, equivalently, x is a solution to LLSQ if and only if $b - Ax = b_N$, where b_N is the projection of b onto $\mathcal{R}(A)^\perp = \mathcal{N}(A^T)$.

(3.2-3) LLSQ has a unique solution if and only if A has full column rank.

(3.2-4) The problem

MINLSQ

$$\min_{p \in \mathcal{S}} \{\|p\|_2\}, \quad \text{where} \quad \mathcal{S} \equiv \{\bar{x} \mid \|A\bar{x} - b\|_2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2\} \tag{3.2.2}$$

has a unique solution.

(3.2-5) The vector x is a solution to LLSQ if and only if the projection of x onto $\mathcal{R}(A^T)$ solves MINLSQ.

3.3 Orthogonal Factorizations

3.3.1 Orthogonal Factorizations and Linear Least Squares

For any matrix A, there exist orthogonal matrices Q and V such that

$$A = Q \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} V, \tag{3.3.1}$$

where R is a nonsingular triangular matrix of dimension r, the rank of A (see, for example, Lawson and Hanson [1974], Chapter 3). Factorizations of the form (3.3.1) can be used to analyze LLSQ because the l_2 norm is invariant under orthogonal transformations, and also to obtain solutions to LLSQ, since there are efficient and stable computational procedures for computing (3.3.1) (see Stewart [1973], Chapter 5; Lawson and Hanson [1974]; Golub and Van Loan [1983], Chapter 6).

To see this, let

$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} \equiv Vx \quad \text{and} \quad \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} \equiv Q^T b$$

be partitions of Vx and $Q^T b$ into the first r rows and the remaining rows. Then

$$\begin{aligned} \|Ax - b\|_2^2 &= \left\|Q^T \left(A(V^{-1}V)x - b\right)\right\|_2^2 = \left\|\left(Q^T A V^{-1}\right) Vx - Q^T b\right\|_2^2 \\ &= \left\| \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} - \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} \right\|_2^2 = \left\|R\hat{x}_1 - \hat{b}_1\right\|_2^2 + \left\|\hat{b}_2\right\|_2^2. \end{aligned}$$

It follows that any solution x to LLSQ must satisfy

$$R\hat{x}_1 = \hat{b}_1, \tag{3.3.2}$$

so that $\hat{x}_1$ is uniquely determined, although $\hat{x}_2$ is arbitrary. The triangular form of R is important from the point of view of solving (3.3.2) efficiently (see, for example, Stewart [1973], Chapter 3). The matrix R will have rank n if and only if A has column rank n. When this happens, (3.3.2) can be written as

$$R(Vx) = \hat{b}_1,$$

which completely determines x. From (3.3.2) it follows that the minimum l_2-norm solution is unique, because

$$\|x\|_2^2 = \|Vx\|_2^2 = \left\| \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} \right\|_2^2 = \|\hat{x}_1\|_2^2 + \|\hat{x}_2\|_2^2,$$

showing that x has minimum l_2 norm only if $\hat{x}_2 = 0$.

3.3.2 QR Factorizations — the Householder Method

We will now describe a direct method for obtaining a factorization of the form (3.3.1), assuming that exact arithmetic is used. The procedure is one that is common in numerical computations, and is given as background for discussion of the numerical properties of solutions to LLSQ in the next section. The factorization is accomplished in two stages. First the matrix is reduced to upper-trapezoidal form via Householder transformations applied to the left and column permutations. Then, if necessary, more transformations are applied to the right so that the result is upper triangular.

We will use the notation

$$\mathcal{H}^n(v, w)$$

to represent an orthogonal (Householder) matrix that transforms the n-vector v into a multiple of the n-vector w (see, for example, Lawson and Hanson [1974], Chapter 3). The notation

$$\mathcal{P}^n(i, j)$$

will be used for permutations (which are also orthogonal). When applied to the right of a matrix, $\mathcal{P}^n(i, j)$ has the effect of swapping the ith and jth rows of the matrix, while

Proof:

The theorem follows from Lemmas 3.3-1 and 3.3-2 if $V \equiv PU$. ∎

3.3.3 Singular-Value Decomposition (SVD)

Another useful variant of (3.3.1) is the *singular-value decomposition*, or SVD. It differs from the complete orthogonal factorization in that R is diagonal with non-negative diagonal entries. Because of its relation to the eigenvalue problem, computation of the SVD normally requires an iterative procedure. It is nonetheless important because the existence of the diagonal form makes analysis of many matrix problems, including LLSQ, transparent.

Theorem 3.3-4 (Singular-Value Decomposition):

For any $m \times n$ matrix A, there exist orthogonal matrices U and V such that

$$U^{\mathrm{T}} A V = \begin{cases} (S \quad 0), & \text{if } m < n; \\ S, & \text{if } m = n; \\ \begin{pmatrix} S \\ 0 \end{pmatrix}, & \text{if } m > n, \end{cases} \tag{3.3.6}$$

where S is diagonal with non-negative diagonal entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{m,n\}}$. ∎

For a proof see Stewart [1973], Chapter 6. The diagonals $\sigma_1, \sigma_2, \ldots, \sigma_{\min\{m,n\}}$ are called the *singular values* of A, and the columns of U and V are the *singular vectors*. The index r of the smallest nonzero singular value is equal to the rank of A.

In terms of the SVD, the minimum l_2-norm solution $\tilde{x}$ to MINLSQ can be expressed as

$$\tilde{x} = \sum_{i=1}^{r} \tau_i v_i, \quad \text{where} \quad \tau_i \equiv \frac{u_i^{\mathrm{T}} b}{\sigma_i},$$

where u_i, v_i represent the ith columns of U, V, respectively, so that

$$\|\tilde{x}\|_2^2 = \sum_{i=1}^{r} \tau_i^2.$$

Unless $u_i^{\mathrm{T}} b$ is vanishes, τ_i becomes infinitely large as σ_i approaches zero, which means that the minimum l_2-norm least-squares solution can become arbitrarily large if A is nearly rank

deficient. The effects of perturbations on the solution to MINLSQ depend on the *condition number* of A, which is usually defined by

$$cond(A) \equiv \frac{\sigma_1}{\sigma_r}, \tag{3.3.7}$$

where σ_r is the smallest nonzero singular value of A. The matrix A is said to be *ill-conditioned* if $cond(A)$ is large. Small perturbations for ill-conditioned A may result in substantial changes in the solution to MINLSQ, particularly if the original matrix and the perturbed matrix do not have the same rank. This property of the linear least-squares problem makes the numerical solution of MINLSQ difficult when the columns of A are linearly dependent, or nearly so.

3.4 Computational Considerations

This section is concerned with issues involved in the numerical solution of LLSQ, including rank estimation. The emphasis will be on orthogonal (SVD and QR) factorizations, because they are the most stable numerical methods known for MINLSQ, in the sense that numerical errors do not cause disproportionately large errors in the solution. This discussion is intended to apply only to linear least-squares problems that are reasonably small and dense — a somewhat different set of considerations and priorities would be associated with large, sparse problems.

In what follows, the relation

$$X \cong Y$$

means that Y is a computed version of X, so that any zeros appearing in Y should be interpreted as quantities that are assumed to be negligible in the computation.

3.4.1 Rank Estimation

3.4.1.1 Defining Rank with the Singular-Value Decomposition

For details concerning computation of the SVD, see Wilkinson [1978], or Golub and Van Loan [1983], Chapters 6 and 8. What is important for our purposes is that an iterative procedure is generally required to obtain the SVD of a matrix, and that the stopping criteria

are chosen so that the computed result is the SVD of a matrix $A+E$, with $\|E\|_2 < \epsilon \|A\|_2$, implying that the error in any one of the computed singular values is no greater than $\|E\|_2$. For rank estimation, ϵ should be of the order of the relative machine precision ϵ_M, so that the singular values computed by the SVD algorithm are as accurate as is possible in floating-point arithmetic. Matrices in the examples of this subsection are presented in terms of their computed singular-value decompositions.

In the nearly rank-deficient case, some method is needed for deciding which, if any, computed singular values would have been zero in exact arithmetic. One possiblity is to have an absolute tolerance δ, and define

$$rank(A) \equiv \max \{i \mid \sigma_i > \delta\}. \tag{3.4.1}$$

However, rank estimated in this way does not take into account the relative size of the singular values. For example, the matrices

$$A_1 \cong U_1^{\mathrm{T}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} V_1 \quad \text{and} \quad A_2 \cong U_2^{\mathrm{T}} \begin{pmatrix} 10^{10} & 0 \\ 0 & 1 \end{pmatrix} V_2,$$

would have the same estimated rank for all $\delta < 1$ with (3.4.1), even though the numerical uncertainty in $\sigma_2(A_2)$ would be significant compared to its computed value for ϵ_M near 10^{-10}. Numerical rank could instead be defined relative to $\|A\|_2 = \sigma_1$, using

$$rank(A) \equiv \max \{i \mid \sigma_i > \epsilon\sigma_1\}. \tag{3.4.2}$$

Basically, (3.4.2) says that the rank will be chosen so that the matrix is not too ill-conditioned (see (3.3.7)). By (3.4.2), $rank(A_1) = 2$ and $rank(A_2) = 1$ for $10^{-10} < \epsilon < 1$. However, when σ_1 is small compared to machine precision, rank may be overestimated. For example, the matrices

$$A_1 \cong U_1^{\mathrm{T}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} V_1 \quad \text{and} \quad A_3 \cong U_3^{\mathrm{T}} \begin{pmatrix} 10^{-10} & 0 \\ 0 & 10^{-10} \end{pmatrix} V_3$$

have the same rank for all values of ϵ, using (3.4.2). If $\epsilon_M \approx 10^{-10}$, and A_3 is the result of some computation, then each element of A_3 may be a "noise-level" quantity, in the sense that the numerical uncertainty in its value is significant compared to its magnitude. An alternative that allows for the possibility that the relative uncertainty in σ_1 may increase as its magnitude decreases when $\sigma_1 < 1$ is

$$rank(A) \equiv \max \{i \mid \sigma_i > \epsilon(1 + \sigma_1)\}. \tag{3.4.3}$$

Definition (3.4.3) is also not entirely statisfactory because there are matrices, such as

$$A_4 \cong U_4^{\mathrm{T}} \begin{pmatrix} 1 & 0 \\ 0 & 2\epsilon \end{pmatrix} V_4,$$

in which small perturbations can cause a change in numerical rank. Moreover, if there are more than two singular values, then the decision about how to define the estimated rank r is more clearcut if there is a gap in the sequence of singular values :

$$\frac{\sigma_{r+1}}{\sigma_r} \ll \min_{i<r} \frac{\sigma_{i+1}}{\sigma_i}. \tag{3.4.4}$$

But even condition (3.4.4) may not be satisfied for some matrices, for example

$$A_5 \cong U_5^{\mathrm{T}} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 10^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 10^{-n} \end{pmatrix} V_5. \tag{3.4.5}$$

The vector b can also be brought into consideration in rank estimation. Singular values would then be considered negligible if they contribute relatively large components to a least-squares solution. If

$$A = U^{\mathrm{T}} \begin{pmatrix} S & 0 \end{pmatrix} V \quad \text{or} \quad A = U^{\mathrm{T}} \begin{pmatrix} S \\ 0 \end{pmatrix} V,$$

where

$$S = diag(\sigma_1, \dots, \sigma_{\min\{m,n\}}),$$

then the solution $\tilde{x}(r)$ to MINLSQ if $rank(A) = r$ has the following characterization :

$$\tilde{x}(r) = \sum_{i=1}^{r} \tau_i v_i ; \qquad \|\tilde{x}(r)\|_2^2 = \sum_{i=1}^{r} \tau_i^2 ; \qquad \tau_i \equiv \frac{u_i^{\mathrm{T}} b}{\sigma_i}. \tag{3.4.6}$$

where u_i, v_i are the ith columns of U and V, respectively. We could, for example, define $rank(A\,;b)$ — the *rank of A relative to b* — to be the largest integer i that satisfies the conditions

$$\epsilon_1 \|\tilde{x}(i)\|_2 < 1 \tag{3.4.7}$$

and

$$\frac{\|\tilde{x}(i)\|_2}{\|\tilde{x}(i-1)\|_2} > \epsilon_2 \frac{\|\tilde{x}(i+1)\|_2}{\|\tilde{x}(i)\|_2}, \tag{3.4.8}$$

for some small quantities ϵ_1 and ϵ_2. If $rank(A\,;b) = r$, condition (3.4.7) places an upper limit on the size of $\|\tilde{x}(r)\|_2$, while condition (3.4.8) says that there must be a break in the

sequence $\{\|\tilde{x}(r)\|_2\}$ after the rth term. Even with (3.4.7) and (3.4.8) as rank-estimation criteria, there are cases in which suitable values for ϵ_1 and ϵ_2 could not be found for a given matrix A and vector b. To illustrate this point, values of $\|\tilde{x}(r)\|_2$ when S is the diagonal matrix in (3.4.5), for $r = 1, 2, \ldots, 11$, and two different vectors b, are given in the following table.

$\|\tilde{x}(r)\|_2$, when A is the diagonal matrix of (3.4.5), and $b = (b_1, \ldots, b_m)^T$

r	$b_i \equiv 10^{1-i}$	$b_i \equiv 1$
1	1.00	10^0
2	1.41	10^1
3	1.73	10^2
4	2.00	10^3
5	2.24	10^4
6	2.45	10^5
7	2.65	10^6
8	2.82	10^7
9	3.00	10^8
10	3.16	10^9
11	3.32	10^{10}

For $u_i^T b = 10^{1-i}$, $\|\tilde{x}(r)\|_2 = \sqrt{r}$ does not grow very rapidly with r, so that (3.4.7) cannot be satisfied for small values of ϵ, unless r is very large. On the other hand, $\|\tilde{x}(r)\|_2 \approx 10^{r-1}$ for $u_i^T b = 1$, so that $\|\tilde{x}(r)\|_2$ is large compared to $\|A\|_2$ and $\|b\|_2$ for relatively small values of r. However, in neither case can (3.4.8) be satisfied for small values of ϵ_2, because there are no large gaps in the sequence $\{\|\tilde{x}(i)\|_2\}$. In situations like these, rank estimation is difficult.

3.4.1.2 Defining Rank with QR Factorizations

There are two important interrelated decisions in the orthogonal reduction from the left (Lemma 3.3-1) when performed numerically. First, a decision has to be made concerning the order in which to reduce the columns. In exact arithmetic, all that matters is that a nonzero column be chosen at each step; linear dependence would be detected by the eventual appearance of column of zeros. In floating-point arithmetic, there will probably not be any columns that are exactly zero due to numerical errors. What is often done is to reduce the column of largest l_2 norm at each step, which means that the sequence of diagonals will be decreasing in magnitude.

The second decision in a QR factorization involves terminating the column-wise reduction, which is equivalent to estimating the rank of the matrix. After the kth step,

$$Q_k \ldots Q_2 Q_1 A P_1 P_2 \ldots P_k \cong \begin{pmatrix} T_{11}^k & T_{12}^k \\ 0 & T_{22}^k \end{pmatrix},$$

where T_{11}^k is $k \times k$ upper triangular and nonsingular, Q_i are Householder transformations, and P_i are permutations. In theory, this stage can be terminated after the rth step, if r is the rank of A, since then either $T_{22}^r = 0$, or else r is equal to the number of rows in A. When A has linearly dependent columns, it is unlikely that the submatrix T_{22}^k will vanish for any k in finite-precision arithmetic, so a criterion such as

$$\|T_{22}^k\|_2 \leq \epsilon \|A\|_2, \tag{3.4.9}$$

for some machine-related constant ϵ, is used to decide when to stop. Information about the nature of the data A and b, and about how the solution x will ultimately be interpreted, can sometimes be used to influence the choice. In Chapter 4 we will see an example of this in the discussion of Gauss-Newton methods. Because $\sigma_{k+1} \leq \|T_{22}^k\|_2$, use of (3.4.9) to estimate the rank of A is justified if $\|T_{22}^k\|_2$ is very small in magnitude. But there are nearly rank-deficient triangular matrices that have no small diagonal elements (see Wilkinson [1965], or Lawson and Hanson [1974], Chapter 6), so that it is possible for the reduction to proceed without detecting ill-conditioning.

To illustrate the importance of column pivoting in the algorithm, consider the 2×2 example

$$\begin{pmatrix} \alpha & \beta \\ 0 & \gamma \end{pmatrix}. \tag{3.4.10}$$

If the second column is chosen first for reduction, then the Householder transformation that restores triangular form can be written as

$$\begin{pmatrix} c & s \\ s & -c \end{pmatrix}, \quad \text{where} \quad c \equiv \frac{\beta}{\sqrt{\beta^2 + \gamma^2}} \quad \text{and} \quad s \equiv \frac{\gamma}{\sqrt{\beta^2 + \gamma^2}}.$$

Applying this to (3.4.9) with the columns interchanged, we have

$$\begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} \beta & \alpha \\ \gamma & 0 \end{pmatrix} = \begin{pmatrix} c\beta + s\gamma & c\alpha \\ 0 & s\alpha \end{pmatrix}. \tag{3.4.11}$$

If $\alpha \approx \epsilon$, $\gamma \approx \epsilon$, and $\beta \approx 1$, for $\epsilon \ll 1$ then $c\beta + s\gamma \approx 1$ and $s\alpha \approx \epsilon^2$. When ϵ is of the order of the square root of machine precision, the estimated rank could be 2 if it were based on the diagonals of (3.4.10) and 1 if it were based on the diagonals of (3.4.11).

More sophisticated techniques than QR with column pivoting for extracting rank information from orthogonal factorizations have been developed (see Karasalo [1974], Golub, Klema, and Stewart [1976], Manteuffel [1981], Stewart [1984], and Foster [1986]). The extra computational expense involved in rank estimation with these methods is often not worthwhile, because although σ_{k+1} could be small when $\left\|T_{22}^k\right\|_2$ is relatively large, in practice the diagonals of the triangular factor in the QR factorization tend to reflect the magnitude of the singular values when the largest column is always chosen for reduction.

3.4.1.3 Effects of Data Transformation

Numerical methods for rank estimation are critically dependent on the representation of the data A and b in LLSQ. The exact solution to the problem

$$\min_{x \in \Re^n} \|W(Ax - b)\|_2 ,$$

with W nonsingular, will generally be different from that of LLSQ when the minimum value is nonzero. The numerical solution could be changed even in the zero residual case, because the decisions made by the algorithm with respect to WA and Wb — for example, the column-pivoting strategy in the Householder method — may not be the same as those with A and b. The same remarks apply to transformation of the variables : if we choose to solve LLSQ for $w = Cx + c$, with C nonsingular, rather than for x, then it will be necessary to determine the rank of the matrix AC^{-1} relative to the vector $b + AC^{-1}c$, which may be numerically different than that of A relative to b. Some discussion of the effects of data transformation on linear least-squares problems can be found in Lawson and Hanson [1974], Chapter 25. It is not possible, in general, to give a computer algorithm the information necessary for it to decide what transformations should be used. Moreover, any automatic transformation could destroy the effects of deliberate choices made in setting up the problem, which may already have taken into account both the nature of the problem to be solved, and the limitations of floating-point arithmetic.

3.4.2 Computational Error for the Householder Method

We have just seen that the minimum l_2-norm solution to LLSQ can be very sensitive to small changes in the data when the problem is nearly rank deficient. Bounds on the

effect of perturbations on the solution to LLSQ depend on the condition number of A (see (3.3.7)). When perturbations do not cause an increase in rank, the errors they introduce can be magnified by as much as $(cond(A))^2$ times the size of the residual, although in the zero-residual case magnification by a factor of $cond(A)$ is the worst that can occur when techniques based on orthogonal factorizations are used (see Stewart [1973], Chapter 5, Lawson and Hanson [1974], Chapter 16, and Golub and Van Loan [1983], Chapter 6).

The computational error incurred in the solution of MINLSQ by the Householder method described in Lemmas 3.3-1 and 3.3-2 can be summarized as follows. If $\hat{x}$ is the exact solution, and $\hat{x} + \delta x$ is the computed solution, then

$$\|\delta x\|_2 \leq \mathcal{O}\left(\epsilon_M (m+n)^2\right) \|\hat{x} + \delta x\|_2 . \tag{3.4.12}$$

It is not possible to bound δx in terms of $\hat{x}$ alone, because the size of the computed solution can vary greatly depending on the estimated rank (see (3.4.6)). If the estimated rank is r, then $\hat{x} + \delta x$ is the exact solution of a perturbed problem posed in terms of the data $A + \delta A$ and $b + \delta b$ rather than A and b, where

$$\|\delta A\|_F \leq \|T_{22}^r\|_F + \mathcal{O}\left(\epsilon_M (m+n)^2\right) \|A\|_F , \tag{3.4.13}$$

and

$$\|\delta b\|_2 \leq \mathcal{O}\left(\epsilon_M (m+n)^2\right) \|b\|_2 . \tag{3.4.14}$$

Notice that unless $\|T_{22}^r\|_2$ is small, the perturbation δA in A could be large. However, if A is well-conditioned and has full column rank, significant errors are not introduced by the numerical algorithm, provided m and n are small. For proofs of these and other results on the numerical stability of linear least-squares problems, see Lawson and Hanson [1974], Chapters 9, 15-17.

3.4.3 Other Approaches to Linear Least Squares

3.4.3.1 QR with Column Deletion

After orthogonal reduction from the left, if the estimated rank is less than n, then the reduced matrix has the form

$$QAP \cong \begin{pmatrix} T_{11} & T_{12} \\ 0 & 0 \end{pmatrix},$$

with T_{11} a nonsingular upper-triangular matrix. To obtain the solution x_{MIN} of minimum l_2 norm, Householder transformations may be applied to introduce zeros in the submatrix T_{12}. In some applications, other solutions to LLSQ may be adequate, and there may be no need for the second phase of the reduction. To see this, let

$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} \equiv P^{\mathrm{T}} x \quad \text{and} \quad \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} \equiv Qb,$$

so that

$$\begin{aligned} \|Ax - b\|_2^2 &= \left\| \begin{pmatrix} T_{11} & T_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} - \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} \right\|_2^2 \\ &= \|T_{11}\hat{x}_1 + T_{12}\hat{x}_2 - \hat{b}_1\|_2^2 + \|\hat{b}_2\|_2^2. \end{aligned}$$

Solutions x to LLSQ have the representation

$$x = P \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix},$$

where

$$T_{11}\hat{x}_1 = \hat{b}_1 - T_{12}\hat{x}_2. \tag{3.4.15}$$

Once the vector $\hat{x}_2$ is specified, $\hat{x}_1$ (and hence x) is determined. The simplest choice, corresponding to $\hat{x}_2 = 0$, is called a *basic* solution. Forming a basic solution is equivalent to replacing the matrix A in LLSQ by a linearly independent subset of its columns. A bound on the size of a basic solution is

$$\|x_{BASIC}\|_2 \leq \|x_{MIN}\|_2 \sqrt{1 + \|T_{11}^{-1}T_{12}\|_2^2}.$$

See Golub and Van Loan [1983], Chapter 6, for further discussion.

3.4.3.2 Gram-Schmidt Orthogonalization

A modified version of Gram-Schmidt orthogonalization can be used instead of Householder transformations for reduction from the left without affecting the stability of the method. The computed matrix Q that results from this process may not be close to being orthogonal, which could be a disadvantage relative to the Householder method when an orthogonal Q is needed for other purposes within an algorithm. See Golub and Van Loan [1983], Chapter 6, for further discussion.

3.4.3.3 Elimination Methods

A variety of elimination methods could also be applied to LLSQ. It is possible to solve the normal equations (3.2.1) using the Cholesky factorization. However, errors in the data can be magnified in the solution by a factor of $(cond(A))^2$ even if the residual is zero. Other variations include combining orthogonal reduction on the left with elimination on the right to get a factorization similar to (3.3.1), in which V is not orthogonal, or applying Gaussian elimination directly to A for square and underdetermined systems. The choice depends on the trade-off between efficiency and stability. Elimination methods typically require fewer operations than similar methods involving orthogonal transformations, but only at the risk of significantly greater numerical error in the solution. See Lawson and Hanson [1974], Chaper 19, and Golub and Van Loan [1983], Chapter 6, for further discussion.

3.4.3.4 Regularization

A common technique for ill-conditioned linear least-squares problems is to solve

$$\min_{x \in \Re^n} \|Ax - b\|_2^2 + \lambda \|x\|_2^2, \tag{3.4.16}$$

for some $\lambda > 0$, with the intent of preventing $\|x\|_2$ from becoming large when A is ill-conditioned. Methods of this type are called *regularization* methods; they are trust-region methods (Section 2.4.2) for minimizing the quadratic function $\|Ax - b\|_2$. Solving (3.4.16) is equivalent to solving the linear least-squares problem

$$\min_{x \in \Re^n} \left\| \begin{pmatrix} A \\ \sqrt{\lambda} I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2, \tag{3.4.17}$$

in which the coeffcient matrix has full column rank. For further discussion about regularization techniques, and, in particular, about the choice of the parameter λ, see Chapter 25 of Lawson and Hanson [1974], Eldén [1977; 1984], Varah [1979], and Gander [1981]. The Levenberg-Marquardt methods for nonlinear least squares, which are discussed in detail in Section 5.2, solve a regularized linear least-squares subproblem at each iteration.

3.5 Bibliography

Eldén, L., "An Algorithm for the Regularization of Ill-Conditioned Least Squares Problems", *BIT* 17 (1977) 134–145.

Eldén, L., "A Note on the Computation of the Generalized Cross-Validation Function for Ill-Conditioned Least Squares Problems", *BIT* 24 (1984) 467–472.

Foster, L. V., "Rank and Null Space Calculation using Matrix Decomposition without Column Interchanges", *Linear Algebra and its Applications* 74 (February 1986) 47–72.

Gander, W., "Least Squares with a Quadratic Constraint", *Numerische Mathematik*, Vol. 36 No. 3 (1981) 291–308.

Golub, G. H., V. Klema, and G. W. Stewart, "Rank Degeneracy and Least Squares Problems", Technical Report STAN-CS-76-559, Computer Science Department, Stanford University (August 1976).

Golub, G. H., and C. L. Van Loan, *Matrix Computations*, Johns-Hopkins (1983).

Karasalo, I., "A Criterion for Truncation of the QR–Decomposition Algorithm for the Singular Linear Least Squares Problem", *BIT* 14 (1974) 156–166.

Lawson, C. L., and R. J. Hanson, *Solving Least Squares Problems*, Prentice Hall (1974).

Manteuffel, T. A., "An Interval Analysis Approach to Rank Determination in Linear Least Squares Problems", SIAM *Journal on Scientific and Statistical Computing*, Vol. 2 No. 3 (September 1981) 335–348.

Stewart, G. W., *Introduction to Matrix Computations*, Academic Press (1973).

Stewart, G. W., "Rank Degeneracy", SIAM *Journal on Scientific and Statistical Computing*, Vol. 5 No. 2 (June 1984) 403–413.

Varah, J. M., "A Practical Examination of Some Numerical Methods for Linear Discrete Ill-Posed Problems", SIAM *Review*, Vol. 21 No. 1 (January 1979) 100–111.

Wilkinson, J. H., *The Algebraic Eigenvalue Problem*, Oxford (1965).

Wilkinson, J. H., "Singular Value Decomposition — Basic Aspects", in *Numerical Software — Needs and Availablity*, Academic Press (1978) 109–135.

4. Gauss-Newton Methods

4.1 Overview

We recall from Chapter 1 that the nonlinear least-squares problem is given by

$$\min_{x \in \Re^n} \frac{1}{2} \sum_{i=1}^{m} \phi_i^2(x),$$

or

$$\min_{x \in \Re^n} \frac{1}{2} \|f(x)\|_2^2,$$

where $\phi_i(x)$ are real-valued functions, and $f(x)$ maps $\Re^n$ to $\Re^m$. The classical approach to nonlinear least squares, called the *Gauss-Newton* method, locally approximates each residual component ϕ_i of f by a linear function, using the relationship

$$f(x+p) \approx f(x) + J(x)p + \mathcal{O}(\|p\|^2).$$

The step to the new iterate from the current point is in the direction of the vector p that minmizes

$$\|f + Jp\|_2^2,$$

which is equivalent to modeling the change in the nonlinear least-squares objective $\frac{1}{2} f^T f$ by the quadratic function

$$\bar{g}^T p + \frac{1}{2} p^T J^T J p, \tag{4.1.1}$$

where

$$\bar{g} \equiv \nabla \left(\frac{1}{2} f^T f \right) = J^T f.$$

Hence the Gauss-Newton method differs from Newton's method by approximating the Hessian matrix

$$J^T J + \sum_{i=1}^{m} \phi_i \nabla^2 \phi_i$$

by $J^T J$, a strategy that would seem reasonable when the residuals are small.

In Section 4.2 we show that a class of numerical methods, rather than a single method, is defined by the linearization of f, and motivate these methods from considerations discussed in Chapter 2 (Unconstrained Optimization) and Chapter 3 (Linear Least Squares). Section 4.3 surveys some research on computational aspects of Gauss-Newton methods. By our

definition Gauss-Newton methods are linesearch methods, and in Section 4.4 conditions are stated and proved under which solutions to the linear least-squares subproblem by the SVD, and by QR factorization with column pivoting, are descent directions for the nonlinear least-squares objective. Examples of the performance of the Gauss-Newton method on problems with ill-conditioned Jacobians are presented in Section 4.5. An example of poor performance of a Gauss-Newton method on a zero-residual problem with a well-conditioned Jacobian is analyzed in Section 4.6. A final section gives numerical results for two different Gauss-Newton methods using the test problems described in Chapter 1.

4.2 Motivation

The Gauss-Newton method for nonlinear least squares can be viewed as a modification of Newton's method in which $J^{\mathrm{T}}J$ is used to approximate the Hessian matrix

$$J(x)^{\mathrm{T}}J(x) + \sum_{i=1}^{m} \phi_i(x)\nabla^2\phi_i(x).$$

Two promising aspects of this approximation are that computation of $J^{\mathrm{T}}J$ involves only first derivatives, and that $J^{\mathrm{T}}J$ is always at least positive semi-definite. Moreover, if $f(x^*) = 0$ and $J(x^*)^{\mathrm{T}}J(x^*)$ is positive-definite, then x^* is an isolated local minimum and the method is locally quadratically convergent. To see this, define

$$B \equiv \sum_{i=1}^{m} \phi_i(x)\nabla^2\phi_i(x),$$

and consider the expansion

$$0 = J(x^*)^{\mathrm{T}}f(x^*) = \bar{g} + \left(J^{\mathrm{T}}J + B\right)(x - x^*) + \mathcal{O}(\|x - x^*\|^2),$$

which is valid since it is assumed that f has continuous second derivatives. The Gauss-Newton search direction at the current iterate minimizes the quadratic function

$$\bar{g}^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}J^{\mathrm{T}}Jp, \tag{4.2.1}$$

and therefore satisfies the equations

$$J^{\mathrm{T}}Jp = -\bar{g}. \tag{4.2.2}$$

Because $J(x^*)^T J(x^*)$ is positive definite and J is continuous, $(J^T J)^{-1}$ exists and has bounded norm in a neighborhood of x^*. Hence convergence is quadratic when $(J^T J)^{-1} B$ is $\mathcal{O}(\|x - x^*\|)$. In particular, there will be quadratic convergence whenever $f(x^*) = 0$, because then f, and also B, is $\mathcal{O}(\|x - x^*\|)$. When the objective vanishes at a minimum, (4.2.1) is a quadratic approximation to $\frac{1}{2} f(x^*)^T f(x^*)$, and the Gauss-Newton direction approaches the Newton search direction in the limit. When $f(x^*) \neq 0$, the rate of convergence is linear if the smallest singular value of $J^T J$ exceeds the the largest singular value of B, but may otherwise diverge. It is not convergent when the minimum singular value of B exceeds the maximum singular value of $J^T J$ in a neighborhood of a solution. For more detailed convergence analysis see, for example, Osborne [1972], McKeown [1975a, 1975b], Ramsin and Wedin [1977], Deuflhard and Apostolescu [1980], Dennis and Schnabel [1983], Chapter 10, Schaback [1985], and Haussler [1986].

A drawback of the Gauss-Newton method is that when $J^T J$ is singular, or, equivalently, when J has linearly dependent columns, (4.2.1) does not have a unique minimizer. For this reason the Gauss-Newton method should more accurately be viewed as a class of methods, each member being distinguished by a different choice of p when $J^T J$ is singular. The set of vectors that minimize (4.2.1) is the same as the set of solutions to the linear least-squares problem

$$\min_{p \in \Re^n} \|Jp + f\|_2 . \tag{4.2.3}$$

In Chapter 3 we gave two alternatives to (4.2.3) that have a unique solution for any J. The first was to find the solution of minimum l_2 norm :

$$\min_{p \in \mathcal{S}} \|p\|_2 , \tag{4.2.4}$$

where $\mathcal{S}$ is the set of solutions to (4.2.3). The second was to solve

$$\min_{p \in \Re^n} \|\hat{J}p + f\|_2, \tag{4.2.5}$$

where $\hat{J}$ is a matrix consisting of exactly $rank(J)$ linearly independent columns of J, for a basic solution of (4.2.3). Note that (4.2.5) may actually describe more than one alternative, since $\hat{J}$ is not uniquely specified if J has linearly dependent columns, although there are only a finite number of possibilities. We have already discussed at length in Chapter 3 the difficulties inherent in computing solutions to (4.2.4) and (4.2.5) when J is ill-conditioned, and showed that the numerical solution of these problems is dependent on the criteria used

to estimate the rank of J. From now on, the term "Gauss-Newton method" will refer to any linesearch method in which the search direction is the result of some computational procedure for solving (4.2.4) or (4.2.5).

For the moment, let us assume that exact solutions to (4.2.4) or (4.2.5) can be computed. Because these search directions satisfy (4.2.2), they are directions of descent for $f^T f$ whenever $J^T f \neq 0$. To guarantee convergence, the search direction must also be bounded away from orthogonality to the gradient, a condition that may not be met by a Gauss-Newton method unless the eigenvalues of $J^T J$ are bounded away from zero for the sequence of iterates. Powell [1970] gives an example of convergence of a Gauss-Newton method with exact line search to a non-stationary point. Moreover, it was shown in Chapter 3 that when $J^T J$ is nearly singular, the (unique) solution to (4.2.2) can be very large in magnitude compared to $\|J\|_2$ and $\|f\|_2$, while in Chapter 2 we mentioned that linesearch methods may not be able to determine an adequate step length when $\|p\|_2$ is large. In finite-precision arithmetic, bounding the size of the norm of the solution was already a concern in formulating the basic criteria for rank estimation suggested in Chapter 3 and, in the context of nonlinear least squares, the other requirement of a linesearch method — that p must be a descent direction bounded away from orthogonality to the gradient — can be added as yet another consideration in giving a numerical definition of $rank(J)$. We will return to this idea in Section 4.5, where examples of Gauss-Newton methods applied to problems with ill-conditioned Jacobians are given.

The performance of Gauss-Newton methods is not fully understood. Gauss-Newton methods are of practical interest because there are many instances in which they work very well in comparison to other methods, and in fact most successful specialized approaches to nonlinear least-squares problems are based to some extent on Gauss-Newton methods and attempt to exploit this behavior whenever possible. However, it is not hard to find cases where Gauss-Newton methods perform poorly, so that they cannot be successfully applied without modification to general nonlinear least-squares problems. These remarks will be substantiated by examples in the next three sections, and also in Section 5.7, where a comparison is made of the performance of various numerical methods applied to nonlinear least-squares problems.

Perhaps a reason for the variability in the performance of Gauss-Newton methods is that they are not theoretically well-defined. To see this, let $Q(x)$ be a $k \times m$ orthogonal

matrix function on $\Re^n$, that is, $Q(x)^T Q(x) = I$ for all x. Then $\|Q(x)f(x)\|_2^2 = \|f(x)\|_2^2$ for all x, and consequently the function $\tilde{f} \equiv Qf$ defines the same nonlinear least-squares problem as does f. The Jacobian matrix of $\tilde{f}$ is $\tilde{J} \equiv QJ + (\nabla Q)f$, so that a minimizer of $\|\tilde{J}p + \tilde{f}\|_2$ will ordinarily be different from a minimizer of $\|Jp + f\|_2$, unless $Q(x)$ happens to be a constant transformation. However, if both Q and f have k continuous derivatives, then $\nabla^i \|Q(x)f(x)\|_2^2 = \nabla^i \|f(x)\|_2^2$ for $i = 1, 2, \ldots, k$. Letting $W = (\nabla Q)f$, so that $\tilde{J} = QJ + W$, we have

$$\tilde{J}^T \tilde{J} = J^T J + (J^T Q^T W + W^T Q J) + W^T W,$$

showing that the Gauss-Newton approximation $J^T J$ to the full Hessian matrix is changed when f is transformed by an orthogonal function that varies with x. Thus, with exact arithmetic, there are many Gauss-Newton methods corresponding to a given vector function (in fact, each step of a Gauss-Newton method could be defined by a different transformation of f), although Newton's method remains invariant (see also Nocedal and Overton [1985], p. 826). Moreover, the conditioning of $\tilde{J}$ may be very different from that of J, so that, for example, $\tilde{J}$ might have full rank when J is nearly rank deficient. Since k may be greater than n, it is possible to imbed the given nonlinear least-squares problem in a larger one. To the best of our knowledge the idea of preconditioning a Gauss-Newton method by an orthogonal function at each step has never been explored, although some work has been done on conjugate-gradient acceleration for Gauss-Newton methods in the full-rank case (see Section 5.5.3).

4.3 Studies of Gauss-Newton Methods

Our main concern in this section is with research that specifically addresses computational aspects of Gauss-Newton methods. For a survey of some of the early (mostly theoretical) research on Gauss-Newton methods, see Dennis [1977].

Bard [1970] compares some safeguarded Gauss-Newton methods with a Levenberg-Marquardt method (see Section 5.2) and some quasi-Newton methods for unconstrained optimization on a set of ten test problems from nonlinear parameter estimation with relatively small residuals. Since his implementations include bounds on the variables that are enforced by adding a penalty term to the objective function, his results are not directly comparable to any of the methods described in this research. He finds that the Gauss-Newton methods

are more efficient in terms of function and derivative evaluations than the quasi-Newton methods, but that there is no significant difference in the relative performance of the Gauss-Newton methods and the Levenberg-Marquardt method.

McKeown [1975a, 1975b] studies test problems of the form,

$$f(x) = f_0 + G_0 x + \frac{1}{2} \begin{pmatrix} x^{\mathrm{T}} H_1 x \\ \vdots \\ x^{\mathrm{T}} H_m x \end{pmatrix},$$

chosen in order that factors affecting asymptotic linear convergence could be controlled. He uses three different problems, each with seven different values of a parameter that varies an asymptotic linear convergence factor. The algorithms tested include some quasi-Newton methods for unconstrained optimization, as well as some specialized methods for nonlinear least squares that have since been superseded. He concludes that, when the asympotic convergent factor is small, the Gauss-Newton method is more efficient than the quasi-Newton methods but that the opposite is true when the asympotic convergence factor is large. No mention is made of strategies to deal with rank-deficient Jacobians in the Gauss-Newton method, so that presumably this situation is never encountered in his experiments. We included these test problems in our numerical experiments (see the results for problems 39. - 41. in Sections 2.6, 4.7, and 5.6 and also the discussion in Section 5.7), and reached the same conclusions relative to the quasi-Newton methods. The Jacobian matrix was well-conditioned at every iteration in all of the cases tested.

Ramsin and Wedin [1977] compare the performance of a Gauss-Newton method with that of a Levenberg-Marquardt method for nonlinear least squares (see Chapter 5, Section 2) and a quasi-Newton method for unconstrained optimization. The test problems are of the form

$$f(x) = f(x^*) + J(x^*)(x - x^*) + \frac{1}{2} \begin{pmatrix} (x - x^*)^{\mathrm{T}} G_1 (x - x^*) \\ \vdots \\ (x - x^*)^{\mathrm{T}} G_m (x - x^*) \end{pmatrix},$$

constructed so that asymptotic properties could be monitored (see also McKeown [1975a, 1975b]). In all cases considered, $J(x^*)$ had full column rank. The algorithm of Ramsin and Wedin uses the steepest-descent direction, rather than the Gauss-Newton direction, whenever the decrease in the objective is considered unacceptably small. The quasi-Newton routine required an initial estimate H_0 of the Hessian matrix, and the choice $H_0 = J(x_0)^{\mathrm{T}} J(x_0)$ was made on the basis of preliminary tests that showed equal or better performance over

$H_0 = I$. The experiments involved variation of a large number of parameters. Rather than presenting all of their results, they give a summary, together with some representative figures. They conclude that the Gauss-Newton method and the Levenberg-Marquardt method are identical when the asymptotic convergence factor is small, but that the results do not support either method as being better than the other for large asymptotic convergence factors. Also, they find that in instances when the asymptotic convergence factor is large, the quasi-Newton method may be more efficient, although superlinear convergence of the quasi-Newton method was not observed in any of the tests. Ramsin and Wedin maintain that Gauss-Newton should not be used when (i) x_k is close to x^*, and the relative decrease in the size of the gradient is small, when (ii) x_k is not near x^*, and the decrease in the sum of squares relative to the size of the gradient is small, or when (iii) J_k is nearly rank-deficient. Conditions (i) and (ii) are merely indicators of inefficiency for any minimization algorithm, and have little practical significance, since in general the problem of ascertaining the closeness of an iterate to a minimum is as difficult as solving the original problem. As for condition (iii), we will show in Section 4.5 that rapidly convergent Gauss-Newton methods may exist even if J_k is nearly rank-deficient, but that it appears that different rules for defining $rank(J_k)$ must be applied to different types of nonlinear least-squares problems in order to obtain this favorable behavior.

Deuflhard and Apostolescu [1980] suggest selecting a step length for the Gauss-Newton direction based on decreasing the merit function $\|J_k^\dagger f(x)\|_2$ rather than $\|f(x)\|_2^2$, for a class of nonlinear least-squares problems that includes zero-residual problems. The function $J_k^\dagger$ is the *pseudo-inverse* of J_k (see Golub and Van Loan [1983], Chapter 6); $J_k^\dagger f_k$ is another way of representing the minimum l_2-norm solution to $\|J_k p + f_k\|_2$. They reason that the Gauss-Newton direction is the steepest-descent direction for the function $\|J_k^\dagger f(x)\|_2^2$, so that the geometry of the level surfaces defined by $\|J_k^\dagger f(x)\|_2^2$ is more favorable to avoiding small steps in the linesearch. A significant shortcoming of this approach is that is that there are no global convergence results for the method. The merit function depends on x_k, so that a different function is being reduced at each step. Another drawback is that, although the authors claim that numerical experience supports selection of a step length based on $\|J_k^\dagger f(x)\|^2$ for ill-conditioned problems, the transformation $J_k^\dagger$ is not numerically well-defined under these circumstances. Therefore neither the Gauss-Newton search direction, nor the merit function, is numerically well-defined when the columns of J_k are nearly linearly-dependent.

4.4 Descent Conditions for Gauss-Newton Search Directions

Recall from Chapter 3 that the most stable techniques for solving ill-conditioned linear least-squares problems involve orthogonal factorizations : the singular-value decomposition (SVD) for (4.2.4), and the QR factorization for either (4.2.4) or (4.2.5). The purpose of this section is to characterize Gauss-Newton search directions in terms of these factorizations, and state conditions under which they are descent directions for the nonlinear least-squares objective.

4.4.1 Search Directions Computed via the Singular-Value Decomposition

Given the computed singular-value decomposition of the Jacobian

$$J = \begin{cases} U \begin{pmatrix} S & 0 \end{pmatrix} V^{\mathrm{T}}, & \text{if } m < n; \\ USV^{\mathrm{T}}, & \text{if } m = n; \\ U \begin{pmatrix} S \\ 0 \end{pmatrix} V^{\mathrm{T}}, & \text{if } m > n; \end{cases} \tag{4.4.1}$$

where S is diagonal with non-negative diagonal entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)}$, and U and V are orthogonal, define

$$r_{\max} \equiv \max \{ \, i \mid \sigma_i \neq 0 \, \}.$$

Let

$$p_i \equiv \sum_{j=1}^{i} \tau_j v_j; \quad \tau_j \equiv -\frac{u_j^{\mathrm{T}} f}{\sigma_j}; \quad i = 1, 2, \ldots, r_{\max}, \tag{4.4.2}$$

where u_j, v_j are the jth columns of U and V, respectively. The rank of J is estimated to be some value of $r \leq r_{\max}$, so that the vector p_r is then the numerical solution to (4.2.4). The columns of V form an orthonormal basis for $\Re^n$, and τ_j, $j = 1, 2, \ldots, i$, are the components of p_i in terms of this basis, with

$$\|p_i\|_2^2 = \sum_{j=1}^{i} \tau_j^2.$$

The next theorem shows that each p_i is either orthogonal to the gradient $\bar{g} \equiv J^{\mathrm{T}} f$ of the nonlinear least-squares objective, or it is a descent direction.

Theorem 4.4-1 :

For each $i = 1, 2, \ldots, r_{\max}$, *if* p_i *is defined by* (4.4.2), *then*

$$\bar{g}^{\mathrm{T}} p_i = -\sum_{j=1}^{i} \left(u_j^{\mathrm{T}} f\right)^2 \leq 0.$$

Proof:

For the proof, we use the outer product form of the singular-value decomposition :

$$J = \sum_{j=1}^{\min(m,n)} \sigma_j u_j v_j^{\mathrm{T}}.$$

Then

$$\begin{aligned} \bar{g}^{\mathrm{T}} p_i = f^{\mathrm{T}} J p_i &= f^{\mathrm{T}} \left(\sum_{j=1}^{\min(m,n)} \sigma_j u_j v_j^{\mathrm{T}} \right) \sum_{j=1}^{i} \tau_j v_j \\ &= f^{\mathrm{T}} \sum_{j=1}^{i} \sigma_j \tau_j u_j = \sum_{j=1}^{i} \sigma_j \left(\frac{-u_j^{\mathrm{T}} f}{\sigma_j} \right) \left(f^{\mathrm{T}} u_j \right) \\ &= -\sum_{j=1}^{i} \left(u_j^{\mathrm{T}} f\right)^2 \leq 0. \end{aligned}$$

■

4.4.2 Search Directions Computed via the QR Factorization

Now consider the QR factorization of the Jacobian

$$J = \begin{cases} Q \begin{pmatrix} R & 0 \end{pmatrix} P, & \text{if } m < n; \\ QRP, & \text{if } m = n; \\ Q \begin{pmatrix} R \\ 0 \end{pmatrix} P, & \text{if } m > n; \end{cases} \tag{4.4.3}$$

where R is upper triangular, Q and P are orthogonal, and P is a permutation of the column of J. If d_i is the ith diagonal of R, then

$$r_{\max} \equiv \max \{ \, i \mid d_i \neq 0 \, \}$$

is an upper bound for the rank of J. In Chapter 3 it was mentioned that selecting the largest remaining column is a practical strategy from the point of view of determining the rank, because the diagonals then satisfy $|d_1| \geq |d_2| \geq \ldots \geq |d_{\min(m,n)}|$ if $m \geq n$, and tend to reflect the magnitude of the singular values.

For $i = 1, 2, \ldots, r_{\max}$, partition the matrix R as

$$R = \begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \\ 0 & R_{22}^{(i)} \end{pmatrix},$$

where $R_{11}^{(i)}$ is an $i \times i$ upper triangular matrix, and the vector $Q^{\mathrm{T}} f$ as

$$Q^{\mathrm{T}} f \equiv \begin{pmatrix} y_i \\ z_i \end{pmatrix},$$

with y_i consisting of the first i components of $Q^{\mathrm{T}} f$, and z_i consisting of the remaining $m - i$ components. The jth component of $Q^{\mathrm{T}} f$ is $q_j^{\mathrm{T}} f$, where q_j is the jth column of the matrix Q.

4.4.2.1 QR with Column Deletion

If we define

$$p_i \equiv -P^{\mathrm{T}} \begin{pmatrix} R_{11}^{(i)^{-1}} y_i \\ 0 \end{pmatrix}, \tag{4.4.4}$$

and choose $r \leq r_{\max}$ as the rank of J, then p_r is a basic solution to the linear least-squares problem (4.2.3) since $\bar{J}$ in (4.2.5) is completely determined by the column pivoting strategy and the value of r. The following theorem is the analogue of Theorem 4.4-1 for the vectors p_i obtained from (4.4.4).

Theorem 4.4-2:

For each $i = 1, 2, \ldots, r_{\max}$, if p_i is determined by (4.4.4), then

$$\bar{g}^{\mathrm{T}} p_i = -\sum_{j=1}^{i} \left(q_j^{\mathrm{T}} f\right)^2 \leq 0.$$

Proof:

$$
\begin{aligned}
f^{\mathrm{T}} J p_i &= f^{\mathrm{T}} Q \begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \\ 0 & R_{22}^{(i)} \end{pmatrix} P p_i \\
&= f^{\mathrm{T}} Q \begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \\ 0 & R_{22}^{(i)} \end{pmatrix} P \left(-P^{\mathrm{T}} \begin{pmatrix} R_{11}^{(i)^{-1}} y_i \\ 0 \end{pmatrix} \right) \\
&= -f^{\mathrm{T}} Q \begin{pmatrix} y_i \\ 0 \end{pmatrix} = -y_i^{\mathrm{T}} y_i \\
&= -\sum_{j=1}^{i} \left(q_j^{\mathrm{T}} f \right)^2 \leq 0.
\end{aligned}
$$

■

4.4.2.2 Complete Orthogonal Factorization

Finally, we define an $n \times n$ orthogonal matrix V_i and an $i \times i$ upper triangular matrix $\bar{R}_{11}^{(i)}$ by the relation

$$
\begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \end{pmatrix} \equiv \begin{pmatrix} \bar{R}_{11}^{(i)} & 0 \end{pmatrix} V_i,
$$

and let

$$
p_i \equiv -P^{\mathrm{T}} V_i^{\mathrm{T}} \begin{pmatrix} \bar{R}_{11}^{(i)^{-1}} y_i \\ 0 \end{pmatrix}. \tag{4.4.5}
$$

The vector p_r is the solution to (4.2.4) in terms of the complete orthogonal factorization if $r \leq r_{\max}$ is taken to be the rank of J. Whereas the directional derivative of $f^{\mathrm{T}} f$ along p_i as defined by (4.4.2) or (4.4.4) is bounded above by 0, for (4.4.5) it may even be positive, depending on the part of R that is ignored, as shown in the following theorem.

Theorem 4.4-3:

For each $i = 1, 2, \ldots, r_{\max}$, *if* p_i *is defined by* (4.4.5), *then*

$$
\bar{g}^{\mathrm{T}} p_i = -\sum_{j=1}^{i} \left(q_j^{\mathrm{T}} f \right)^2 + z_i^{\mathrm{T}} \begin{pmatrix} 0 & R_{22}^{(i)} \end{pmatrix} P p_i.
$$

Proof:

$$
\begin{aligned}
f^{\mathrm{T}} J p_i &= f^{\mathrm{T}} Q \begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \\ 0 & R_{22}^{(i)} \end{pmatrix} P p_i \\
&= y_i^{\mathrm{T}} \begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \end{pmatrix} P p_i + z_i^{\mathrm{T}} \begin{pmatrix} 0 & R_{22}^{(i)} \end{pmatrix} P p_i \\
&= -y_i^{\mathrm{T}} \begin{pmatrix} R_{11}^{(i)} & R_{12}^{(i)} \end{pmatrix} V_i^{\mathrm{T}} \begin{pmatrix} \bar{R}_{11}^{(i)^{-1}} y_i \\ 0 \end{pmatrix} + z_i^{\mathrm{T}} \begin{pmatrix} 0 & R_{22}^{(i)} \end{pmatrix} P p_i \\
&= -y_i^{\mathrm{T}} \begin{pmatrix} \bar{R}_{11}^{(i)} & 0 \end{pmatrix} \begin{pmatrix} \bar{R}_{11}^{(i)^{-1}} y_i \\ 0 \end{pmatrix} + z_i^{\mathrm{T}} \begin{pmatrix} 0 & R_{22}^{(i)} \end{pmatrix} P p_i \\
&= -y_i^{\mathrm{T}} y_i + z_i^{\mathrm{T}} \begin{pmatrix} 0 & R_{22}^{(i)} \end{pmatrix} P p_i \\
&= -\sum_{j=1}^{i} \left(q_j^{\mathrm{T}} f \right)^2 + z_i^{\mathrm{T}} \begin{pmatrix} 0 & R_{22}^{(i)} \end{pmatrix} P p_i .
\end{aligned}
$$

■

4.4.3 Conclusions

We conclude that if the SVD is used, or if columns are deleted from a QR factorization rather than forming a complete orthogonal factorization, then the resulting Gauss-Newton directions are "safe" in the sense that, in exact arithmetic, they can never be directions of increase for the nonlinear least-squares objective. A Gauss-Newton search direction derived from a complete orthogonal factorization may, however, be an ascent direction for $f^{\mathrm{T}} f$, for some values of $rank(J)$. Regardless of how the Jacobian is factorized, the theorems show that the directional derivative $\bar{g}^{\mathrm{T}} p_i$ can be arbitrarily close to zero far from a stationary point. In the next section, we will see an example of a Gauss-Newton method converging to a point that is not local minimum of $f^{\mathrm{T}} f$ because the search directions become orthogonal to the gradient.

4.5 Performance on Problems with Ill-Conditioned Jacobians

So far we have avoided giving specific rank-estimation criteria for Gauss-Newton methods. In Chapter 3 it was suggested that, for linear least squares, such criteria might include

a lower bound on the singular values, or on the size of the diagonals in the QR factorization, and an upper bound on the norm of the search direction, but we saw that there were instances in which it was virtually impossible to give a numerical definition of rank. Some specific examples will now be given which show that fixed definitions of $rank(J)$ are not generally appropriate for Gauss-Newton methods. In all of the examples, the linear least-squares subproblem (4.2.3) is solved using the SVD. (The LINPACK routine DSVDC [Dongarra et al. (1979)] is used to compute the SVD). Results will not be given for Gauss-Newton methods that use the QR factorization, because the same basic considerations apply in choosing the search direction, and also because in practice the behavior is similar to that observed for the SVD. The linesearch for the examples is taken from the nonlinear programming package NPSOL [Gill et al. (1979), (1986b)].

Recall from Section 4.4.1 that a Gauss-Newton search direction computed from the SVD has the form

$$p_r \equiv \sum_{j=1}^{r} \tau_j v_j; \qquad \tau_j \equiv -\frac{u_j^T f}{\sigma_j}, \tag{4.5.1}$$

for some $r \leq \min\{m, n\}$. The vectors v_j are orthonormal, and τ_j are components of p_r with respect to $\{v_j\}$. If $r < \min\{m, n\}$, then p_r has no component in the space spanned by $\{v_{r+1}, v_{r+2}, \ldots, v_{\min\{m,n\}}\}$.

In the examples, the numerical rank of the Jacobian is defined to be

$$rank(J) \equiv \max \{\, i \mid \sigma_i > \epsilon(1 + \sigma_1) \,\}, \tag{4.5.2}$$

where $\sigma_1, \sigma_2, \ldots$ are the singular values of J in decreasing order of magnitude. This criterion depends only on J and does not take into account the size of the search direction p, the angle between p and the gradient, or the vector f. (See Section 3.4 for a discussion of numerical criteria for estimating rank in linear least-squares problems.)

4.5.1 Chebyquad $n = m = 8$ (# 35a.)

The first example is related to the problem of locating nodes for Chebyschev quadrature [Fletcher (1965); Moré, Garbow, and Hillstrom (1981)]. The example demonstrates that the choice of ϵ in (4.5.1) can be critical.

Gauss-Newton

	$\epsilon = 10^{-14}$	$\epsilon \leq 10^{-15}$
f, J evals.	147	124
iters.	44	19
$\|x^*\|_2$	1.65	1.63
$\|f^*\|_2$	10^{-2}	10^{-1}
$\|\bar{g}^*\|_2$	10^{-11}	10^{-1}
est. err.	10^{-9}	10^{-2}

The algorithm succeeds in finding an approximate minimum when $\epsilon = 10^{-14}$, although it fails when $\epsilon = 10^{-15}$. The problem is rather easily solved by the unconstrained methods of Section 2.6, as shown in the table below.

	MNA		DMNH		NPSOL		DMNG	
f evals.	46	46	14	14	33	35	41	43
J evals.	46	46	11	11	33	35	29	31
iters.	15	15	11	11	19	21	28	30
$\|x^*\|_2$	1.65	1.65	1.65	1.65	1.65	1.65	1.65	1.65
$\|f^*\|_2$	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}
$\|\bar{g}^*\|_2$	10^{-10}	10^{-10}	10^{-9}	10^{-9}	10^{-5}	10^{-7}	10^{-6}	10^{-9}
est. err.	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}

The next two tables trace the progress of the Gauss-Newton methods for $\epsilon = 10^{-14}$ and $\epsilon = 10^{-15}$, respectively.

Gauss-Newton on Problem 35a.

$\epsilon = 10^{-14}$

k	f, J evals.	$\|x_k\|_2$	$\|f_k\|_2$	$\|\bar{g}_k\|_2$	$\|p_k\|_2$	$\bar{g}_k^T p_k$	α_k	*cond* J_k	*rank* J_k
0	8	2.E+00	2.E-01	8.E-01	2.E+00	-4.E-02	7.3E-02	10^2	8
1	16	2.E+00	2.E-01	7.E-01	3.E+00	-3.E-02	1.5E-02	10^2	8
2	24	2.E+00	2.E-01	7.E-01	2.E+00	-3.E-02	1.5E-02	10^2	8
3	32	2.E+00	2.E-01	6.E-01	4.E+00	-3.E-02	3.5E-02	10^2	8
4	35	2.E+00	2.E-01	5.E-01	7.E-01	-3.E-02	3.1E-01	10^2	8
5	37	2.E+00	1.E-01	3.E-01	2.E-01	-1.E-02	2.2E-01	10^1	8
6	41	2.E+00	1.E-01	2.E-01	6.E-01	-1.E-02	1.6E-02	10^2	8
7	47	2.E+00	1.E-01	2.E-01	1.E+01	-9.E-03	5.0E-05	10^3	8
8	54	2.E+00	1.E-01	2.E-01	1.E+02	-9.E-03	4.9E-07	10^4	8
9	62	2.E+00	1.E-01	2.E-01	1.E+03	-9.E-03	4.8E-09	10^5	8
10	69	2.E+00	1.E-01	2.E-01	1.E+04	-9.E-03	5.1E-11	10^6	8
11	76	2.E+00	1.E-01	2.E-01	1.E+05	-9.E-03	5.1E-13	10^7	8
12	83	2.E+00	1.E-01	2.E-01	1.E+06	-9.E-03	5.0E-15	10^8	8
13	90	2.E+00	1.E-01	2.E-01	1.E+07	-9.E-03	4.9E-17	10^9	8
14	97	2.E+00	1.E-01	2.E-01	1.E+08	-9.E-03	4.9E-19	10^{10}	8
15	104	2.E+00	1.E-01	2.E-01	1.E+09	-9.E-03	4.7E-21	10^{11}	8
16	111	2.E+00	1.E-01	2.E-01	1.E+10	-9.E-03	4.7E-23	10^{12}	8
17	118	2.E+00	1.E-01	2.E-01	1.E+11	-9.E-03	4.7E-25	10^{13}	8
18	120	2.E+00	1.E-01	2.E-01	8.E-02	-4.E-03	5.7E-01	10^{14}	7
19	123	2.E+00	8.E-02	2.E-01	8.E-03	-8.E-04	2.1E+00	10^{14}	7
20	124	2.E+00	6.E-02	7.E-02	9.E-03	-5.E-04	1.0E+00	10^{14}	7
21	125	2.E+00	6.E-02	3.E-02	2.E-03	-5.E-05	1.0E+00	10^{14}	7
22	126	2.E+00	6.E-02	1.E-02	1.E-03	-1.E-05	1.0E+00	10^{14}	7
23	127	2.E+00	6.E-02	5.E-03	4.E-04	-2.E-06	1.0E+00	10^{14}	7
24	128	2.E+00	6.E-02	2.E-03	2.E-04	-3.E-07	1.0E+00	10^{14}	7
25	129	2.E+00	6.E-02	8.E-04	6.E-05	-4.E-08	1.0E+00	10^{14}	7
26	130	2.E+00	6.E-02	3.E-04	3.E-05	-2.E-09	1.0E+00	10^{14}	7
27	131	2.E+00	6.E-02	1.E-04	1.E-05	-1.E-09	1.0E+00	10^{14}	7
28	132	2.E+00	6.E-02	5.E-05	4.E-06	-2.E-10	1.0E+00	10^{14}	7
29	133	2.E+00	6.E-02	2.E-05	2.E-06	-2.E-11	1.0E+00	10^{14}	7
30	134	2.E+00	6.E-02	8.E-06	6.E-07	-4.E-12	1.0E+00	10^{14}	7
31	135	2.E+00	6.E-02	3.E-06	2.E-07	-6.E-13	1.0E+00	10^{14}	7
32	136	2.E+00	6.E-02	1.E-06	9.E-08	-9.E-14	1.0E+00	10^{14}	7
33	137	2.E+00	6.E-02	5.E-07	4.E-08	-1.E-14	1.0E+00	10^{14}	7
34	138	2.E+00	6.E-02	2.E-07	1.E-08	-2.E-15	1.0E+00	10^{14}	7
35	139	2.E+00	6.E-02	8.E-08	6.E-09	-3.E-16	1.0E+00	10^{14}	7
36	140	2.E+00	6.E-02	3.E-08	2.E-09	-5.E-17	1.0E+00	10^{14}	7
37	141	2.E+00	6.E-02	1.E-08	9.E-10	-8.E-18	1.0E+00	10^{14}	7
38	142	2.E+00	6.E-02	5.E-09	4.E-10	-1.E-18	1.0E+00	10^{14}	7
39	143	2.E+00	6.E-02	2.E-09	1.E-10	-2.E-19	1.0E+00	10^{14}	7
40	144	2.E+00	6.E-02	7.E-10	5.E-11	-3.E-20	1.0E+00	10^{14}	7
41	145	2.E+00	6.E-02	3.E-10	2.E-11	-5.E-21	1.0E+00	10^{14}	7
42	146	2.E+00	6.E-02	1.E-10	8.E-12	-7.E-22	1.0E+00	10^{14}	7
43	147	2.E+00	6.E-02	4.E-11	3.E-12	-1.E-22	1.0E+00	10^{14}	7
		2.E+00	6.E-02	2.E-11					

Gauss-Newton on Problem **35a.**

$\epsilon = 10^{-15}$

k	f, J evals.	$\|x_k\|_2$	$\|f_k\|_2$	$\|\bar{g}_k\|_2$	$\|p_k\|_2$	$\bar{g}_k^T p_k$	α_k	*cond* J_k	*rank* J_k
0	8	2.E+00	2.E-01	8.E-01	2.E+00	-4.E-02	7.3E-02	10^2	8
1	16	2.E+00	2.E-01	7.E-01	3.E+00	-3.E-02	1.5E-02	10^2	8
2	24	2.E+00	2.E-01	7.E-01	2.E+00	-3.E-02	1.5E-02	10^2	8
3	32	2.E+00	2.E-01	6.E-01	4.E+00	-3.E-02	3.5E-02	10^2	8
4	35	2.E+00	2.E-01	5.E-01	7.E-01	-3.E-02	3.1E-01	10^2	8
5	37	2.E+00	1.E-01	3.E-01	2.E-01	-1.E-02	2.2E-01	10^1	8
6	41	2.E+00	1.E-01	2.E-01	6.E-01	-1.E-02	1.6E-02	10^2	8
7	47	2.E+00	1.E-01	2.E-01	1.E+01	-9.E-03	5.0E-05	10^3	8
8	54	2.E+00	1.E-01	2.E-01	1.E+02	-9.E-03	4.9E-07	10^4	8
9	62	2.E+00	1.E-01	2.E-01	1.E+03	-9.E-03	4.8E-09	10^5	8
10	69	2.E+00	1.E-01	2.E-01	1.E+04	-9.E-03	5.1E-11	10^6	8
11	76	2.E+00	1.E-01	2.E-01	1.E+05	-9.E-03	5.1E-13	10^7	8
12	83	2.E+00	1.E-01	2.E-01	1.E+06	-9.E-03	5.0E-15	10^8	8
13	90	2.E+00	1.E-01	2.E-01	1.E+07	-9.E-03	4.9E-17	10^9	8
14	97	2.E+00	1.E-01	2.E-01	1.E+08	-9.E-03	4.9E-19	10^{10}	8
15	104	2.E+00	1.E-01	2.E-01	1.E+09	-9.E-03	4.7E-21	10^{11}	8
16	111	2.E+00	1.E-01	2.E-01	1.E+10	-9.E-03	4.7E-23	10^{12}	8
17	118	2.E+00	1.E-01	2.E-01	1.E+11	-9.E-03	4.7E-25	10^{13}	8
18	124	2.E+00	1.E-01	2.E-01	1.E+12	-9.E-03	0.0E-01	10^{14}	8

Until iteration 18, the Jacobian has full column rank at each step according to (4.5.1), and it becomes increasingly ill-conditioned as the computation proceeds. The search direction grows very large and approaches orthogonality to the gradient, while the step length decreases. No significant decrease is observed in either $\|f\|_2$ or $\|\bar{g}\|_2$ in iterations 6 - 17. At iteration 18, the two Gauss-Newton methods differ. For $\epsilon = 10^{-14}$, the estimated rank of the Jacobian is reduced to 7, and a significant decrease in the function is achieved. For $\epsilon \leq 10^{-15}$, by (4.5.1) the Jacobian still has full column rank, and the algorithm terminates because $\alpha_k p_k$ is judged to be negligible relative to $\|x_k\|_2$. Detailed information at the start of iteration 18 for the Gauss-Newton methods is given in the next table.

$\epsilon \leq 10^{-14}$; iteration 18

r	σ_r	$\lvert\tau_r\rvert$	$\lVert p_r\rVert_2$	$\lvert\bar{g}^T p_r\rvert$	$\lvert cos(\bar{g}, p_r)\rvert$
1	10^1	10^{-3}	10^{-3}	10^{-4}	10^0
2	10^1	10^{-16}	10^{-3}	10^{-4}	10^0
3	10^0	10^{-16}	10^{-3}	10^{-4}	10^0
4	10^0	10^{-2}	10^{-2}	10^{-3}	10^0
5	10^0	10^{-15}	10^{-2}	10^{-3}	10^0
6	10^0	10^{-1}	10^{-1}	10^{-3}	10^{-1}
7	10^0	10^{-14}	10^{-1}	10^{-3}	10^{-1}
8	10^{-13}	10^{12}	10^{12}	10^{-2}	10^{-13}

It seems reasonable to say that $rank(J) = 7$ rather than $rank(J) = 8$ at this point, because $\sigma_8 \ll \sigma_7$, $\lVert p_8\rVert_2 \gg \lVert p_7\rVert_2$, and $\lvert cos(\bar{g}, p_8)\rvert \ll \lvert cos(\bar{g}, p_7)\rvert$. Hence it is not surprising that it is the method with $\epsilon = 10^{-14}$, rather than the one with $\epsilon = 10^{-15}$, that ultimately makes good progress toward the solution.

The behavior of the Gauss-Newton methods can be explained by comparing the sequence $\{p_k^*\}$ of steps from the iterates to the minimum of the function, to the sequence $\{p_k\}$ of Gauss-Newton steps. The magnitudes of the components of these vectors in terms of the basis $\{v_j(x_k)\}$, for iterations 6 - 18, are listed in the tables below.

components $\{\tau_j^*(x_k)\}$ of $p_k^* = x^* - x_k$ in terms of $\{v_j(x_k)\}$

k	$\lvert\tau_1^*\rvert$	$\lvert\tau_2^*\rvert$	$\lvert\tau_3^*\rvert$	$\lvert\tau_4^*\rvert$	$\lvert\tau_5^*\rvert$	$\lvert\tau_6^*\rvert$	$\lvert\tau_7^*\rvert$	$\lvert\tau_8^*\rvert$
6	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-3}
7	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-4}
8	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-5}
9	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-6}
10	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-7}
11	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-8}
12	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-9}
13	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-10}
14	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-11}
15	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-12}
16	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-13}
17	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-14}
18	10^{-2}	10^{-9}	10^{-8}	10^{-2}	10^{-9}	10^{-2}	10^{-9}	10^{-15}

components $\{\tau_j(x_k)\}$ of p_k in terms of $\{v_j(x_k)\}$

k	$\lvert\tau_1\rvert$	$\lvert\tau_2\rvert$	$\lvert\tau_3\rvert$	$\lvert\tau_4\rvert$	$\lvert\tau_5\rvert$	$\lvert\tau_6\rvert$	$\lvert\tau_7\rvert$	$\lvert\tau_8\rvert$
6	10^{-3}	10^{-17}	10^{-16}	10^{-2}	10^{-14}	10^{-1}	10^{-15}	10^{0}
7	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{1}
8	10^{-3}	10^{-17}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{2}
9	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{3}
10	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{4}
11	10^{-3}	10^{-17}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{5}
12	10^{-3}	10^{-17}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{6}
13	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{7}
14	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{8}
15	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{9}
16	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{10}
17	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{11}
18	10^{-3}	10^{-16}	10^{-16}	10^{-2}	10^{-15}	10^{-1}	10^{-14}	10^{12}

The step p_k^* to the minimum approaches orthogonality to $v_8(x_k)$, while the Gauss-Newton search direction becomes dominated by the component in the direction of $v_8(x_k)$ due to the ill-conditioning in the Jacobian. Hence, by iteration 18, p_k is almost orthogonal to p_k^*. The question of when to say that J has rank 7 rather than rank 8 is a difficult one. If full column rank is assumed until the search direction becomes numerically orthogonal to the gradient then the method may become very inefficient (see iterations 6 - 18, where about seven function evaluations are required per iteration). On the other hand, if the step to the minmum has a component in the estimated null space $null(J)$, underestimating $rank(J)$ will inhibit decrease in $null(J)$, because the Gauss-Newton search direction will be orthogonal to $null(J)$.

4.5.2 Matrix Square Root 1 $n = m = 4$ (# 36a.)

Another instance in which Gauss-Newton methods encounter ill-conditioned Jacobians is the problem of finding the square root of a given (square) matrix (see the Appendix). Although the matrix in question is only of order 2, the problem is a difficult one for the unconstrained methods, as shown in the table below. (For more detail, see Section 2.6).

	MNA		DMNH		NPSOL		DMNG	
f evals.	4001	4001	4000	4000	786	2618	4000	4000
J evals.	4001	4001	2198	2198	786	2618	2880	2880
iters.	2664	2664	2197	2197	477	1437	2880	2880
$\|\|x^*\|\|_2$	50.4	50.4	17.8	17.8	9.22	10.1	17.0	17.0
$\|\|f^*\|\|_2$	10^{-9}	10^{-9}	10^{-6}	10^{-6}	10^{-5}	10^{-5}	10^{-6}	10^{-6}
$\|\|\bar{g}^*\|\|_2$	10^{-9}	10^{-9}	10^{-6}	10^{-6}	10^{-5}	10^{-7}	10^{-6}	10^{-6}
est. err.	10^{-19}	10^{-19}	10^{-12}	10^{-12}	10^{-9}	10^{-9}	10^{-11}	10^{-11}
conv.	F LIM.	F LIM.	F LIM.	F LIM.			F LIM.	F LIM.

MNA is just Newton's method in this case, since the exact Hessian matrix is never modified, although it does become ill-conditioned, with a condition number of order 10^{11} at the solution. In the Gauss-Newton methods, the Jacobian does becomes ill-conditioned, but unlike the previous problem, a solution is obtained only when the Jacobian is assumed to have full rank at each iteration. A summary of the results for $\epsilon = 10^{-10}$ and $\epsilon \leq 10^{-11}$ are given in the following table.

Gauss-Newton

	$\epsilon = 10^{-10}$	$\epsilon \leq 10^{-11}$
f, J evals.	4004	95
iters.	473	39
$\|\|x^*\|\|_2$	10^1	50.0
$\|\|f^*\|\|_2$	10^{-7}	10^{-16}
$\|\|\bar{g}^*\|\|_2$	10^{-6}	10^{-15}
est. err.	10^{-15}	10^{-33}
conv.	F LIM.	

The next two tables trace the iterations of the Gauss-Newton method for $\epsilon = 10^{-10}$ and $\epsilon = 10^{-11}$, respectively.

Gauss-Newton on Problem 36a.

$$\epsilon = 10^{-10}$$

k	f, J evals.	$\|x_k\|_2$	$\|f_k\|_2$	$\|\bar{g}_k\|_2$	$\|p_k\|_2$	$\bar{g}_k^T p_k$	α_k	*cond* J_k	*rank* J_k
0	2	1.E+00	2.E+00	3.E+00	9.E-01	-3.E+00	1.0E+00	10^0	4
1	3	9.E-01	6.E-01	6.E-01	8.E-01	-4.E-01	1.0E+00	10^1	4
2	5	1.E+00	4.E-01	7.E-01	1.E+00	-1.E-01	5.5E-01	10^2	4
3	7	2.E+00	3.E-01	8.E-01	2.E+00	-8.E-02	4.5E-01	10^3	4
4	9	3.E+00	2.E-01	8.E-01	2.E+00	-5.E-02	4.0E-01	10^4	4
5	11	4.E+00	2.E-01	9.E-01	3.E+00	-3.E-02	3.7E-01	10^4	4
6	13	5.E+00	2.E-01	1.E+00	3.E+00	-2.E-02	3.4E-01	10^5	4
7	15	6.E+00	1.E-01	1.E+00	4.E+00	-2.E-02	3.3E-01	10^5	4
8	17	7.E+00	1.E-01	1.E+00	4.E+00	-1.E-02	3.2E-01	10^6	4
9	19	8.E+00	1.E-01	1.E+00	5.E+00	-1.E-02	3.1E-01	10^6	4
10	22	1.E+01	1.E-01	1.E+00	6.E+00	-1.E-02	2.0E-01	10^7	4
11	25	1.E+01	9.E-02	1.E+00	6.E+00	-8.E-03	1.8E-01	10^7	4
12	28	1.E+01	8.E-02	1.E+00	7.E+00	-7.E-03	1.7E-01	10^7	4
13	31	1.E+01	7.E-02	1.E+00	7.E+00	-6.E-03	1.6E-01	10^7	4
14	34	1.E+01	7.E-02	1.E+00	8.E+00	-5.E-03	1.5E-01	10^8	4
15	37	2.E+01	6.E-02	1.E+00	8.E+00	-4.E-03	1.5E-01	10^8	4
16	40	2.E+01	6.E-02	1.E+00	8.E+00	-3.E-03	1.4E-01	10^8	4
17	43	2.E+01	5.E-02	1.E+00	9.E+00	-3.E-03	1.4E-01	10^8	4
18	46	2.E+01	5.E-02	1.E+00	9.E+00	-3.E-03	1.4E-01	10^8	4
19	49	2.E+01	5.E-02	1.E+00	9.E+00	-2.E-03	1.3E-01	10^8	4
20	52	2.E+01	4.E-02	1.E+00	1.E+01	-2.E-03	1.3E-01	10^9	4
21	55	2.E+01	4.E-02	1.E+00	1.E+01	-2.E-03	1.3E-01	10^9	4
22	58	2.E+01	4.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
23	61	3.E+01	4.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
24	64	3.E+01	3.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
25	67	3.E+01	3.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
26	70	3.E+01	3.E-02	1.E+00	1.E+01	-9.E-04	1.4E-01	10^9	4
27	73	3.E+01	3.E-02	1.E+00	1.E+01	-7.E-04	1.4E-01	10^9	4
28	76	3.E+01	3.E-02	1.E+00	1.E+01	-6.E-04	1.5E-01	10^{10}	4
29	79	3.E+01	3.E-02	1.E+00	1.E+01	-6.E-04	1.6E-01	10^{10}	4
30	82	4.E+01	2.E-02	1.E+00	9.E+00	-5.E-04	1.7E-01	10^{10}	4
31	85	4.E+01	2.E-02	1.E+00	9.E+00	-4.E-04	1.9E-01	10^{10}	4
32	86	4.E+01	2.E-02	1.E+00	3.E-04	-3.E-04	1.0E+00	10^{10}	3
33	93	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	2.1E-04	10^{10}	4
34	98	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	9.9E-05	10^{10}	4
35	103	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	9.9E-05	10^{10}	4
36	108	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	9.9E-05	10^{10}	4
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
470	3986	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	2.2E-05	10^{10}	4
471	3995	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	2.2E-05	10^{10}	4
472	4004	4.E+01	9.E-08	4.E-06	6.E+00	-8.E-15	2.2E-05	10^{10}	4
		4.E+01	9.E-08	4.E-06					

Gauss-Newton on Problem **36a.**

$\epsilon = 10^{-11}$

k	f, J evals.	$\|x_k\|_2$	$\|f_k\|_2$	$\|\bar{g}_k\|_2$	$\|p_k\|_2$	$\bar{g}_k^T p_k$	α_k	*cond* J_k	*rank* J_k
0	2	1.E+00	2.E+00	3.E+00	9.E-01	-3.E+00	1.0E+00	10^0	4
1	3	9.E-01	6.E-01	6.E-01	8.E-01	-4.E-01	1.0E+00	10^1	4
2	5	1.E+00	4.E-01	7.E-01	1.E+00	-1.E-01	5.5E-01	10^2	4
3	7	2.E+00	3.E-01	8.E-01	2.E+00	-8.E-02	4.5E-01	10^3	4
4	9	3.E+00	2.E-01	8.E-01	2.E+00	-5.E-02	4.0E-01	10^4	4
5	11	4.E+00	2.E-01	9.E-01	3.E+00	-3.E-02	3.7E-01	10^4	4
6	13	5.E+00	2.E-01	1.E+00	3.E+00	-2.E-02	3.4E-01	10^5	4
7	15	6.E+00	1.E-01	1.E+00	4.E+00	-2.E-02	3.3E-01	10^5	4
8	17	7.E+00	1.E-01	1.E+00	4.E+00	-1.E-02	3.2E-01	10^6	4
9	19	8.E+00	1.E-01	1.E+00	5.E+00	-1.E-02	3.1E-01	10^6	4
10	22	1.E+01	1.E-01	1.E+00	6.E+00	-1.E-02	2.0E-01	10^7	4
11	25	1.E+01	9.E-02	1.E+00	6.E+00	-8.E-03	1.8E-01	10^7	4
12	28	1.E+01	8.E-02	1.E+00	7.E+00	-7.E-03	1.7E-01	10^7	4
13	31	1.E+01	7.E-02	1.E+00	7.E+00	-6.E-03	1.6E-01	10^7	4
14	34	1.E+01	7.E-02	1.E+00	8.E+00	-5.E-03	1.5E-01	10^8	4
15	37	2.E+01	6.E-02	1.E+00	8.E+00	-4.E-03	1.5E-01	10^8	4
16	40	2.E+01	6.E-02	1.E+00	8.E+00	-3.E-03	1.4E-01	10^8	4
17	43	2.E+01	5.E-02	1.E+00	9.E+00	-3.E-03	1.4E-01	10^8	4
18	46	2.E+01	5.E-02	1.E+00	9.E+00	-3.E-03	1.4E-01	10^8	4
19	49	2.E+01	5.E-02	1.E+00	9.E+00	-2.E-03	1.3E-01	10^8	4
20	52	2.E+01	4.E-02	1.E+00	1.E+01	-2.E-03	1.3E-01	10^9	4
21	55	2.E+01	4.E-02	1.E+00	1.E+01	-2.E-03	1.3E-01	10^9	4
22	58	2.E+01	4.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
23	61	3.E+01	4.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
24	64	3.E+01	3.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
25	67	3.E+01	3.E-02	1.E+00	1.E+01	-1.E-03	1.3E-01	10^9	4
26	70	3.E+01	3.E-02	1.E+00	1.E+01	-9.E-04	1.4E-01	10^9	4
27	73	3.E+01	3.E-02	1.E+00	1.E+01	-7.E-04	1.4E-01	10^9	4
28	76	3.E+01	3.E-02	1.E+00	1.E+01	-6.E-04	1.5E-01	10^{10}	4
29	79	3.E+01	3.E-02	1.E+00	1.E+01	-6.E-04	1.6E-01	10^{10}	4
30	82	4.E+01	2.E-02	1.E+00	9.E+00	-5.E-04	1.7E-01	10^{10}	4
31	85	4.E+01	2.E-02	1.E+00	9.E+00	-4.E-04	1.9E-01	10^{10}	4
32	87	4.E+01	2.E-02	1.E+00	8.E+00	-3.E-04	3.2E-01	10^{10}	4
33	89	4.E+01	2.E-02	9.E-01	7.E+00	-3.E-04	3.8E-01	10^{10}	4
34	91	4.E+01	1.E-02	8.E-01	5.E+00	-2.E-04	5.3E-01	10^{10}	4
35	92	5.E+01	1.E-02	6.E-01	3.E+00	-1.E-04	1.0E+00	10^{10}	4
36	93	5.E+01	4.E-03	3.E-01	3.E-01	-1.E-05	1.0E+00	10^{11}	4
37	94	5.E+01	1.E-05	7.E-04	6.E-04	-1.E-10	1.0E+00	10^{11}	4
38	95	5.E+01	2.E-11	1.E-09	1.E-09	-4.E-22	1.0E+00	10^{11}	4
		5.E+01	6.E-17	4.E-15					

The first difference between the two methods occurs at iteration 33. Data available from the SVD at the start of the iteration is shown in the following table. (See Section 4.2.1 for an explanation of the notation.)

$\epsilon \leq 10^{-10}$; iteration 33

r	σ_r	$\lvert\tau_r\rvert$	$\lVert p_r\rVert_2$	$\lvert\bar{g}^T p_r\rvert$	$\lvert cos(\bar{g}, p_r)\rvert$
1	10^2	10^{-4}	10^{-4}	10^{-4}	10^0
2	10^2	10^{-4}	10^{-4}	10^{-3}	10^0
3	10^{-2}	10^{-15}	10^{-4}	10^{-3}	10^0
4	10^{-8}	10^1	10^1	10^{-3}	10^{-5}

The case for saying that $rank(J) = 3$ appears to be fairly strong. There is a large gap between σ_4 and σ_3, and $|cos(\bar{g}, p_4)|$ is significantly smaller than $|cos(\bar{g}, p_3)|$. Moreover, it would appear that the step taken when $\epsilon = 10^{-10}$ and $rank(J) = 3$ is better, in the sense that the reduction in the values of both $\|f\|_2$ and $\|g\|_2$ is appreciably greater than the reduction achieved when $\epsilon = 10^{-11}$ and $rank(J) = 4$. On the other hand, $|p_4|$ is not especially large in magnitude for either choice of rank. For $\epsilon = 10^{-10}$, the algorithm subsequently makes unacceptably slow progress, while for $\epsilon = 10^{-11}$, quadratic convergence occurs after a few more iterations.

To see why no further progress can be made for $\epsilon = 10^{-10}$, consider the following table of information on the state of the method at the start of iteration 34.

$\epsilon \leq 10^{-10}$; iteration 34

r	σ_r	$\lvert\tau_r\rvert$	$\lVert p_r\rVert_2$	$\lvert\bar{g}^T p_r\rvert$	$\lvert cos(\bar{g}, p_r)\rvert$
1	10^2	10^{-9}	10^{-4}	10^{-4}	10^0
2	10^2	10^{-9}	10^{-4}	10^{-4}	10^0
3	10^{-2}	10^{-16}	10^{-4}	10^{-4}	10^0
4	10^{-8}	10^1	10^1	10^{-4}	10^{-5}

The singular values are nearly the same as those of the previous iteration, but the change is enough to have $rank(J) = 4$ rather than $rank(J) = 3$ according to (4.5.1). The value of $\|f\|_2$ has decreased significantly after iteration 33 : $|\tau_1|$ and $|\tau_2|$, which were the dominant components just prior to iteration 33, are much smaller at the start of iteration 34, although

$|\tau_3|$ and $|\tau_4|$ are essentially unchanged. As a consequence, $\|p_4\|_2$ is now very large relative to $\|p_3\|_2$, but $|cos(\bar{g}, p_4)|$ is small since v_4 is close to being orthogonal to g. In fact, if (4.5.1) is disregarded and $rank(J)$ forced to be 3, the method will converge to a local minimum in one step.

As in the previous section, we compare the sequence $\{p_k^*\}$ of steps from the iterates to the minimum of the function, to the sequence $\{p_k\}$ of Gauss-Newton steps.

components $\{\tau_j^*(x_k)\}$ of $p_k^* = x^* - x_k$ in terms of $\{v_j(x_k)\}$

k	$\lvert\tau_1^*\rvert$	$\lvert\tau_2^*\rvert$	$\lvert\tau_3^*\rvert$	$\lvert\tau_4^*\rvert$
28	10^{-3}	10^{-3}	10^{-15}	10^{1}
29	10^{-3}	10^{-3}	10^{-16}	10^{1}
30	10^{-3}	10^{-3}	10^{-14}	10^{1}
31	10^{-3}	10^{-3}	10^{-14}	10^{1}
32	10^{-3}	10^{-3}	10^{-15}	10^{1}

components $\{\tau_j(x_k)\}$ of p_k in terms of $\{v_j(x_k)\}$

k	$\lvert\tau_1\rvert$	$\lvert\tau_2\rvert$	$\lvert\tau_3\rvert$	$\lvert\tau_4\rvert$
28	10^{-4}	10^{-4}	10^{-15}	10^{1}
29	10^{-4}	10^{-4}	10^{-15}	10^{1}
30	10^{-4}	10^{-4}	10^{-15}	10^{1}
31	10^{-4}	10^{-4}	10^{-14}	10^{1}
32	10^{-4}	10^{-4}	10^{-15}	10^{1}

Taking $rank(J) = 3$ is a bad strategy, in this case, because the solution lies mainly in the direction of $v_4(x_k)$.

4.5.3 Watson $n = 20$; $m = 31$ (# 20d.)

The final example for this section is a problem that might seem to be very hard for Gauss-Newton methods. In Watson's problem [Brent (1973); Moré, Garbow, and Hillstrom (1981)], a polynomial of degree n is fitted to approximate the solution of an ordinary differential equation. The Jacobian matrix for $n = 20$ has singular values of order 10^2, 10^1, 10^1, 10^0, 10^0, 10^0, 10^{-1}, 10^{-1}, 10^{-2}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-11}, and 10^{-12} at the origin. Yet there is very little difficulty in obtaining a solution, starting from $x_0 = 0$, for a wide range of values of ϵ, as shown in the table below.

Gauss-Newton

ϵ	10^{-8}	10^{-9}	10^{-10}	$10^{-11}; 10^{-12}$	10^{-13}	$\geq 10^{-14}$
f, J evals.	6	6	6	6	6	6
iters.	5	5	5	5	5	5
$\|x^*\|_2$	1.07	1.11	1.55	5.21	29.2	247.
$\|f^*\|_2$	10^{-8}	10^{-8}	10^{-9}	10^{-9}	10^{-10}	10^{-10}
$\|\bar{g}^*\|_2$	10^{-14}	10^{-14}	10^{-14}	10^{-12}	10^{-14}	10^{-12}

Gauss-Newton compares favorably on this problem with results for the unconstrained methods of Section 2.6, which are summarized in the next table.

	MNA		DMNH		NPSOL		DMNG	
f evals.	(352)	(251)	40	(251)	76	200	109	132
J evals.	(352)	(251)	23	(92)	76	200	107	118
iters.	(189)	(135)	22	(91)	38	99	106	118
$\|x^*\|_2$	10^6	10^6	1.10	1.21	1.06	1.06	1.06	1.06
$\|f^*\|_2$	10^{-3}	10^{-3}	10^{-8}	10^{-8}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
$\|\bar{g}^*\|_2$	10^{-5}	10^{-5}	10^{-14}	10^{-15}	10^{-5}	10^{-8}	10^{-11}	10^{-12}
est. err.	10^{-5}	10^{-5}	10^{-16}	10^{-16}	10^{-8}	10^{-11}	10^{-12}	10^{-13}
conv.	TIME	TIME		LOOP				

In MNA, the Hessian matrix is nearly singular (but not indefinite) at every iteration, with condition number ranging from 10^{11} to 10^{15}, and it is modified at every step. The trust-region algorithm DMNH, which also uses exact second derivatives, loops for some values of the parameters in the termination criteria.

Watson's problem has a number of local minima, so that the value of the Gauss-Newton solution is dependent on ϵ. Nothing can be said concerning which of the local minima is the "better" one without knowing how the solution is going to be used. For the larger values of ϵ, solutions are obtained that are small in magnitude and hence closer to the starting value, because lower values of the rank restrict the size of the search directions. On the other hand, the final value of the sum of squares is smaller for smaller values of ϵ, because the objective function is being decreased in a larger subspace at each step. Details of the Gauss-Newton iterations are given below.

Gauss-Newton on Problem 20d.

k	f, J evals.	$\|x_k\|_2$	$\|f_k\|_2$	$\|\bar{g}_k\|_2$	$\|p_k\|_2$	$\bar{g}_k^T p_k$	α_k	*cond* J_k	*rank* J_k
				$\epsilon = 10^{-8}$					
0	2	0.E+00	5.E+00	2.E+02	1.E+00	-3.E+01	1.0E+00	10^{14}	15
1	3	1.E+00	3.E+00	1.E+02	4.E-01	-6.E+00	1.0E+00	10^{13}	15
2	4	1.E+00	4.E-01	2.E+01	5.E-02	-2.E-01	1.0E+00	10^{13}	15
3	5	1.E+00	2.E-03	1.E-01	5.E-02	-4.E-06	1.0E+00	10^{13}	15
4	6	1.E+00	3.E-08	5.E-07	3.E-05	-2.E-16	1.0E+00	10^{13}	15
		1.E+00	3.E-08	2.E-14					
				$\epsilon = 10^{-9}$					
0	2	0.E+00	5.E+00	2.E+02	1.E+00	-3.E+01	1.0E+00	10^{14}	16
1	3	1.E+00	3.E+00	1.E+02	4.E-01	-6.E+00	1.0E+00	10^{13}	16
2	4	1.E+00	4.E-01	2.E+01	1.E-01	-2.E-01	1.0E+00	10^{13}	16
3	5	1.E+00	2.E-03	1.E-01	2.E-01	-4.E-06	1.0E+00	10^{13}	16
4	6	1.E+00	2.E-08	5.E-07	1.E-04	-2.E-16	1.0E+00	10^{13}	16
		1.E+00	1.E-08	7.E-15					
				$\epsilon = 10^{-10}$					
0	2	0.E+00	5.E+00	2.E+02	1.E+00	-3.E+01	1.0E+00	10^{14}	17
1	3	1.E+00	3.E+00	1.E+02	4.E-01	-6.E+00	1.0E+00	10^{13}	17
2	4	1.E+00	4.E-01	2.E+01	5.E-01	-2.E-01	1.0E+00	10^{13}	17
3	5	1.E+00	2.E-03	1.E-01	7.E-01	-4.E-06	1.0E+00	10^{13}	17
4	6	2.E+00	1.E-08	5.E-07	6.E-04	-2.E-16	1.0E+00	10^{13}	17
		2.E+00	4.E-09	2.E-14					
				$\epsilon = 10^{-11}; 10^{-12}$					
0	2	0.E+00	5.E+00	2.E+02	1.E+00	-3.E+01	1.0E+00	10^{14}	18
1	3	1.E+00	3.E+00	1.E+02	4.E-01	-6.E+00	1.0E+00	10^{13}	18
2	4	1.E+00	4.E-01	2.E+01	2.E+00	-2.E-01	1.0E+00	10^{13}	18
3	5	2.E+00	2.E-03	1.E-01	3.E+00	-4.E-06	1.0E+00	10^{13}	18
4	6	5.E+00	1.E-08	5.E-07	4.E-03	-2.E-16	1.0E+00	10^{13}	18
		5.E+00	1.E-09	6.E-13					
				$\epsilon = 10^{-13}$					
0	2	0.E+00	5.E+00	2.E+02	1.E+00	-3.E+01	1.0E+00	10^{14}	19
1	3	1.E+00	3.E+00	1.E+02	4.E-01	-6.E+00	1.0E+00	10^{13}	19
2	4	1.E+00	4.E-01	2.E+01	1.E+01	-2.E-01	1.0E+00	10^{13}	19
3	5	1.E+00	2.E-03	1.E-01	2.E+01	-4.E-06	1.0E+00	10^{13}	19
4	6	3.E+00	1.E-08	5.E-07	3.E-02	-2.E-16	1.0E+00	10^{13}	19
		3.E+00	5.E-10	3.E-14					
				$\epsilon = 10^{-14}$					
0	2	0.E+00	5.E+00	2.E+02	1.E+00	-3.E+01	1.0E+00	10^{14}	20
1	3	1.E+00	3.E+00	1.E+02	4.E-01	-6.E+00	1.0E+00	10^{13}	20
2	4	1.E+00	4.E-01	2.E+01	8.E+01	-2.E-01	1.0E+00	10^{13}	20
3	5	8.E+01	2.E-03	1.E-01	2.E+02	-4.E-06	1.0E+00	10^{13}	20
4	6	2.E+02	1.E-08	5.E-07	3.E-01	-2.E-16	1.0E+00	10^{13}	20
		2.E+02	2.E-10	4.E-12					

The condition number of the Jacobian remains very large throughout, yet the search direction is never especially large regardless of the choice of rank, because the sequence $\{|u_i^T f|\}$ is monotonically decreasing at about the same rate as the singular values (see (4.5.1)). The unit step gives sufficient decrease in every instance, on account of the many local minima. Moreover, there is superlinear convergence for each value of ϵ, despite the fact that p becomes very close to being orthogonal to the gradient, with $|cos(\bar{g}, p)|$ ranging from 10^{-5} for $\epsilon = 10^{-8}$, to 10^{-9} for $\epsilon \geq 10^{-14}$ in the final step.

4.6 An Example of Poor Performance on a Well-Conditioned Zero-Residual Problem

On problems with well-conditioned Jacobians, Gauss-Newton methods are globally convergent, and they are locally quadratically convergent if in addition the residuals vanish at the solution (see the introduction to this chapter). It is generally believed that Gauss-Newton methods will work well on zero- or small-residual problems in which the Jacobian is never ill-conditioned. In this section, we exhibit a zero-residual problem on which Gauss-Newton performs poorly, although $cond(J_k)$ never exceeds 5×10^3. The example used is the following modification of Rosenbrock's Function [Moré, Garbow, and Hillstrom (1981), p. 21].

Modified Rosenbrock Function $n = m = 2$

$$\phi_1(x) = 100(x_2 - x_1^2)$$

$$\phi_2(x) = 1 - x_1$$

$$x_0 = (0, 0)$$

$$f(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{at } (1, 1)$$

The starting point $(0, 0)$ lies at the bottom of a curved steep-sided valley in which the solution $(1, 1)$ also lies. The following table gives the results for Gauss-Newton and Newton's method on this problem.

Modified Rosenbrock $n = m = 2$; $x_0 = (0, 0)$

	Gauss-Newton	Newton's Method
f, J evals.	467	77
iters.	100	50
$\|x^*\|_2$	1.41	1.41
$\|f^*\|_2$	10^{-15}	10^{-13}
$\|\bar{g}^*\|_2$	10^{-13}	10^{-12}
est. err.	10^{-30}	10^{-26}
conv.	ABS. F, G	G

The linesearch from the nonlinear programming package NPSOL [Gill et al. (1979); (1986b)] is used for both methods. Newton's method can be applied without modification, since the Hessian, as well as the Jacobian, is well-conditioned. In this case, Gauss-Newton is Newton's method for nonlinear equations, because $n = m$. Contour plots of the progress of the two methods are given at the end of this section.

The minimum of the Gauss-Newton model (4.2.1) lies well outside the valley in which the starting value and minimum are located, at least until the iterates are very close to the solution. The univariate function $\Phi(\alpha) = \|f(x_k + \alpha p_k)\|_2^2$ actually has a maximum at $\alpha = 1$ for $\alpha \in [0, 1]$, rather than a minimum as predicted by the quadratic model; moreover, the function rises very steeply from the valley floor to the maximum. Hence a significant number of function evaluations are required in the linesearch in order to minimize $\Phi(\alpha)$, and, initially, rather small steps are taken along the search directions. Strategies for improving the efficiency of the method include decreasing the maximum steplength $\alpha^{\max}$ and relaxing the parameter η in (2.4.4). For example, if N_k is the number of function evaluations required to determine α_k, and the following scheme is used to define $\alpha_k^{\max}$

$$\alpha_k^{\max} = \gamma_k(1 + \|x_k\|_2)$$

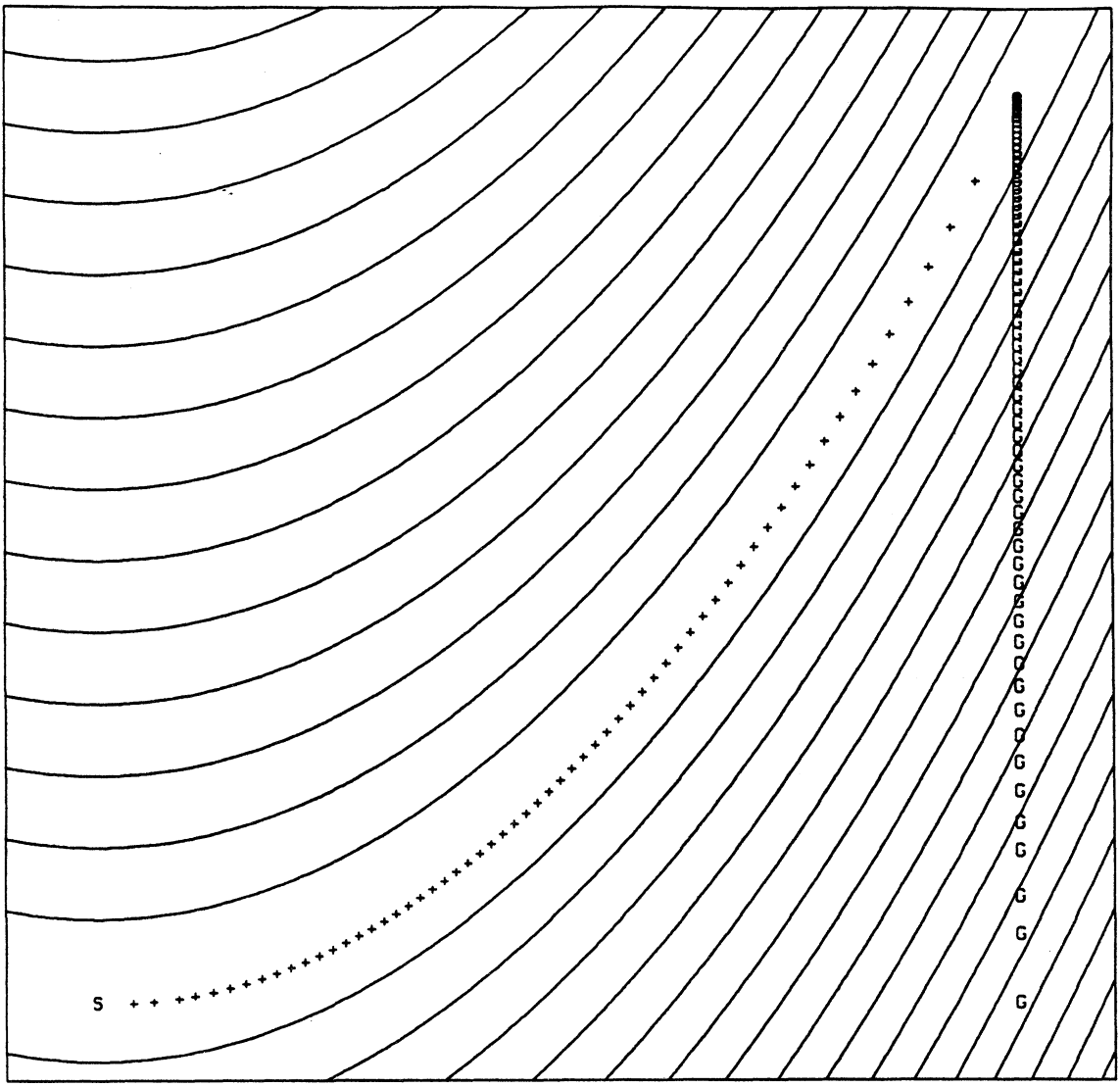
$$\gamma_0 = 1.0$$

$$\gamma_k = \begin{cases} 2\gamma_{k-1} & \text{if } \alpha_{k-1} = \alpha_{k-1}^{\max} \\ \gamma_{k-1} & \text{if } \alpha_{k-1} \neq \alpha_{k-1}^{\max} \text{ and } N_{k-1} \leq 2 \\ \gamma_{k-1}/2 & \text{if } \alpha_{k-1} \neq \alpha_{k-1}^{\max} \text{ and } N_{k-1} > 2, \end{cases}$$

then the Gauss-Newton method solves the problem in only 63 iterations and 135 function evaluations with $\eta = 0.5$. By contrast, the relatively efficient performance of Newton's

method can be explained by the fact that the minimum of the Newton quadratic m
very near the curve along the valley floor connecting $(0,0)$ to $(1,1)$ (which is fol
the iterates of both methods), at all iterations except the first one.

Performance on the Modified Rosenbrock Function



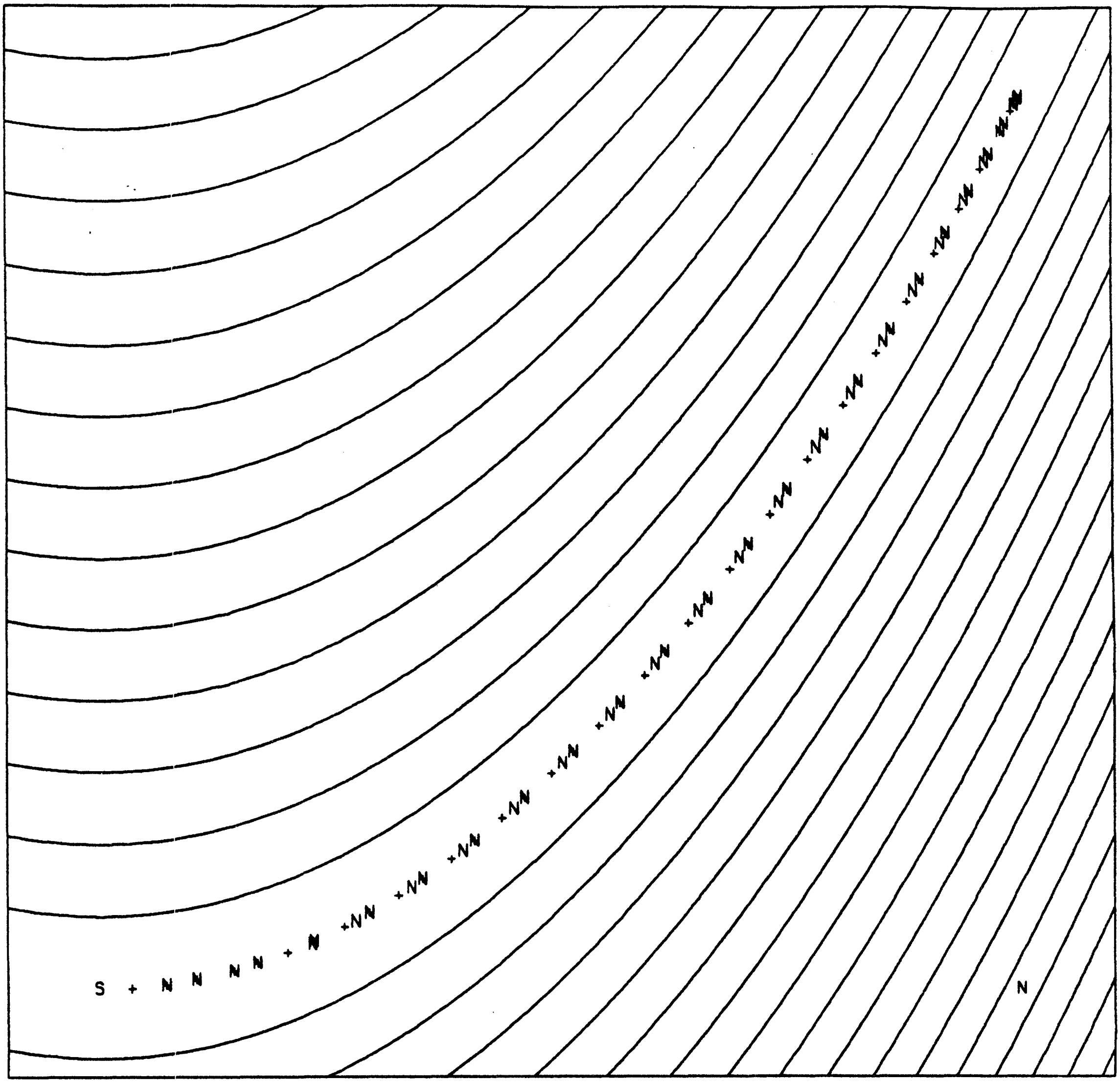
Gauss-Newton Method on the Modified Rosenbrock Function

starting value : $x_0 = (0, 0)$

solution : $x^* = (1, 1)$

G - $x_k + p_k$

\+ - $x_k + \alpha_k p_k$



Newton's Method on the Modified Rosenbrock Function

starting value : $x_0 = (0, 0)$

solution : $x^* = (1, 1)$

N - $x_k + p_k$

+ - $x_k + \alpha_k p_k$

4.7 Numerical Results

4.7.1 Software and Algorithm

In this section numerical results are given for the test problems described in the Appendix. The software package LSSOL [Gill et al.(1986a)] is used to solve the linear least-squares subproblem (4.2.4). The linesearch procedure used for the numerical examples in this section, and also in Sections 4.3 and 4.4, requires both function and gradient information. It is taken from the nonlinear programming code NPSOL [Gill et al. (1979); (1986b)]. As in Chapter 2, presentation of these results is intended primarily for comparison with specialized methods for nonlinear least-squares, so that discussion is postponed until Section 5.7.

4.7.2 Parameters

Parameters in LSSOL were kept at their default values with the following exceptions :

Rank Tolerance	-	varied, see tables
Infinite Bound Size	-	10^{20}

See Gill et al. [1986a] for details concerning the parameters.

In addition, the following parameters are chosen for the linesearch :

η	-	0.5
$\alpha_{\max}$	-	$\min\{(100(1+\|\|x\|\|_2)+1)/\|\|p\|\|_2, 10^{20}\}$ †

† In some cases the default value $\alpha_{\max}$ was too large and overflow occurred during function evaluation in the linesearch. These cases are indicated in the tables by giving the value $\gamma < 100$ such that $\alpha_{\max} = \min\{(\gamma(1+\|x\|_2)+1)/\|p\|_2, 10^{20}\}$ that was subsequently used to obtain the results in the column labeled "step fac.".

See Section 2.4.1 for a discussion of the linesearch parameters.

4.7.3 Convergence Criteria

Convergence is judged to have occurred at the kth iterate if either

$$\|f_k\|_2 \leq \epsilon_M^{0.9} \tag{4.7.1}$$

or

$$\|\bar{g}_k\|_2 \leq \epsilon_M^{2/3}(1 + \|f_k\|_2). \tag{4.7.2}$$

The algorithm is also terminated if there is a negligible change in x,

$$\alpha_k \|p_k\|_2 \leq \epsilon_M^{0.9}(1 + \|x_k\|_2), \tag{4.7.3}$$

where α_k is the step length determined by the linesearch.

4.7.4 Table Information

In the tables, the following notation is used to describe conditions under which the algorithm terminates :

ABS. F	-	(4.7.1)
G	-	(4.7.2)
X	-	(4.7.3)
F LIM.	-	function evaluation limit reached

In the tables, we include the quantity

$$\frac{\|f^*\|_2^2 - \|f_{best}\|_2^2}{1 + \|f_{best}\|_2^2}, \tag{4.7.4}$$

where f^* is the value of f at the point of termination, and $\|f_{best}\|_2$ is the best available estimate of the norm of the solution, in order to get some idea of the error in $\|f^*\|_2$. For those problems that have nonzero residuals, the value of $\|f_{best}\|_2$ is given to six figures of accuracy, rounded down.

For further details on the numerical tests, see Section 1.3, as well as the individual description of each method that follows. For information on the test problems, see the Appendix.

Numerical Results for the Gauss-Newton Methods

	n	m	rank tol.	step fac.	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
1.^{0}	2	2	1.49×10^{-8}		34	13	1.41	10^{-16}	10^{-15}	10^{-32}	ABS. F, G
			2.23×10^{-16}		34	13	1.41	10^{-16}	10^{-15}	10^{-32}	ABS. F, G
2.^{0}	2	2	1.49×10^{-8}		138	33	11.4	10^{1}	10^{-6}	10^{1}	X
			2.23×10^{-16}		225	38	11.4	10^{1}	10^{-13}	10^{1}	G
3.^{0}	2	2	1.49×10^{-8}		31	16	7.22	10^{-3}	10^{-7}	10^{-7}	X
			2.23×10^{-16}		44	24	9.11	10^{-16}	10^{-11}	10^{-33}	ABS. F, G
4.^{0}	2	3	1.49×10^{-8}		56	14	10^{6}	10^{-22}	10^{-22}	10^{-44}	ABS. F, G, X
			2.23×10^{-16}		56	14	10^{6}	10^{-22}	10^{-22}	10^{-44}	ABS. F, G, X
5.^{0}	2	3	1.49×10^{-8}		8	6	3.04	10^{-14}	10^{-13}	10^{-28}	G
			2.23×10^{-16}		8	6	3.04	10^{-14}	10^{-13}	10^{-28}	G
6.	2	10	1.49×10^{-8}	5.0	1180	166	.365	10^{1}	10^{-6}	10^{-6}	X
			2.23×10^{-16}	5.0	154	22	.501	10^{2}	10^{4}	10^{1}	X
7.^{0}	3	3	1.49×10^{-8}		13	10	1.00	10^{-24}	10^{-23}	10^{-48}	ABS. F, G
			2.23×10^{-16}		13	10	1.00	10^{-24}	10^{-23}	10^{-48}	ABS. F, G
8.	3	15	1.49×10^{-8}		7	6	2.60	10^{-1}	10^{-11}	10^{-8}	G
			2.23×10^{-16}		7	6	2.60	10^{-1}	10^{-11}	10^{-8}	G
9.	3	15	1.49×10^{-8}		3	2	1.08	10^{-4}	10^{-12}	10^{-14}	G
			2.23×10^{-16}		3	2	1.08	10^{-4}	10^{-12}	10^{-14}	G
10.	3	16	1.49×10^{-8}		87	21	10^{4}	10^{1}	10^{0}	10^{1}	X
			2.23×10^{-16}		30	10	10^{4}	10^{1}	10^{-3}	10^{-7}	X
11.^{0}	3	10	1.49×10^{-8}		(3000)	(1232)	252.	10^{-1}	10^{3}	10^{-2}	F LIM.
			2.23×10^{-16}		(3000)	(998)	295.	10^{-1}	10^{-1}	10^{-2}	F LIM.
12.^{0}	3	10	1.49×10^{-8}		7	6	10.1	10^{-16}	10^{-16}	10^{-33}	ABS. F, G
			2.23×10^{-16}		7	6	10.1	10^{-16}	10^{-16}	10^{-33}	ABS. F, G
13.^{0}	4	4	1.49×10^{-8}		16	15	10^{-5}	10^{-8}	10^{-11}	10^{-16}	G
			2.23×10^{-16}		16	15	10^{-5}	10^{-8}	10^{-11}	10^{-16}	G
14.^{0}	4	6	1.49×10^{-8}		96	40	2.00	10^{-14}	10^{-13}	10^{-29}	ABS. F, G
			2.23×10^{-16}		96	40	2.00	10^{-14}	10^{-13}	10^{-29}	ABS. F, G
15.	4	11	1.49×10^{-8}		26	12	.328	10^{-2}	10^{-11}	10^{-9}	G
			2.23×10^{-16}		26	12	.328	10^{-2}	10^{-11}	10^{-9}	G
16.	4	20	1.49×10^{-8}		3484	1692	17.6	10^{2}	10^{-8}	10^{-8}	X
			2.23×10^{-16}		3484	1692	17.6	10^{2}	10^{-8}	10^{-8}	X
17.	5	33	1.49×10^{-8}		13	9	2.46	10^{-2}	10^{-11}	10^{-11}	G
			2.23×10^{-16}		13	9	2.46	10^{-2}	10^{-11}	10^{-11}	G
18.^{0}	6	13	1.49×10^{-8}	10.0	(6005)	(770)	2.97	10^{0}	10^{0}	10^{0}	F LIM.
			2.23×10^{-16}		(6006)	(750)	56.6	10^{0}	10^{0}	10^{0}	F LIM.
19.	11	65	1.49×10^{-8}		24	16	9.38	10^{-1}	10^{-11}	10^{-8}	G
			2.23×10^{-16}		24	16	9.38	10^{-1}	10^{-11}	10^{-8}	G

Numerical Results for the Gauss-Newton Methods

	n	m	rank tol.	step fac.	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
20a.	6	31	1.49×10^{-8}		12	11	2.44	10^{-2}	10^{-11}	10^{-10}	G
			2.23×10^{-16}		12	11	2.44	10^{-2}	10^{-11}	10^{-10}	G
20b.	9	31	1.49×10^{-8}		6	5	6.06	10^{-3}	10^{-11}	10^{-13}	G
			2.23×10^{-16}		6	5	6.06	10^{-3}	10^{-11}	10^{-13}	G
20c.	12	31	1.49×10^{-8}		6	5	16.6	10^{-5}	10^{-13}	10^{-16}	G
			2.23×10^{-16}		6	5	16.6	10^{-5}	10^{-13}	10^{-16}	G
20d.	20	31	1.49×10^{-8}		6	5	1.07	10^{-8}	10^{-13}	10^{-15}	G
			2.23×10^{-16}		6	5	247.	10^{-10}	10^{-12}	10^{-26}	G
21a.^{0}	10	10	1.49×10^{-8}		34	13	3.16	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
			2.23×10^{-16}		34	13	3.16	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
21b.^{0}	20	20	1.49×10^{-8}		34	13	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
			2.23×10^{-16}		34	13	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
22a.^{0}	12	12	1.49×10^{-8}		16	15	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
			2.23×10^{-16}		16	15	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
22b.^{0}	20	20	1.49×10^{-8}		16	15	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
			2.23×10^{-16}		16	15	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
23a.	4	5	1.49×10^{-8}		86	43	.500	10^{-3}	10^{-14}	10^{-10}	G
			2.23×10^{-16}		86	43	.500	10^{-3}	10^{-14}	10^{-10}	G
23b.	10	11	1.49×10^{-8}		99	36	.500	10^{-2}	10^{-12}	10^{-11}	G
			2.23×10^{-16}		99	36	.500	10^{-2}	10^{-12}	10^{-11}	G
24a.	4	8	1.49×10^{-8}		781	204	.759	10^{-3}	10^{-12}	10^{-11}	G
			2.23×10^{-16}		781	204	.759	10^{-3}	10^{-12}	10^{-11}	G
24b.	10	20	1.49×10^{-8}		(10002)	(1636)	.594	10^{-2}	10^{-1}	10^{-4}	F LIM.
			2.23×10^{-16}		(10002)	(1636)	.594	10^{-2}	10^{-1}	10^{-4}	F LIM.
25a.^{0}	10	12	1.49×10^{-8}		11	10	3.16	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
			2.23×10^{-16}		11	10	3.16	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
25b.^{0}	20	22	1.49×10^{-8}		13	12	4.47	10^{-15}	10^{-13}	10^{-30}	ABS. F, G
			2.23×10^{-16}		13	12	4.47	10^{-15}	10^{-13}	10^{-30}	ABS. F, G
26a.^{0}	10	10	1.49×10^{-8}		16	8	.306	10^{-11}	10^{-11}	10^{-22}	G
			2.23×10^{-16}		16	8	.306	10^{-11}	10^{-11}	10^{-22}	G
26b.^{0}	20	20	1.49×10^{-8}		25	10	.222	10^{-11}	10^{-11}	10^{-22}	G
			2.23×10^{-16}		25	10	.222	10^{-11}	10^{-11}	10^{-22}	G
27a.^{0}	10	10	1.49×10^{-8}		21	7	3.18	10^{-15}	10^{-14}	10^{-29}	ABS. F, G
			2.23×10^{-16}		21	7	3.18	10^{-15}	10^{-14}	10^{-29}	ABS. F, G
27b.^{0}	20	20	1.49×10^{-8}	10.0	22	9	4.47	10^{-12}	10^{-11}	10^{-23}	G
			2.23×10^{-16}	10.0	22	9	4.47	10^{-12}	10^{-12}	10^{-23}	G
28a.^{0}	10	10	1.49×10^{-8}		4	3	.412	10^{-15}	10^{-16}	10^{-31}	ABS. F, G
			2.23×10^{-16}		4	3	.412	10^{-15}	10^{-16}	10^{-31}	ABS. F, G
28b.^{0}	20	20	1.49×10^{-8}		4	3	.571	10^{-16}	10^{-16}	10^{-32}	ABS. F, G
			2.23×10^{-16}		4	3	.571	10^{-16}	10^{-16}	10^{-32}	ABS. F, G

Numerical Results for the Gauss-Newton Methods

	n	m	rank tol.	step fac.	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
29a.^{0}	10	10	1.49×10^{-8}		4	3	.412	10^{-14}	10^{-14}	10^{-29}	ABS. F, G
			2.23×10^{-16}		4	3	.412	10^{-14}	10^{-14}	10^{-29}	ABS. F, G
29b.^{0}	20	20	1.49×10^{-8}		4	3	.571	10^{-14}	10^{-14}	10^{-28}	G
			2.23×10^{-16}		4	3	.571	10^{-14}	10^{-14}	10^{-28}	G
30a.^{0}	10	10	1.49×10^{-8}		6	5	2.05	10^{-16}	10^{-15}	10^{-31}	ABS. F, G
			2.23×10^{-16}		6	5	2.05	10^{-16}	10^{-15}	10^{-31}	ABS. F, G
30b.^{0}	20	20	1.49×10^{-8}		6	5	3.04	10^{-15}	10^{-15}	10^{-31}	ABS. F, G
			2.23×10^{-16}		6	5	3.04	10^{-15}	10^{-15}	10^{-31}	ABS. F, G
31a.^{0}	10	10	1.49×10^{-8}		7	6	1.80	10^{-15}	10^{-15}	10^{-31}	ABS. F, G
			2.23×10^{-16}		7	6	1.80	10^{-15}	10^{-15}	10^{-31}	ABS. F, G
31b.^{0}	20	20	1.49×10^{-8}		7	6	2.66	10^{-15}	10^{-15}	10^{-31}	ABS. F, G
			2.23×10^{-16}		7	6	2.66	10^{-15}	10^{-15}	10^{-31}	ABS. F, G
32.^{L}	10	20	1.49×10^{-8}		2	1	3.16	10^{0}	10^{-14}	0.00	G
			2.23×10^{-16}		2	1	3.16	10^{0}	10^{-14}	0.00	G
33.^{L}	10	20	1.49×10^{-8}		3	2	5.40	10^{0}	10^{-10}	10^{-6}	G, X
			2.23×10^{-16}		8	8	10^{13}	10^{0}	10^{2}	10^{-3}	$g^Tp \geq 0$
34.^{L}	10	20	1.49×10^{-8}		3	2	4.90	10^{0}	10^{-11}	10^{-6}	G, X
			2.23×10^{-16}		3	2	4.90	10^{0}	10^{-11}	10^{-6}	G, X
35a.	8	8	1.49×10^{-8}		222	60	1.65	10^{-1}	10^{-11}	10^{-9}	G
			2.23×10^{-16}		110	17	1.63	10^{-1}	10^{-1}	10^{-2}	X
35b.^{0}	9	9	1.49×10^{-8}		107	21	1.73	10^{-16}	10^{-15}	10^{-31}	ABS. F, G
			2.23×10^{-16}		257	37	1.70	10^{-1}	10^{0}	10^{-2}	X
35c.	10	10	1.49×10^{-8}		(10003)	1261	1.79	10^{-1}	10^{-1}	10^{-2}	F LIM.
			2.23×10^{-16}		203	28	1.79	10^{-1}	10^{0}	10^{-2}	X
36a.^{0}	4	4	1.49×10^{-8}		(4001)	(677)	17.2	10^{-6}	10^{-4}	10^{-11}	F LIM.
			2.23×10^{-16}		95	39	50.0	10^{-16}	10^{-15}	10^{-33}	ABS. F, G
36b.^{0}	9	9	1.49×10^{-8}		1840	273	18.8	10^{-6}	10^{-12}	10^{-12}	G
			2.23×10^{-16}		92	38	50.0	10^{-16}	10^{-15}	10^{-33}	ABS. F, G
36c.^{0}	9	9	1.49×10^{-8}		20	19	1.73	10^{-11}	10^{-11}	10^{-22}	G
			2.23×10^{-16}		20	19	1.73	10^{-11}	10^{-11}	10^{-22}	G
36d.^{0}	9	9	1.49×10^{-8}		1793	268	19.0	10^{-6}	10^{-11}	10^{-12}	G
			2.23×10^{-16}		(9002)	(1158)	343.	10^{-6}	10^{-5}	10^{-13}	F LIM.
37.	2	16	1.49×10^{-8}		37	35	8.85	10^{1}	10^{-8}	10^{-6}	X
			2.23×10^{-16}		37	35	8.85	10^{1}	10^{-8}	10^{-6}	X
38.	3	16	1.49×10^{-8}		31	24	26.1	10^{1}	10^{-10}	10^{-6}	G
			2.23×10^{-16}		31	24	26.1	10^{1}	10^{-10}	10^{-6}	G

Numerical Results for the Gauss-Newton Methods

	n	m	rank tol.	step fac.	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
39a.	2	3	1.49×10^{-8}		8	7	10^{-6}	10^{-1}	10^{-11}	10^{-7}	G
			2.23×10^{-16}		8	7	10^{-6}	10^{-1}	10^{-11}	10^{-7}	G
39b.	2	3	1.49×10^{-8}		10	7	10^{-7}	10^{-1}	10^{-11}	10^{-7}	G
			2.23×10^{-16}		10	7	10^{-7}	10^{-1}	10^{-11}	10^{-7}	G
39c.	2	3	1.49×10^{-8}		23	14	10^{-7}	10^{-1}	10^{-12}	10^{-7}	G
			2.23×10^{-16}		23	14	10^{-7}	10^{-1}	10^{-12}	10^{-7}	G
39d.	2	3	1.49×10^{-8}		699	343	10^{-7}	10^{-1}	10^{-11}	10^{-7}	G
			2.23×10^{-16}		699	343	10^{-7}	10^{-1}	10^{-11}	10^{-7}	G
39e.	2	3	1.49×10^{-8}		1962	951	10^{-7}	10^{-1}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		1962	951	10^{-7}	10^{-1}	10^{-10}	10^{-7}	G
39f.	2	3	1.49×10^{-8}		(2001)	(750)	10^{-9}	10^{-1}	10^{-7}	10^{-7}	F LIM.
			2.23×10^{-16}		(2001)	(750)	10^{-9}	10^{-1}	10^{-7}	10^{-7}	F LIM.
39g.	2	3	1.49×10^{-8}		(2000)	(670)	10^{-9}	10^{-1}	10^{-6}	10^{-7}	F LIM.
			2.23×10^{-16}		(2000)	(670)	10^{-9}	10^{-1}	10^{-6}	10^{-7}	F LIM.
40a.	3	4	1.49×10^{-8}		13	12	10^{-6}	10^{0}	10^{-11}	10^{-7}	G
			2.23×10^{-16}		13	12	10^{-6}	10^{0}	10^{-11}	10^{-7}	G
40b.	3	4	1.49×10^{-8}		16	10	10^{-6}	10^{0}	10^{-12}	10^{-7}	G
			2.23×10^{-16}		16	10	10^{-6}	10^{0}	10^{-12}	10^{-7}	G
40c.	3	4	1.49×10^{-8}		381	188	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		381	188	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
40d.	3	4	1.49×10^{-8}		2695	1302	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		2695	1302	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
40e.	3	4	1.49×10^{-8}		(3001)	(983)	10^{-7}	10^{0}	10^{-6}	10^{-7}	G
			2.23×10^{-16}		(3001)	(983)	10^{-7}	10^{0}	10^{-6}	10^{-7}	G
40f.	3	4	1.49×10^{-8}		(3003)	(505)	10^{-1}	10^{0}	10^{0}	10^{-2}	F LIM.
			2.23×10^{-16}		(3003)	(505)	10^{-1}	10^{0}	10^{0}	10^{-2}	F LIM.
40g.	3	4	1.49×10^{-8}		(3002)	(514)	10^{-1}	10^{0}	10^{2}	10^{0}	F LIM.
			2.23×10^{-16}		(3002)	(514)	10^{-1}	10^{0}	10^{2}	10^{0}	F LIM.
41a.	5	10	1.49×10^{-8}		5	4	10^{-6}	10^{0}	10^{-13}	10^{-7}	G
			2.23×10^{-16}		5	4	10^{-6}	10^{0}	10^{-13}	10^{-7}	G
41b.	5	10	1.49×10^{-8}		6	5	10^{-6}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		6	5	10^{-6}	10^{0}	10^{-10}	10^{-7}	G
41c.	5	10	1.49×10^{-8}		12	11	10^{-6}	10^{0}	10^{-11}	10^{-7}	G
			2.23×10^{-16}		12	11	10^{-6}	10^{0}	10^{-11}	10^{-7}	G
41d.	5	10	1.49×10^{-8}		31	18	10^{-6}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		31	18	10^{-6}	10^{0}	10^{-10}	10^{-7}	G
41e.	5	10	1.49×10^{-8}		154	77	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		154	77	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
41f.	5	10	1.49×10^{-8}		812	368	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		812	368	10^{-7}	10^{0}	10^{-10}	10^{-7}	G
41g.	5	10	1.49×10^{-8}		2137	815	10^{-8}	10^{0}	10^{-10}	10^{-7}	G
			2.23×10^{-16}		2137	815	10^{-8}	10^{0}	10^{-10}	10^{-7}	G

Numerical Results for the Gauss-Newton Methods

	n	m	rank tol.	step fac.	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
42a.^{0}	4	24	1.49×10^{-8}	0.1	67	38	60.8	10^{-13}	10^{-10}	10^{-25}	x
			2.23×10^{-16}	0.1	67	38	60.8	10^{-13}	10^{-10}	10^{-25}	x
42b.^{0}	4	24	1.49×10^{-8}	0.1	611	316	61.9	10^{-13}	10^{-10}	10^{-25}	x
			2.23×10^{-16}	0.1	(4002)	(1561)	10^{5}	10^{2}	10^{11}	10^{5}	F LIM.
42c.^{0}	4	24	1.49×10^{-8}	0.1	33	26	60.3	10^{-13}	10^{-11}	10^{-27}	G, X
			2.23×10^{-16}	0.1	33	26	60.3	10^{-13}	10^{-11}	10^{-27}	G, X
42d.^{0}	4	24	1.49×10^{-8}	0.1	27	23	60.3	10^{-14}	10^{-11}	10^{-28}	G, X
			2.23×10^{-16}	0.1	27	23	60.3	10^{-14}	10^{-11}	10^{-28}	G, X
43a.^{0}	5	16	1.49×10^{-8}	1.0	22	14	54.0	10^{-14}	10^{-11}	10^{-27}	G
			2.23×10^{-16}	1.0	22	14	54.0	10^{-14}	10^{-11}	10^{-27}	G
43b.^{0}	5	16	1.49×10^{-8}	1.0	1167	392	62.1	10^{-1}	10^{-11}	10^{-2}	G
			2.23×10^{-16}	1.0	1167	392	62.1	10^{-1}	10^{-11}	10^{-2}	G
43c.^{0}	5	16	1.49×10^{-8}	1.0	23	14	54.0	10^{-14}	10^{-12}	10^{-28}	G
			2.23×10^{-16}	1.0	23	14	54.0	10^{-14}	10^{-12}	10^{-28}	G
43d.^{0}	5	16	1.49×10^{-8}	1.0	19	9	54.0	10^{-14}	10^{-12}	10^{-27}	G
			2.23×10^{-16}	1.0	19	9	54.0	10^{-14}	10^{-12}	10^{-27}	G
43e.^{0}	5	16	1.49×10^{-8}	1.0	37	19	54.0	10^{-14}	10^{-11}	10^{-27}	G
			2.23×10^{-16}	1.0	37	19	54.0	10^{-14}	10^{-11}	10^{-27}	G
43f.^{0}	5	16	1.49×10^{-8}	2.0	20	11	54.0	10^{-14}	10^{-12}	10^{-27}	G
			2.23×10^{-16}	2.0	20	11	54.0	10^{-14}	10^{-12}	10^{-27}	G
44a.^{0}	6	6	1.49×10^{-8}		125	29	4.06	10^{-14}	10^{-12}	10^{-27}	G
			2.23×10^{-16}		125	29	4.06	10^{-14}	10^{-12}	10^{-27}	G
44b.^{0}	6	6	1.49×10^{-8}		5	4	3.52	10^{-15}	10^{-13}	10^{-29}	ABS. F, G
			2.23×10^{-16}		5	4	3.52	10^{-15}	10^{-13}	10^{-29}	ABS. F, G
44c.^{0}	6	6	1.49×10^{-8}		52	18	20.6	10^{-14}	10^{-11}	10^{-29}	ABS. F
			2.23×10^{-16}		52	18	20.6	10^{-14}	10^{-11}	10^{-29}	ABS. F
44d.^{0}	6	6	1.49×10^{-8}		36	15	15.3	10^{-14}	10^{-11}	10^{-29}	ABS. F, G
			2.23×10^{-16}		36	15	15.3	10^{-14}	10^{-11}	10^{-29}	ABS. F, G
44e.^{0}	6	6	1.49×10^{-8}		70	23	9.27	10^{-15}	10^{-12}	10^{-29}	ABS. F, G
			2.23×10^{-16}		70	23	9.27	10^{-15}	10^{-12}	10^{-29}	ABS. F, G
45a.^{0}	8	8	1.49×10^{-8}		125	29	4.06	10^{-14}	10^{-12}	10^{-27}	G
			2.23×10^{-16}		125	29	4.06	10^{-14}	10^{-12}	10^{-27}	G
45b.^{0}	8	8	1.49×10^{-8}		5	4	3.56	10^{-15}	10^{-13}	10^{-29}	ABS. F, G
			2.23×10^{-16}		5	4	3.56	10^{-15}	10^{-13}	10^{-29}	ABS. F, G
45c.^{0}	8	8	1.49×10^{-8}		52	18	20.6	10^{-14}	10^{-11}	10^{-29}	ABS. F
			2.23×10^{-16}		52	18	20.6	10^{-14}	10^{-11}	10^{-29}	ABS. F
45d.^{0}	8	8	1.49×10^{-8}		36	15	15.3	10^{-15}	10^{-11}	10^{-29}	ABS. F, G
			2.23×10^{-16}		36	15	15.3	10^{-15}	10^{-11}	10^{-29}	ABS. F, G
45e.^{0}	8	8	1.49×10^{-8}		70	23	9.31	10^{-14}	10^{-11}	10^{-28}	G
			2.23×10^{-16}		70	23	9.31	10^{-14}	10^{-11}	10^{-28}	G

4.8 Bibliography

Bard, Y., "Comparison of Gradient Methods for the Solution of Parameter Estimation Problems", SIAM *Journal on Numerical Analysis* Vol. 7 No. 1 (March 1970) 157–186.

Brent, R. P., *Algorithms for Minimization Without Derivatives*, Prentice Hall (1973).

Dennis, J. E., "III.2 Nonlinear Least Squares and Equations", in *The State of the Art in Numerical Analysis*, D. Jacobs (ed.), Academic Press (1977) 269–312.

Dennis, J. E., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall (1983).

Deuflhard, P., and V. Apostolescu, "A Study of the Gauss-Newton Algorithm for the Solution of Nonlinear Least Squares Problems", in *Special Topics in Applied Mathematics*, North-Holland (1980) 129–150.

Dongarra, J. J., J. R. Bunch, C. B. Moler, and G. W. Stewart, LINPACK *Users' Guide*, SIAM (1979).

Fletcher, R., "Function Minimization Without Evaluating Derivatives — A Review", *Computer Journal* 8 (1963) 163–168.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Two Steplength Algorithms for Numerical Optimization", Technical Report SOL 79-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University (1979).

Gill, P. E., S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright, "User's Guide for LSSOL (Version 1.0): A Fortran Package for Constrained Linear Least-Squares and Convex Quadratic Programming", Technical Report SOL 86-1, Systems Optimization Laboratory, Department of Operations Research, Stanford University (January 1986a).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Users Guide for NPSOL (Version 4.0) : A Fortran Package for Nonlinear Programming", Technical Report SOL 86-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University (January 1986b).

Golub, G. H., and C. L. Van Loan, *Matrix Computations*, Johns-Hopkins (1983).

Haussler, W. M., "A Kantorovich-Type Convergence Analysis for the Gauss-Newton Method", *Numerische Mathematik*, Vol. 48 No. 1 (1986) 119–125.

McKeown, J. J., "Specialized versus General-Purpose Algorithms for Minimising Functions that are Sums of Squared Terms", *Mathematical Programming* 9 (1975a) 57–68.

McKeown, J. J., "On Algorithms for Sums of Squares Problems", Paper 14 in *Towards Global Optimization*, L. C. W. Dixon and G. Szegö (eds.), North Holland (1975b) 229–257.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization Software", ACM *Transactions on Mathematical Software*, Vol. 7 No. 1 (March 1981) 17–41.

Nocedal, J., and M. L. Overton, "Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization", SIAM *Journal on Numerical Analysis*, Vol. 22 No. 5 (October 1985).

Osborne, M. R., "Some Aspects of Non-linear Least Squares Calculations", in *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma (ed.), Academic Press (1972) 171–189.

Powell, M. J. D., "A Hybrid Method for Nonlinear Equations", in *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz (ed.), Gordon and Breach (1970a) 87–114.

Ramsin, H., and P-Å. Wedin, "A Comparison of some Algorithms for the Nonlinear Least Squares Problem", *BIT* 17 No. 1 (1977) 72–90.

Schaback, R., "Convergence Analysis of the General Gauss-Newton Algorithm", *Numerische Mathematik*, Vol. 46 No. 2 (1985) 281–309.

5. Survey of Algorithms and Software

5.1 Overview

The purpose of this chapter is to survey research in algorithms for small, dense nonlinear least-squares problems, with emphasis on those for which software is readily available and has been extensively tested. The three principal approaches to solving general nonlinear least-squares problems are the subject of the next three sections — Levenberg-Marquardt methods, one of which is implemented in the software package MINPACK [Moré (1978), Moré, Garbow, and Hillstrom (1980)]; corrected Gauss-Newton methods [Gill and Murray (1978)]; which form the basis for the NAG Library nonlinear least-squares software; and methods that form quasi-Newton approximations to the term $B = \sum_{i=1}^{m} \phi_i \nabla^2 \phi_i$ in the nonlinear least-squares Hessian, a strategy that is adaptively combined with a Gauss-Newton method and a Levenberg-Marquardt method in the computer algorithm NL2SOL [Dennis, Gay, and Welsch (1981a, 1981b)]. Each of these methods modifies the Gauss-Newton search direction in a different way. The Levenberg-Marquardt methods alter the search direction in the range of J, by replacing $J^T J$ with $J^T J + \lambda D^T D$, D diagonal, in the quadratic model function. The corrected Gauss-Newton methods compute a Gauss-Newton search direction in a subspace of the range of J^T, and obtain a component in the corresponding null space by a projected Newton method. Special quasi-Newton methods for nonlinear least squares use a Hessian of the form $J^T J + \tilde{B}$ in the quadratic model, so that the search direction differs from the Gauss-Newton direction in $\mathcal{R}(J^T)$, and also has a component in $\mathcal{N}(J)$ when J is rank-deficient. Some other nonlinear least-squares algorithms are discussed briefly in Section 5.5. Numerical results are presented in Section 5.6 for the test problems (see the Appendix). Finally, a summary of all of the numerical results (including those for unconstrained optimization methods from Chapter 2 and for Gauss-Newton methods from Chapter 4) is given in Section 5.7.

5.2 Levenberg-Marquardt Methods

In Levenberg-Marquardt methods, the Gauss-Newton quadratic model (4.1.2) is minimized subject to a trust-region constraint (see Sections 2.4.2 and 2.5.1). The step p between successive iterates solves

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T J^T J p \qquad (5.2.1)$$
$$\text{subject to} \quad \|Dp\|_2 \leq \delta,$$

for some $\delta > 0$ and some diagonal scaling matrix D with positive diagonal entries. Equivalently, p solves

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T (J^T J + \lambda D^T D) p, \qquad (5.2.2)$$

for some $\lambda \geq 0$. Since the matrix $J^T J + \lambda D^T D$ is positive semidefinite, solutions p_λ to (5.2.2) satisfy the equations

$$(J^T J + \lambda D^T D) p = -\bar{g} = -J^T f, \qquad (5.2.3)$$

which are the normal equations for the linear least-squares problem

$$\min_{p \in \Re^n} \left\| \begin{pmatrix} J \\ \sqrt{\lambda} D \end{pmatrix} p - \begin{pmatrix} f \\ 0 \end{pmatrix} \right\|_2^2. \qquad (5.2.4)$$

Hence a regularization method (see Section 3.4.3.4) is being used to obtain the step to the next iterate.

The paper by Levenberg [1944] is the earliest known reference to methods of this type. Based on the observation that the unit Gauss-Newton step p_{GN} often fails to reduce the sum of squares when $\|p_{GN}\|$ is not especially small, he suggests limiting the size of the search direction by solving a "damped" least-squares subproblem,

$$\min_{p \in \Re^n} \omega(\bar{g}^T p + \frac{1}{2} p^T J^T J p) + \|Dp\|_2^2, \qquad (5.2.5)$$

in which a weighted sum of squares of linearized residuals and components of the search direction is minimized. He proves the existence of a value of ω for which

$$\|f(x + p_\omega)\|_2 < \|f(x)\|_2,$$

where p_ω solves (5.2.5), thus ensuring a reduction in the sum of squares for a suitable value of ω. A major drawback is that no automatic procedure is given for obtaining ω. Levenberg

suggests computing the value of $\|f(x+p_\omega)\|_2$ for several trial values of ω, locating an approximate minimum graphically, and then repeating this procedure with the improved estimates until a satisfactory value of ω is obtained, but precise criteria for accepting a trial value are not given. Two alternatives are proposed for the diagonal scaling matrix D in (5.2.5) :- $D = I$, because it minimizes the directional derivative $\bar{g}^T p_\omega$ for $\omega = 0$, and the square root of the diagonal of $J^T J$, based on empirical observations. The claim is that the new method solves a wider class of problems than existing methods, and that it does so with relative efficiency.

Somewhat later, a similar method was (apparently independently) proposed. Morrison [1960] considers a quadratic model

$$\bar{g}^T p + \frac{1}{2} p^T H p, \tag{5.2.6}$$

in which either $H = J^T J$ or $H = \nabla^2 (f^T f)$ (in the later case, it is implicitly assumed that $\nabla^2 (f^T f)$ is positive semidefinite). He advocates minimizing (5.2.6) over a neighborhood of the current point as does Levenberg, because (5.2.6) may not be a good approximation to $\|f(x+p)\|_2^2 - \|f(x)\|_2^2$ if the minimizer p^* is large in magnitude, and consequently the sum of squares may not be reduced at $x + p^*$. (In Hartley [1961], a linesearch is used with the Gauss-Newton direction for the same reason.) Morrison proves that the solution p_λ to

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T (H + \lambda D) p$$

for $\lambda > 0$ is the constrained minimum of (5.2.6) on the sphere of radius $\|D p_\lambda\|_2$, and that $\|p_\lambda\|_2 \to 0$ as $\lambda \to \infty$. In Morrison's method, the step bound δ is the independent parameter, rather than λ. No specifications are given for either δ or D, although it is implied that they can be chosen heuristically for a given problem. Instead of minimizing (5.2.6) subject to $\|Dp\|_2 \leq \delta$, constraints of the form $|d_i x_i| < \delta$ are imposed, and the resulting subproblem is then solved using the eigenvalue decomposition of H. Although the theory and methods apply for any positive semi-definite H in (5.2.6), no generalization to unconstrained minimization is mentioned.

Marquardt [1963] extended Morrison's work, showing that the vector p_λ that solves (5.2.3) becomes parallel to the steepest-descent direction as $\lambda \to \infty$, so that p_λ interpolates between the Gauss-Newton search direction, p_0, and the steepest-descent direction,

p_∞. He points out that the method determines both the direction from the current iterate to the next one, and the distance between the iterates along that direction, and that increasing λ decreases the step length, while shifting the direction away from orthogonality to the gradient of the sum of squares. Marquardt's strategy controls λ automatically by multiplying or dividing the current value by a constant factor ν greater than 1. He maintains that the minimum of the Gauss-Newton model should be taken over the largest possible neighborhood, that is, that λ should be chosen as small as possible, so as to achieve faster convergence by biasing the search direction toward the Gauss-Newton direction when Gauss-Newton methods would work well. Thus, at the kth iteration, $\lambda_k = \lambda_{k-1}/\nu$ is tried first, and then increased if necessary by multiples of ν until a reduction in the sum of squares is obtained. A shortcoming of this scheme is that λ is always positive, so that the constraint in (5.2.1) is active in every subproblem, and consequently a full Gauss-Newton step can never be taken. Also, no efficient method is given for solving (5.2.3) for different values of λ. Motivated by statistical considerations, Marquardt uses the diagonal of J^TJ for the scaling matrix D (one of the alternatives proposed by Levenberg), and mentions that this scaling has been widely used as a technique for computing solutions to ill-conditioned linear least-squares problems.

Since the appearance of Marquardt's paper, much research has been directed toward improvements within the framework presented there. Bard [1970] takes the eigenvalue decompostion of J^TJ at each iteration, so that (5.2.3) can be easily solved for several values of λ, and so that it will be known whether or not J^TJ is singular. Bartels, Golub, and Saunders [1970] show how to use the SVD of J instead of the eigenvalue decomposition for the same purpose. They also give an algorithm for computing λ given δ that involves determining some eigenvalues of a diagonal matrix after a symmetric rank-one update. Meyer [1970] discusses the use of a linesearch with Marquardt's method (see also Osborne [1972]). Shanno [1970] selects λ so that p_λ is a direction of decrease for $\|f(x)\|_2^2$. The value $\lambda = 0$ is tried first, and then increases are made by multiplying a threshold value by a factor greater than one until $\psi'(\lambda) < 0$, where $\psi(\lambda) = \|f(x + p_\lambda)\|_2$. In addition, a linesearch is also used when $cos(p_\lambda, \bar{g})$ is above a threshold value, that is, when p_λ is judged to be nearly in the direction of $-\bar{g}$. Shanno's method is meant for general unconstrained or linearly-constrained minimization, as well as for nonlinear least squares.

Several methods have attempted to approximate Levenberg-Marquardt directions by a vector that is the sum of a component in the steepest descent direction, and a component in the Gauss-Newton direction p_{GN}. Jones [1970] combines searches along a spiral arc connecting p_{GN} and the origin with parabolic interpolation in order to obtain a decrease in the sum of squares. If a reduction is not achieved after trying several arcs, then the steepest descent direction is searched. The method of Powell [1970a] for nonlinear equations and [1970b] for unconstrained optimization searches along a piecewise linear curve. The algorithm for unconstrained optimization requires some agreement between the reduction predicted by the quadratic model and the actual reduction in the sum of squares, before the step is accepted. Global convergence results that include use of the quadratic model (4.1.2) for nonlinear least squares are given in Powell [1975] (see also Moré [1983]). Steen and Byrne [1973] approximate a search along an arc that intersects $\bar{g}$ at a nonzero point. Their algorithm requires that $J^{\mathrm{T}}J$ be scaled so that its smallest eigenvalue is 2, which they accomplish by computing $(J^{\mathrm{T}}J)^{-1}$ and finding either $\|(J^{\mathrm{T}}J)^{-1}\|_1$ or $\|(J^{\mathrm{T}}J)^{-1}\|_\infty$. A diagonal of unspecified small magnitude is added to $J^{\mathrm{T}}J$ in the event of singularity. A difficulty with any algorithm based on this type of approach is that it is not clear how to specify the approximation when the Gauss-Newton direction is not numerically well defined.

Fletcher [1971] implements a modified version of Marquardt's algorithm, in which adjustments in the parameter λ are made on the basis of a comparison of the actual reduction in the sum of squares

$$\frac{1}{2}\left(\|f(x+p_\lambda)\|_2^2 - \|f(x)\|_2^2\right), \tag{5.2.7}$$

with the reduction predicted by the model

$$\bar{g}^{\mathrm{T}}p_\lambda + \frac{1}{2}p_\lambda^{\mathrm{T}}J^{\mathrm{T}}Jp_\lambda, \tag{5.2.8}$$

the optimum value of the objective in (5.2.1). The step p_λ is taken only when there is sufficient agreement between (5.2.7) and (5.2.8), instead of accepting p_λ whenever the trial step results a reduction in the sum of squares. Fletcher also introduces more complicated techniques for updating λ. The scheme for decreasing λ differs from that given by Marquardt in that division by a constant factor is used only until λ reaches a threshold value, λ_c, below which it is replaced by zero. This modification is motivated by a desire to allow the Gauss-Newton step ($\lambda = 0$) when Gauss-Newton methods would work well, since λ is always positive in Marquardt's method, and to allow the initial choice of $\lambda = 0$ rather than some

arbitrary positive value. Because numerical experiments show that multiplying by a fixed constant factor may be inefficient, Fletcher uses safeguarded quadratic interpolation to increase λ when (5.2.7) and (5.2.8) differ substantially. If the current value of λ is nonzero, then it is divided by a factor

$$\gamma = \begin{cases} 0.1, & \text{if } \alpha_{min} < 0.1; \\ \alpha_{min}, & \text{if } \alpha_{min} \in [0.1, 0.5]; \\ 0.5, & \text{if } \alpha_{min} > 0.5, \end{cases} \tag{5.2.9}$$

where α_{min} is the minimum of the quadtratic interpolant to the function $\phi(\alpha) = \|f(x + \alpha p)\|_2^2$ at $\phi(0)$, $\phi'(0)$, and $\phi(1)$. There is also a provision to increase $\lambda = 0$ to the threshold value λ_c under certain circumstances. The choice of λ_c appears to be a major difficulty.

Fletcher gives some theoretical justification for choosing λ_c to be the reciprocal of the smallest eigenvalue of $(J^{\mathrm{T}}J)^{-1}$. Since he chooses to solve (5.2.3) directly for each value of λ via the Cholesky factorization, rather than compute the eigenvalue decomposition of $J^{\mathrm{T}}J$ or the singular values of J, the minimum eigenvalue of $J^{\mathrm{T}}J$ is not available without further computation. He therefore updates the estimate of λ_c only when λ is increased from 0, calculating $(J^{\mathrm{T}}J)^{-1}$ from the Cholesky factorization of $J^{\mathrm{T}}J$, and then takes either $\lambda_c = 1/\left\|(J^{\mathrm{T}}J)^{-1}\right\|_\infty$, or $\lambda_c = 1/trace\left((J^{\mathrm{T}}J)^{-1}\right)$. A drawback is that λ_c is not defined when $J^{\mathrm{T}}J$ is singular, and it is not well defined when $J^{\mathrm{T}}J$ is ill-conditioned. Harwell subroutine VA07A is an implementation of Fletcher's method. It allows the user to select the scaling matrix D, which then remains fixed throughout the computation. The default takes the square root of the diagonal of $J^{\mathrm{T}}J$ at the starting value as the scaling matrix.

An efficient and stable method for solving (5.2.3) for several values of λ based on the linear least-squares formulation (5.2.4) is given by Osborne [1972]. The method is accomplished in two stages. First, the QR factorization of J is computed, to obtain

$$\begin{pmatrix} Q & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} J \\ \sqrt{\lambda} D \end{pmatrix} = \begin{pmatrix} R \\ \sqrt{\lambda} D \end{pmatrix}, \tag{5.2.10}$$

after which a series of elementary orthogonal transformations are applied to reduce the right-hand side of (5.2.10) to triangular form. Thus it is only necessary to repeat the second stage of this procedure when the value of λ is changed, provided the QR factorization of J is saved. In a later paper, Osborne [1976] discusses a variant of Marquardt's algorithm for which he proves global convergence to a stationary point of $f^{\mathrm{T}}f$ under the assumption that the sequence $\{\lambda_k\}$ remains bounded. In this method, he uses a simple scheme similar to

the one proposed by Marquardt to update λ, but controls adjustments in λ by comparing (5.2.7) and (5.2.8). His implementation takes D to be the square root of the diagonal of $J^{\mathrm{T}}J$, as in Marquardt's method.

The algorithm of Moré [1978] adjusts the step bound δ in (5.2.1) rather than λ, a strategy used in trust-region methods for unconstrained optimization (see Moré [1983] for a survey). Changes in δ depend on agreement between (5.2.7) and (5.2.8); increases are accomplished by taking $\delta_{k+1} = 2\|D_k p_k\|_2$, while δ is decreased by multiplying by the factor γ defined by (5.2.9). In order to obtain λ when the bound in (5.2.1) is active, the nonlinear equation

$$\Psi(\lambda) = \|Dp_\lambda\|_2 - \delta = \left\| \left(J^{\mathrm{T}}J + \lambda D^{\mathrm{T}}D\right)^{-1} \bar{g} \right\|_2 - \delta = 0 \qquad (5.2.11)$$

is approximately solved by truncating a safeguarded Newton method based on the work of Hebden [1973] (see also Reinsch [1971]). Moré reports that, on the average, (5.2.11) is solved fewer than two times per iteration. Also, he proves global convergence to a stationary point of $f^{\mathrm{T}}f$, without assuming boundedness for $\{\lambda_k\}$. Many computational details are given, including an efficient method for calculating the derivative of $\Psi(\lambda)$ in (5.2.11) that uses the QR factorization of J. A modification of the two-stage factorization described by Osborne that allows column pivoting is used to solve (5.2.3). Subroutine LMDER in MINPACK [Moré, Garbow, and Hillstrom (1980)] is an implementation of the method. Variables are scaled internally in LMDER according to the following scheme : the initial scaling matrix D_0 is the square root of the diagonal of $J^{\mathrm{T}}J$ evaluated at x_0, and the ith diagonal element of D_k is taken to be the maximum of the ith diagonal element of D_{k-1} and the square root of the ith diagonal element of $J^{\mathrm{T}}J$. Numerical results are presented that show that this scaling compares favorably with those used by Fletcher, and by Marquardt and Osborne. The user also has the option of providing an initial diagonal scaling matrix that is retained throughtout the computation.

5.3 Corrected Gauss-Newton Methods

Gill and Murray [1976] propose a linesearch algorithm that divides $\Re^n$ into complementary subspaces $\tilde{\mathcal{R}}$ and $\tilde{\mathcal{N}}$, where $\tilde{\mathcal{R}} \subseteq \mathcal{R}(J^{\mathrm{T}})$ and $\tilde{\mathcal{N}}$ is nearly orthogonal to $\mathcal{R}(J^{\mathrm{T}})$. The search direction is the sum of a projected Gauss-Newton direction in $\tilde{\mathcal{R}}$, and a projected Newton direction in $\tilde{\mathcal{N}}$. This strategy avoids a shortcoming of Gauss-Newton methods —

that components of the search direction that are nearly orthogonal to $\mathcal{R}(J^T)$ may not be well determined when J is ill-conditioned — because each component is computed from a reasonably well-conditioned subproblem. In the example of Section 4.3.1, the vector $x - x^*$ becomes almost entirely in $\mathcal{R}(J^T)$ in a Gauss-Newton method, yet the algorithm computes a search direction that is virtually orthogonal to $\mathcal{R}(J^T)$ due to ill conditioning in the Jacobian. Gill and Murray also show that both Gauss-Newton algorithms defined by (4.1.5) and Levenberg-Marquardt algorithms generate search directions that lie in $\mathcal{R}(J^T)$, while the Newton search direction generally will have a component in $\mathcal{N}(J)$, the orthogonal complement of $\mathcal{R}(J^T)$, whenever J has linearly dependent columns. For problems with small residuals, they point out that J^TJ is a reasonable approximation to the full Hessian in $\mathcal{R}(J^T)$, but not in $\mathcal{N}(J)$. Thus, in situations where $x - x^*$ is orthogonal to $\mathcal{R}(J^T)$, and J is well-conditioned but has linearly dependent columns (for example, when $m < n$), the Gauss-Newton and Levenberg-Marquardt directions have no component in the direction of $x - x^*$, while Newton's method and also the method of Gill and Murray would have components in both $\mathcal{R}(J^T)$ and $\mathcal{N}(J)$.

The basic idea of the method is as follows. Suppose

$$J = QTV^T \tag{5.3.1}$$

is an orthogonal factorization of J, in which T is triangular with diagonal elements in decreasing order of magnitude (either a QR factorization with column pivoting or the singular-value decomposition). Let

$$V = (Y \quad Z) \tag{5.3.2}$$

be a partition of V into the first $grade(J)$ columns and the remaining $n-grade(J)$ columns. The columns of Y form an orthonormal basis for $\tilde{\mathcal{R}}$, and those of Z form an orthonormal basis for $\tilde{\mathcal{N}}$. The Newton search direction for NLSQ is given by

$$(J^TJ + B)p = -J^Tf,$$

with

$$B = \sum_{i=1}^{m} \phi_i \nabla^2 \phi_i,$$

or, equivalently,

$$V^T(J^TJ + B)p = -V^TJ^Tf, \tag{5.3.3}$$

since V is nonsingular. Using (5.3.2), equation (5.3.3) can be split into two equations :

$$Y^{\mathrm{T}}(J^{\mathrm{T}}J + B)p = -Y^{\mathrm{T}}J^{\mathrm{T}}f, \tag{5.3.4}$$

and

$$Z^{\mathrm{T}}(J^{\mathrm{T}}J + B)p = -Z^{\mathrm{T}}J^{\mathrm{T}}f. \tag{5.3.5}$$

Substituting $p = Yp_Y + Zp_Z$ into (5.3.4) yields

$$Y^{\mathrm{T}}J^{\mathrm{T}}JYp_Y + Y^{\mathrm{T}}J^{\mathrm{T}}JZp_Z + Y^{\mathrm{T}}Bp = -Y^{\mathrm{T}}J^{\mathrm{T}}f.$$

Since $grade(J)$ is chosen to approximate $rank(J)$, $\|JZ\|$ is presumed to be zero, so that $Y^{\mathrm{T}}J^{\mathrm{T}}JZp_Z$ vanishes. Also, for zero residual problems, the term $Y^{\mathrm{T}}Bp$ would be small near a minimum relative to $Y^{\mathrm{T}}J^{\mathrm{T}}JYp_Y$, since $\|B\|$ approaches zero. Defining ϵ to be $\|x - x^*\|$, where x^* is a minimum at which the residuals are zero, and assuming $\|f\| = \mathcal{O}(\epsilon)$ we have

$$Y^{\mathrm{T}}J^{\mathrm{T}}JYp_Y = \mathcal{O}(\epsilon);\quad Y^{\mathrm{T}}Bp = \mathcal{O}(\epsilon^2);\quad Y^{\mathrm{T}}J^{\mathrm{T}}f = \mathcal{O}(\epsilon).$$

The range-space component of the search direction is therefore chosen to satisfy

$$Y^{\mathrm{T}}J^{\mathrm{T}}JYp_Y = -Y^{\mathrm{T}}J^{\mathrm{T}}f. \tag{5.3.6}$$

With $grade(J) = rank(J)$, the vector Yp_Y is the minimal l_2-norm least-squares solution to $Jp \approx -f$ (Chapter 3), and is therefore a Gauss-Newton direction (Chapter 4). For the null-space portion, since $JZ = 0$ is assumed, (5.3.6) reduces to

$$Z^{\mathrm{T}}Bp = 0,$$

which may be solved for Zp_Z given Yp_Y from (5.3.5) using

$$Z^{\mathrm{T}}BZp_Z = -Z^{\mathrm{T}}BYp_Y. \tag{5.3.7}$$

When exact second derivatives are not available, the use of finite difference approximations along the columns of Z is suggested.

A version of this algorithm called the *corrected Gauss-Newton method* [Gill and Murray (1978)] forms the basis for the nonlinear least-squares software currently in the NAG Library [1984]. It uses the singular-value decomposition of J, rather than a QR factorization. Rules based on the relative size of the singular values are given for choosing an integer $grade(J)$ to approximate $rank(J)$, and an attempt is made to group together singular

values that are similar in magnitude. The method is not as sensitive to $grade(J)$ as Gauss-Newton is to rank estimation, both because of the division of the computation of the search direction into separate components in $\tilde{\mathcal{R}}$ and $\tilde{\mathcal{N}}$, and because $grade(J)$ is varied adaptively based on a measure of the progress of the minimization. Moreover, the rate of convergence is potentially faster than Gauss-Newton or Levenberg-Marquardt methods on problems with nonzero residuals. The quantity $grade(J)$ is reduced when the sum of squares is not adequately decreasing, so that there is the potential of having $\tilde{\mathcal{N}} = \Re^n$ (with exact second derivatives, this implies taking full Newton steps) in the vicinity of a solution. The derivation below shows how the corrected Gauss-Newton method differs from the earlier version based on the QR factorization.

Because of (5.3.1), $J^{\mathrm{T}}J$ can be written as $VT^{\mathrm{T}}TV^{\mathrm{T}}$, so that (5.3.3) is equivalent to

$$T^{\mathrm{T}}TV^{\mathrm{T}}p + V^{\mathrm{T}}Bp = -T^{\mathrm{T}}Q^{\mathrm{T}}f. \tag{5.3.8}$$

Using $p = Yp_Y + Zp_Z$, along with

$$V^{\mathrm{T}}Y = \begin{pmatrix} I_k \\ 0 \end{pmatrix} \quad \text{and} \quad V^{\mathrm{T}}Z = \begin{pmatrix} 0 \\ I_{n-k} \end{pmatrix},$$

(5.3.8) becomes

$$T^{\mathrm{T}}T \begin{pmatrix} I_k \\ 0 \end{pmatrix} p_Y + T^{\mathrm{T}}T \begin{pmatrix} 0 \\ I_{n-k} \end{pmatrix} p_Z + V^{\mathrm{T}}Bp = -T^{\mathrm{T}}Q^{\mathrm{T}}f. \tag{5.3.9}$$

If we let

$$T = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}$$

be a partition of T, where T_{11} is the submatrix consisting of the first k rows and columns of T, then

$$T^{\mathrm{T}}T = \begin{pmatrix} (T_{11}^{\mathrm{T}}T_{11} + T_{21}^{\mathrm{T}}T_{21}) & (T_{11}^{\mathrm{T}}T_{12} + T_{21}^{\mathrm{T}}T_{22}) \\ (T_{12}^{\mathrm{T}}T_{11} + T_{22}^{\mathrm{T}}T_{21}) & (T_{12}^{\mathrm{T}}T_{12} + T_{22}^{\mathrm{T}}T_{22}) \end{pmatrix},$$

and (5.3.9) can be split into two equations :

$$(T_{11}^{\mathrm{T}}T_{11} + T_{21}^{\mathrm{T}}T_{21})p_Y + (T_{11}^{\mathrm{T}}T_{12} + T_{21}^{\mathrm{T}}T_{22})p_Z + Y^{\mathrm{T}}Bp = -\begin{pmatrix} T_{11}^{\mathrm{T}} & T_{12}^{\mathrm{T}} \end{pmatrix} Q^{\mathrm{T}}f, \tag{5.3.10}$$

and

$$(T_{12}^{\mathrm{T}}T_{11} + T_{22}^{\mathrm{T}}T_{21})p_Y + (T_{12}^{\mathrm{T}}T_{12} + T_{22}^{\mathrm{T}}T_{22})p_Z + Z^{\mathrm{T}}Bp = -\begin{pmatrix} T_{12} & T_{22}^{\mathrm{T}} \end{pmatrix} Q^{\mathrm{T}}f. \tag{5.3.11}$$

As in the earlier version, the term $Y^{\mathrm{T}}Bp$ is ignored in (5.3.10). Moreover, in the case that (5.3.1) is the singular value decomposition, both T_{12} and T_{21} vanish and the two equations can be further simplified to

$$S_1^2 p_Y = -\begin{pmatrix} S_1 & 0 \end{pmatrix} Q^{\mathrm{T}} f, \tag{5.3.12}$$

and

$$S_2^2 p_Z + Z^{\mathrm{T}} B p = -\begin{pmatrix} 0 & S_2 \end{pmatrix} Q^{\mathrm{T}} f, \tag{5.3.13}$$

where

$$S_1 \equiv T_{11} \quad \text{and} \quad S_2 \equiv T_{22}.$$

Note that S_1 and S_2 are diagonal matrices, and that the p_Y term in the second equation could not be ignored if (5.3.1) were a triangular factorization of J, because then $(T_{12}^{\mathrm{T}} T_{11} + T_{22}^{\mathrm{T}} T_{21})$ could not be assumed negligible relative to $(T_{12}^{\mathrm{T}} T_{12} + T_{22}^{\mathrm{T}} T_{22})$. The equations which are ultimately solved are

$$S_1 p_Y = -\begin{pmatrix} I_k & 0 \end{pmatrix} Q^{\mathrm{T}} f, \tag{5.3.14}$$

and

$$(S_2^2 + Z^{\mathrm{T}} B Z) p_Z = -\begin{pmatrix} 0 & S_2 \end{pmatrix} Q^{\mathrm{T}} f - Z^{\mathrm{T}} B p_Y. \tag{5.3.15}$$

The matrix $S_2^2 + Z^{\mathrm{T}} B Z$ is replaced by a modified Cholesky factorization if it is computationally singular or indefinite. The range-space component is a Gauss-Newton search direction, while, in the positive-definite case, the null-space component is a projected Newton direction. When no modification is necessary, the subproblem being solved is

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} (J^{\mathrm{T}} J + B) p \tag{5.3.16}$$

$$\text{subject to} \quad Jp \simeq -f,$$

where '$\simeq$' is taken in a least-squares sense if the rows of J are linearly dependent, as in the case when $m > n$, and otherwise as equality. Subproblem (5.3.16) is an equality constrained quadratic program. When $rank(J) = grade(J) = n$, its solution is a full-rank Gauss-Newton direction that is completely determined by the constraints in (5.3.16). When $rank(J) = grade(J) < n$, the search direction is computed as the sum of two mutually orthogonal components, p_Y and p_Z defined by equations (5.3.14) and (5.3.15). In this case $S_2 = 0$, so that the projected Hessian in (5.3.15) is $Z^{\mathrm{T}} B Z$ and therefore involves only the second derivatives of the residuals. We will return to this point in Chapter 6, when

we discuss SQP methods for nonlinear least squares. Although the range-space component solving (5.3.14) can never be a direction of increase for $f^T f$ (see Theorem (4.2-1)), the search direction computed by (5.3.14) and (5.3.15) may not be a descent direction for $f^T f$, regardless of whether or not $S_2^2 + Z^T B Z$ is modifed, on account of the p_Y term in (5.3.15). Thus, if $|cos(\bar{g}, p)|$ is smaller than some prescribed value, or if $\bar{g}^T p$ is positive, then a modified Newton search direction (corresponding to the case $k = 0$) is used instead. A finite-difference approximation to the projected matrix $Z^T B Z$ along the columns of Z, and a quasi-Newton approximation to B (see the discussion in Section 5.4) are given as alternatives to handle cases in which second derivatives of the residual functions are not available or are difficult to compute. Gill and Murray test their method on a set of twenty-three problems, and find that the version that uses quasi-Newton approximations to B does not perform as well as those that use exact second derivatives or finite-difference approximations to a projection of B. They observe only linear convergence on problems with large residuals. The algorithms sre implemented in the NAG Library [1984]; subroutine `E04HEF` uses exact second derivatives, while subroutine `E04GBF` is the quasi-Newton version.

5.4 Special Quasi-Newton Methods

Another approach to the nonlinear least-squares problem is a based on a quadratic model

$$\bar{g}^T p + \frac{1}{2} p^T (J^T J + \tilde{B}) p,$$

where $\tilde{B}$ involves quasi-Newton approximations to the term

$$B(x) = \sum_{i=1}^{m} \phi_i(x) \nabla^2 \phi_i(x)$$

in the Hessian of the nonlinear least-squares objective. Brown and Dennis [1971] first proposed a method in which the Hessian matrix of each of the residuals was updated separately. This approach is impractical because it entails the storage of m symmetric matrices of order n, and more recent research has aimed to approximate B as a sum.

Dennis [1973] suggests choosing the updates to satisfy a quasi-Newton condition

$$\tilde{B}_{k+1} s_k = y_k - J_{k+1}^T J_{k+1} s_k, \tag{5.4.1}$$

where

$$s_k \equiv x_{k+1} - x_k \quad \text{and} \quad y_k \equiv \bar{g}_{k+1} - \bar{g}_k.$$

It is implied that the update can then be chosen as in the unconstrained case (see Section 2.5.2), although there is some ambiguity as to the application of the update. One possibility is to update $\tilde{B}_k$ directly to obtain $\tilde{B}_{k+1}$, subject to a quasi-Newton condition such as (5.4.1) on $\tilde{B}_{k+1}s_k$. Another approach consistent with Dennis' description is to modify $\bar{H}_k = J_{k+1}^T J_{k+1} + \tilde{B}_k$, requiring the updated matrix $\bar{H}_{k+1}$ to satisfy a quasi-Newton condition

$$\bar{H}_{k+1}s_k = y_k. \tag{5.4.2}$$

Then $\tilde{B}_{k+1} = \bar{H}_{k+1} - J_{k+1}^T J_{k+1}$ is the new approximation to B at x_{k+1}. Depending on the update and quasi-Newton conditions, the two alternatives may not yield the same result. Moreover, updates defined by minimizing the change in the inverse of $\tilde{B}_k$, such as the BFGS update to $\tilde{B}_k$, make no sense in this context, since the matrix B would not, by itself, be expected to be invertible.

Betts [1976] implements a linesearch method in which the symmetric rank-one update (see Dennis and Moré [1977]) is applied to $\tilde{B}$, with the quasi-Newton condition

$$\tilde{B}_{k+1}s_k = y_k - J_k^T J_k s_k. \tag{5.4.3}$$

This scheme is equivalent to applying the symmetric rank-one formula to the matrix $\bar{H}_k = J_k^T J_k + \tilde{B}_k$ with the updated matrix $\bar{H}_{k+1}$ satisfying (5.4.2), and then taking $\tilde{B}_{k+1} = \bar{H}_{k+1} - J_k^T J_k$. Betts also proposes a hybrid algorithm that starts with Gauss-Newton, switching to the augmented Hessian $\bar{H}_k$ when the iterates are judged to be sufficiently close together to be near a solution. The criterion for the switch is

$$\|p_k\|_2 < \epsilon\,(1 + \|x_k\|_2), \tag{5.4.4}$$

for some $\epsilon < 1$. Results are presented for these methods, as well as for a Gauss-Newton method, on a set of eleven test problems. Betts concludes that the hybrid method is superior, especially on problems with nonzero residuals, although the results he lists in his tables do not all have the same value of ϵ in (5.4.4). In addition, he reports observing quadratic convergence for the special quasi-Newton methods. Issues that are not clarified include whether or not the update is performed when $\tilde{B}$ is not used in the hybrid method, and

treatment of near singularity or indefiniteness in the quadratic model in all of the methods tested. Also the test (5.4.4) may not necessarily imply that the Gauss-Newton iterates are in the vicinity of a solution, and could instead indicate inefficiency in the Gauss-Newton method at some arbitrary point.

Bartholomew-Biggs [1977] compares the PSB update (see Dennis and Moré [1977]) and the symmetric rank-one update applied directly to $\tilde{B}$ in a linesearch method. These updates are tested with the quasi-Newton condition (5.4.1), as well as with the condition

$$\tilde{B}_{k+1}s_k = J_{k+1}^{\mathrm{T}}f_{k+1} - J_k^{\mathrm{T}}f_{k+1}, \tag{5.4.5}$$

which is derived from the relation

$$\begin{aligned}\sum_{i=1}^{m}\phi_i(x_{k+1})\nabla^2\phi_i(x_{k+1})s_k &= \sum_{i=1}^{m}\phi_i(x_{k+1})\left[\nabla\phi_i(x_{k+1}) - \nabla\phi_i(x_k) + \mathcal{O}(\|s_k\|^2)\right] \\ &\approx J_{k+1}^{\mathrm{T}}f_{k+1} - J_k^{\mathrm{T}}f_{k+1}\end{aligned}$$

(see also Dennis [1976]). Bartholomew-Biggs points out that, in general, quasi-Newton approximations to B may not adequately reflect changes that are due to the contribution of the residuals. For example, when each residual function ϕ_i is quadratic, and consequently each $\nabla^2\phi_i$ is constant, B_{k+1} may differ from B_k by a matrix of rank n. For this reason, he does some experiments with updating $\tau\tilde{B}_k$ for $\tau = f_{k+1}^{\mathrm{T}}f_k/f_k^{\mathrm{T}}f_k$, which is the appropriate scaling for the special case in which $f_{k+1} = \tau f_k$ and the ϕ_i are quadratic. In his implementation, a Levenberg-Marquardt step is used whenever the linesearch fails to produce an acceptable reduction in the sum of squares and $cos(\bar{g}, p) > -10^{-4}$. The scaled symmetric rank-one update with (5.4.5) is selected to compare with other methods after preliminary tests, because it exhibited the best overall performance, and required fewer Levenberg-Marquardt steps. The other methods tested include a Gauss-Newton method, a method that combines Gauss-Newton with a Levenberg-Marquardt method, an implementation of Fletcher's [1971] Levenberg-Marquardt method, and a quasi-Newton method for unconstrained optimization. All of the fourteen test problems have nonzero residuals. Bartholomew-Biggs finds that the special quasi-Newton method is more robust than the other specialized methods for nonlinear least-squares, and that it is particularly suitable for problems with large residuals. He also observes that on problems on which the Gauss-Newton and Levenberg-Marquardt-based methods perform poorly, the special quasi-Newton method is more effective than the quasi-Newton method for general unconstrained optimization. Nothing is said about the

observed rate of convergence for any of the methods. He concludes that further research is needed to determine the best updating strategy, some desirable features being hereditary positive definiteness, and the ability to update a factorization of $\tilde{B}$. Finally, he indicates that it would be worthwhile to develop a hybrid method combining Gauss-Newton with a special quasi-Newton method, in order to avoid the cost of the updates on problems that are easily solved by Gauss-Newton methods.

Gill and Murray [1978] discuss a linesearch method in which they use the augmented Gauss-Newton quadratic model only to compute a component of the search direction in a subspace that approximates the null space of the Jacobian (see the preceding section). They apply the BFGS formula for unconstrained optimization (see Dennis and Moré [1977]) to the matrix $\bar{H}_k = J_{k+1}^T J_{k+1} + \tilde{B}_k$ with the quasi-Newton condition (5.4.2), and then form $\tilde{B}_{k+1} = \bar{H}_{k+1} - J_{k+1}^T J_{k+1}$. The choice of the BFGS update is based on performance comparisons to a number of other updates, including the symmetric rank-one update and Davidon's optimally-conditioned update [Davidon (1975)], as well as the symmetric rank-one update applied to $H_k = J_k^T J_k + \tilde{B}_k$ used in Betts [1976]. They point out that, if $J_{k+1}^T J_{k+1} + \tilde{B}_k$ is positive definite, and $s_k^T y_k > 0$, then $J_{k+1}^T J_{k+1} + \tilde{B}_{k+1}$ is also positive definite with this scheme. In order to safeguard the method, the projected approximate Hessian is replaced by a modified Cholesky factorization when it is singular or indefinite. In addition, if $cos(p, \bar{g})$ exceeds a fixed threshold value, a modified Newton step with the full augmented approximate Hessian is taken. See Section 5.3 for a summary of their observations on the performance of the methods.

Dennis, Gay, and Welsch [1981a] apply a scaled DFP update to $\tilde{B}_k$ at each step. The new approximation $\tilde{B}_{k+1}$ solves

$$\min_{B;H} \|H^{-1/2}(\tau_k \tilde{B}_k - B)H^{-1/2}\|_F \tag{5.4.6}$$

subject to

$$Hs_k = y_k\,; \qquad H \text{ positive definite} \tag{5.4.7}$$

$$Bs_k = J_{k+1}^T f_{k+1} - J_k^T f_{k+1}\,; \qquad B \text{ symmetric,} \tag{5.4.8}$$

where

$$\tau_k \equiv \min\{|y_k^T s_k / s_k^T \tilde{B}_k s_k|, 1\}. \tag{5.4.9}$$

The scale factor τ_k is based on the observation that the quasi-Newton approximation to B is often too large with the unscaled update, on account of the contribution of the residuals.

The term $|y_k^T s_k / s_k^T \tilde{B}_k s_k|$ in τ_k is derived from the self-scaling principles for quasi-Newton methods of Oren [1973], and attempts to shift the eigenvalues of the approximation $\tilde{B}_k$ to overlap with those of B_k, using new curvature information at x_k. This method forms the basis for the ACM computer program NL2SOL [Dennis, Gay, and Welsch (1981b)], which is distributed by the PORT Library [1984] as subroutines N2G and DN2G. It is implemented as an adaptive method, in that Gauss-Newton steps are taken if the Gauss-Newton quadratic model predicts the reduction in the function better than the quadratic model that includes the term involving $\tilde{B}$. A trust-region strategy (see Section 2.4.2) is used to enforce global convergence. Numerical results are given in Dennis, Gay, and Welsch [1981a] for a set of twenty-four test problems, many with two or three different starting values.

Al-Baali and Fletcher [1985] describe some linesearch methods that are similar to the method of Dennis, Gay, and Welsch [1981a] discussed above. They observe that the DFP update defined by (5.4.6) - (5.4.9) is equivalent to finding $\bar{H}_{k+1}$ to solve

$$\min_{\bar{H};H} \|H^{-1/2}(J_{k+1}^T J_{k+1} + \tau_k \tilde{B}_k - \bar{H})H^{-1/2}\|_F \tag{5.4.10}$$

subject to

$$H s_k = y_k; \quad H \text{ positive definite} \tag{5.4.11}$$

$$\bar{H} s_k = J_{k+1}^T f_{k+1} - J_k^T f_{k+1} + J_{k+1}^T J_{k+1} s_k; \quad \bar{H} \text{ symmetric,}$$

where

$$\tau_k \equiv \min\{|y_k^T s_k / s_k^T \tilde{B}_k s_k|, 1\},$$

and then forming

$$\tilde{B}_{k+1} = \bar{H}_{k+1} - J_{k+1}^T J_{k+1}.$$

Moreover, they use the condition

$$H s_k = \bar{y}_k; \quad H \text{ positive definite,} \tag{5.4.12}$$

with

$$\bar{y}_k \equiv J_{k+1}^T J_{k+1} s_k + J_{k+1}^T f_{k+1} - J_k^T f_{k+1} = y_k + \mathcal{O}(\|s_k\|^2) \tag{5.4.13}$$

as an alternative to (5.4.11), and mention that (5.4.11) has been replaced by (5.4.12) in newer versions of NL2SOL. A corresponding BFGS method is also given in which the update is defined by

$$\min_{\bar{H};H} \|H^{-1/2}((J_{k+1}^T J_{k+1} + \tau_k \tilde{B}_k)^{-1} - \bar{H}^{-1})H^{-1/2}\|_F$$

instead of (5.4.10). If the matrix $J_{k+1}^T J_{k+1} + \tau_k \tilde{B}_k$ is not positive semi-definite, τ_k is replaced by a quantity $\hat{\tau}_k$ that is calculated by a method similar to a Rayleigh quotient iteration, so that $J_{k+1}^T J_{k+1} + \hat{\tau}_k \tilde{B}_k$ is positive semi-definite and singular. The claim is that the updated matrix is almost always positive definite. They conclude from computational tests (described in Al-Baali [1984]) that their method is somewhat more efficient in terms of the number of Jacobian evaluations than NL2SOL, but requires more function evaluations, and that there is no significant difference between the DFP and BFGS updates. Al-Baali and Fletcher also introduce scaling factors based on finding a measure of the error in the inverse Hessian. They observe that, for the BFGS update for unconstrained optimization,

$$\left\| H_k^{-1/2}(H_k^{-1} - H_{k+1}^{-1})H_k^{-1/2} \right\|_F^2 = \Delta_k(H_k; y_k),$$

where

$$\Delta_k(H_k; y_k) \equiv \left(\frac{y_k^T H_k^{-1} y_k}{y_k^T s_k} \right)^2 - 2\frac{y_k^T s_k}{s_k^T H_k s_k} + 1. \tag{5.4.14}$$

Hence an "optimal" value of τ can be found by minimizing $\Delta_k J_{k+1}^T J_{k+1} + \tau \tilde{B}_k$ as a function of τ. Newton's method is used to find τ, an iterative process that requires factorization of $J_{k+1}^T J_{k+1} + \tau \tilde{B}_k$ for each intermediate value of τ. They were apparently unable to draw any broad conclusions from numerical experiments with this scaling, and refer to Al-Baali [1984] for details.

A convergence analysis for minimization algorithms based on a quadratic model in which part of the Hessian is computed by a quasi-Newton method is given by Dennis and Walker [1981] (see also Chapter 11 of Dennis and Schnabel [1983]). These results are restricted to methods that satisfy a least-change condition on the matrix $\tilde{B}_k$ (analogous to the PSB and DFP updates). Only a fairly mild assumption is needed to prove superlinear convergence to an isolated local minimum x^* : that the vector y_k^B in the quasi-Newton condition

$$\tilde{B}_k s_k = y_k^B$$

be chosen so that the norm of the update is

$$\mathcal{O}(\max\{\|x_k - x^*\|^p, \|x_{k+1} - x^*\|^p\}),$$

for some $p > 0$. This assumption is satisfied for y_k^B in each quasi-Newton update to $\tilde{B}_k$ described above. Their treatment of inverse updates is for the case in which part of the

inverse Hessian is computed, and hence does not apply here. To the best of our knowledge, no convergence results have yet been proven for scaled versions of the updates, or for updates to $J_{k+1}^T J_{k+1} + \tilde{B}_k$ that are not equivalent to some direct quasi-Newton update to $\tilde{B}_k$.

5.5 Other Approaches

So far, only methods that are applicable to general nonlinear least-squares problems, and for which software is widely available, have been discussed. In this section we briefly summarize some other relevant research.

5.5.1 Modifications of Unconstrained Optimization Methods

Besides Gauss-Newton methods, several straightforward modifications of unconstrained optimization methods are possible for nonlinear least squares. In quasi-Newton methods (see Section 2.5.2), $J_0^T J_0$ can be used as the initial approximation to the Hessian matrix. Ramsin and Wedin [1977] report favorable results with this technique. We note that a perturbed matrix $\tilde{J}_0^T \tilde{J}_0$ can be used as the initial approximate Hessian, where $\tilde{J}_0$ is a modified Cholesky factor of $J_0^T J_0$ (see Section 2.5.1), in order to maintain positive definiteness when J_0 is ill-conditioned.

Al-Baali and Fletcher [1985] suggest the use of $\bar{y}_k$ defined by (5.4.13) rather than y_k in the quasi-Newton condition (2.5.6). They report improvements with the BFGS and DFP formulas when this substitution is made. However, they remark that the condition $\bar{y}_k^T s_k > 0$ for hereditary positive definiteness of the updates is not guaranteed by the linesearch requirements, and they replace $\bar{y}_k^T s_k$ in the update formulas by $\max\{\bar{y}_k^T s_k, 0.01 y_k^T s_k\}$ as a safeguard. They do not consider this a major drawback, because $\bar{y}_k^T s_k > 0.01 y_k^T s_k$ almost always occurred in their examples. A modification of the safeguard is used in a later related paper [see Fletcher and Xu (1986) p. 26] discussed in Section 5.5.4.

Wedin [1974] (see also Ramsin and Wedin [1977]) suggests a modification of Newton's method in which the search direction is defined by

$$(J^T J + \sum_{i=1}^{m} \bar{\phi}_i \nabla^2 \phi_i) p = -\bar{g}, \tag{5.5.1}$$

where $\bar{\phi}_i$ is the ith component of the projection $\bar{f}$ of f onto $\mathcal{R}(J)$. This iteration approaches Newton's method in the limit, since $f(x^*) = \bar{f}(x^*)$, and is parameter-independent, in the

sense that minimization of f as a function of x is equivalent to minimization of f as a function of a new variable z, provided the mapping that defines x as a function of z has a nonsingular Jacobian. An obvious difficulty is that $\bar{f}$, and hence (5.5.1), is not well-defined when J is ill-conditioned.

5.5.2 Special Linesearches

Lindström and Wedin [1984] and Al-Baali and Fletcher [1986] propose specialized linesearch methods for nonlinear least-squares problems in which each residual is interpolated by a quadratic function, in contrast to the strategy of interpolating to the sum of squares used in conventional linesearches for unconstrained minimization. As a result a quartic polynomial, rather than a simpler cubic or quadratic, is minimized at each iteration of the linesearch.

Lindström and Wedin substitute their linesearch, which uses only function values, for the quadratic interpolation and cubic interpolation routines in the NAG Library (1980 version) nonlinear least-squares algorithm E04GBF (see Sections 5.3, 5.4, and 5.6.2), and compare the performance with the NAG linesearch routines on a set of eighteen test problems. They find that no linesearch algorithm is superior over all, but that their algorithm makes a better initial prediction to the steplength that minimizes the sum of squares along the search direction. In a second set of tests that includes multiple starting values for many of the test problems, they add a modified version of their linesearch algorithm that reverts to a simple backtracking strategy if an acceptable decrease in the sum of squares is not obtained after two function evaluations. They observe that their modified method requires fewer function evaluations than either of the NAG linesearch routines, and that the total for their original method falls between cubic interpolation and quadratic interpolation to the sum of squares. They note occasional inefficiencies in their methods due to extrapolation, but comment that such effects are more pronounced for quadratic interpolation of the sum of squares.

Al-Baali and Fletcher [1986] test similar linesearch methods that use gradients on a set of fifty-five test problems with a number of nonlinear least-squares algorithms described in Al-Baali [1984] (see also Al-Baali and Fletcher [1985]). They conclude that considerable overall savings can be made by interpolating to each of the residuals rather to than the sum of squares. They also obtain favorable results for two different schemes designed to save Jacobian evaluations in the new linesearch.

5.5.3 Conjugate-Gradient Acceleration of Gauss-Newton Methods

Ruhe [1979] uses preconditioned conjugate gradients to speed up convergence of Gauss-Newton methods. General references on conjugate gradients include Fletcher [1981], Chapter 4, and Gill, Murray, and Wright [1981], Chapter 4. We give a brief explanation below.

The conjugate gradient method minimizes an n-variate quadratic function

$$\mathcal{Q}(x) = q^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}Qp,$$

in at most n iterations. The iteration is

$$p_k = -g_k + \beta_{k-1}p_{k-1}; \tag{5.5.2}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

where

$$\alpha_k = \frac{\|g_k\|_2^2}{p_k^{\mathrm{T}}Qp_k};$$

$$\beta_k = \frac{\|g_{k+1}\|_2^2}{\|g_k\|_2^2};$$

$$g_k = \nabla\mathcal{Q}(x_k) = q + Qx_{k+1}.$$

The method produces a sequence of search directions that are *Q-conjugate*, that is

$$p_iQp_j = 0 \quad \text{if } i \neq j.$$

The number of iterations needed to minimize $\mathcal{Q}$ by conjugate gradients (with exact arithmetic) is equal to the number of distinct eigenvalues of Q. The idea of preconditioning is to transform Q into a matrix whose eigenvalues are nearly identical in magnitude. If a positive-definite matrix W is used as a *preconditioner*, then convergence occurs in the same number of steps that would be taken for a quadratic function with the Hessian matrix

$$W^{-1/2}QW^{-1/2}.$$

The ideal preconditioner would be $W = Q$, but since conjugate gradients are competitive mainly when n is large, an approximation that is relatively inexpensive to factorize is used. For a smooth nonlinear function $\mathcal{F}(x)$, the conjugate gradient method (5.5.2) can also be applied, with $g_k = \nabla\mathcal{F}(x_k)$ and α_k determined by a linesearch, with safeguards to ensure descent. There are several possible choices for β_k that are equivalent to the one given

above for the quadratic case (see, for example, Fletcher [1981], Chapter 4). The method is restarted every n iterations on account to the loss of conjugacy that occurs with inexact arithmetic (see, for example, Gill, Murray, and Wright [1981], Chapter 4). Preconditioners for the nonlinear case attempt to approximate $\nabla^2 \mathcal{F}(x)$.

In Ruhe's algorithm, the matrix $J^T J$ is used as the preconditioner, and an orthogonal factorization of J is used to compute the necessary quantities. The method is applied to problems in which the residuals are nonzero and the Jacobian has full rank, and is restarted every n iterations. He concludes that the preconditioned conjugate-gradient method never increases the total number of iterations required to solve a given problem relative to Gauss-Newton, and that significant improvements in the speed of linear convergence of Gauss-Newton on large-residual problems can be achieved with conjugate-gradient acceleration.

Al-Baali and Fletcher [1985] point out that conjugate-gradient acceleration of the type described by Ruhe is equivalent to applying a BFGS update to the Gauss-Newton approximate Hessian $J^T J$ at each step. They implement and test both this method (without restarts) and a scaled version, where the scale parameter τ is chosen to minimize $\Delta_k(\tau J_k^T J_k; \bar{y}_k)$ as a function of τ (see (5.4.14)). They give no conclusions as to the relative efficiency of the scaled and unscaled versions of the method, but find that the modified methods offer some improvement over Gauss-Newton, while exhibiting the same difficulties.

5.5.4 Hybrid Methods

Nazareth [1980, 1983] describes a hybrid method that combines the Levenberg-Marquardt method with a quasi-Newton approximation H_k to the full Hessian. The search directions solve a system of the form

$$\left(\theta_k J_k^T J_k + (1-\theta_k) H_k + \lambda_k D_k^T D_k\right) p = -\bar{g}_k,$$

with $\theta_k \in [0,1]$ and $\lambda_k \geq 0$. He compares the reduction in the sum of squares predicted by both the Levenberg-Marquardt and quasi-Newton models with the actual reduction, and then chooses θ_k on the basis of this comparison. In Nazareth [1983], a simple version of the hybrid strategy is implemented that uses Davidon's optimally conditioned update, with $D_k = I$, and a variation of Fletcher's [1971] method for updating λ. Results are reported for a set of eleven test problems — including five problems with nonzero residuals — and

compared to the use of the algorithm as a quasi-Newton method ($\theta_k = 0$) or a Levenberg-Marquardt method ($\theta_k = 1$). He concludes that the hybrid method is somewhat better for the nonzero residual problems, and recommends development of a more sophisticated implementation.

Al-Baali and Fletcher [1985] develop several hybrid linesearch methods in which the models are assessed in terms of the function Δ_k defined by (5.4.14), an approximate measure of the error in the inverse Hessian. In one class of methods, the modified BFGS update (which uses $\bar{y}_k$ defined by (5.4.13) rather than y_k; see Section 5.5.1) is applied to a matrix of the form

$$H_{k+1} = (1 - \theta_k)H_k + \theta_k \tau_k J_{k+1}^{\mathrm{T}} J_{k+1},$$

where τ_k minimizes $\Delta_k(\tau_k J_{k+1}^{\mathrm{T}} J_{k+1}; \bar{y}_k)$, and θ_k is chosen to minimize $\Delta_k(H_{k+1}; \bar{y}_k)$, in order to obtain the new approximate Hessian. In their implementation, in which θ_k is restricted to be either 0 or 1, they find that the method has difficulties on singular problems, and that the scaling of the search direction often does not allow $\alpha = 1$ as a trial step in the linesearch (see Section 2.4.1). They refer to Al-Baali [1984] for more details of the tests.

Another class of hybrid methods defined by Al-Baali and Fletcher compares the value

$$\Delta_{QN} \equiv \Delta_k(H_k; \bar{y}_k)$$

for the current quasi-Newton approximation H_k with

$$\Delta_{GN} \equiv \Delta_k(J_{k+1}^{\mathrm{T}} J_{k+1}; \bar{y}_k)$$

for the Gauss-Newton approximation. The basic algorithm can be summarized as follows :

$$\begin{aligned} &\textbf{if } \Delta_{QN} < \Delta_{GN} \textbf{ then } \text{use the modified BFGS search direction} \\ &\qquad\qquad \textbf{else } \text{use the Gauss-Newton search direction} \end{aligned} \tag{5.5.3}$$

They test several versions of this method that differ in the action taken whenever a switch from Gauss-Newton to quasi-Newton takes place. In one, H_{k+1} is reset to $J_{k+1}^{\mathrm{T}} J_{k+1}$, while in another H_{k+1} is reset to the result of applying the modified BFGS update to $J_{k+1}^{\mathrm{T}} J_{k+1}$ (conjugate-gradient acceleration). They observe little difference in performance between these two alternatives, and find them to be the best of the many methods for nonlinear least squares treated in their study. A version of the first strategy that substitutes the quantity $\min_\tau \Delta_k(\tau J_{k+1}^{\mathrm{T}} J_{k+1}; \bar{y}_k)$ for Δ_{GN} in the comparison with Δ_{QN} is also tried, but

it is found to have some difficulties on a problem for which the Jacobian is singular at the solution. A final variant maintains the quasi-Newton update throughout, and never resets the approximate Hessian. They find that this method is not as efficient as the others on some types of large-residual problem.

Fletcher and Xu [1986] give an example in which the hybrid method (5.5.3) has a linear rate of convergence when the BFGS method would converge superlinearly. The difficulty is that the comparison between Δ_{QN} and Δ_{GN} may fail to distinguish between zero-residual problems and those with nonzero residuals. They propose two new hybrid algorithms and show them to be superlinearly convergent. The first algorithm computes the modified BFGS search direction if

$$\frac{\|f(x_k)\|_2 - \|f(x_{k+1})\|_2}{\|f(x_k)\|_2} < \sigma, \tag{5.5.4}$$

for some fixed $\sigma \in (0,1)$, and a Gauss-Newton step otherwise. The method is motivated by the following relationship

$$\lim_{k\to\infty} \frac{\|f(x_k)\|_2 - \|f(x_{k+1})\|_2}{\|f(x_k)\|_2} = \begin{cases} 0, & \text{if } \|f(x^*)\|_2 \neq 0; \\ 1, & \text{if } \|f(x^*)\|_2 = 0. \end{cases}$$

The second algorithm computes a modified BFGS step if

$$\frac{\|f(x_k) - f(x_{k+1})\|_2}{\|f(x_k)\|_2} < \sigma \quad \text{and} \quad \frac{\Delta_k(J_{k+1}^T J_{k+1}; \bar{y}_k)}{\Delta_k(J_k^T J_k; \bar{y}_k)} \geq \gamma, \tag{5.5.5}$$

where both σ and γ are fixed parameters in $(0,1)$, and a Gauss-Newton step otherwise. The additional condition for choosing the BFGS search direction is derived from another asymptotic relationship

$$\lim_{k\to\infty} \frac{\Delta_k(J_{k+1}^T J_{k+1}; \bar{y}_k)}{\Delta_k(J_k^T J_k; \bar{y}_k)} = \begin{cases} 0, & \text{if } \|f(x^*)\|_2 = 0; \\ 1, & \text{if } \|f(x^*)\|_2 \neq 0. \end{cases}$$

Numerical results are given for set of fifty-six test problems, a few with multiple starting values. They conclude that the new methods offer some overall improvement over those based on (5.5.3), but that there is no reason to prefer the more complicated test (5.5.5) over (5.5.4).

5.5.5 Continuation Methods

Continuation methods have also been applied to nonlinear least-squares problems. These methods solve a sequence of parameterized subproblems

$$\min \Phi(x; \tau_i); \quad i = 1, 2, \ldots, i_{\max} \tag{5.5.6}$$

where

$$0 = \tau_0 < \tau_1 < \ldots < \tau_{i_{max}} = 1$$

and

$$\arg\min \Phi(x;0) = x_0 \quad \text{and} \quad \arg\min \Phi(x;1) = x^*.$$

The idea is that methods that have fast local convergence, but may not be robust in a global sense, can be applied to solve each subproblem in relatively few steps, because information from the solution of previous subproblems may be used to predict a good starting value for the next one.

DeVilliers and Glasser [1981] define

$$\Phi(x;\tau) \equiv \frac{1}{2}\,\|f(x)\|_2^2 + \frac{1}{2}\,(\tau^k - 1)\,\|f(x_0)\|_2^2 \tag{5.5.7}$$

where k is a positive integer, with a fixed spacing between the parameters τ_i in (5.5.6). They test two different continuation methods, one that uses Newton's method (with linesearch) to solve the intermediate problems, and one that uses a Gauss-Newton method (with linesearch). An unspecified "device" is included in the implementation of both minimization techniques to ensure a decrease in the objective at every iteration. The continuation methods are compared with results obtained by applying both minimization algorithms to the original problem. Intermediate subproblems are not solved exactly; the criterion

$$\|\nabla_x \Phi(x;\tau_i)\|_2 \leq \epsilon_i,$$

where $\epsilon_i = 10^{-2}$ if $i < i_{max}$, and $\epsilon_{i_{max}} = 10^{-6}$, is used to determine convergence of a subproblem.

Numerical experiments are carried out on three different test problems, with multiple starting values, most of which are points of failure for both Newton's method and Gauss-Newton. They conclude that, although the continuation method is less efficient than the underlying method when both are successful, it will converge on many problems for which the underlying method fails when used alone. However, the results they present are for different values of the step size, and the exponent k, and no mechanism is given for the automatic choice of either of the parameters. Hence there is no indication that the method is robust in a practical sense. DeVilliers and Glasser point out that their methods may require modification if the optimization method that is used to solve the subproblems encounters

difficulties, or if the continuation path is not well-behaved. We have observed that the first two test problems of DeVilliers and Glasser are very sensitive to the choice of the maximum step bound, or the initial trust-region size for most methods (see the results for problems 42 and 43 in Sections 2.6, 4.7, and 5.6, as well as the discussion in Section 5.7), and that the methods can be quite efficient provided an appropriate non-default choice is made for these parameters.

Salane [1986] incorporates a trust-region strategy into a continuation method by defining

$$\Phi(x;\tau) \equiv \frac{1}{2}\left(\|f(x)\|_2 + (\tau-1)\,\|f(x_0)\|_2^2 + \lambda(\tau-1)\,\|D(x-x_0)\|_2^2\right), \qquad (5.5.8)$$

and then applying Gauss-Newton to this function for the inner iterations. Instead of allowing the continuation parameter τ to range from 0 to 1, he advocates stopping when it becomes inefficient to solve the subproblems, and then restarting the method after replacing x_0 by the new iterate. He points out that that his approach is especially suitable for large-residual problems, because it transforms the original problem into a sequence of subproblems with small residuals. The idea is to attempt to determine when the neglected terms become significant, and then pose a new subproblem. An initial value, τ_1, of the continuation parameter must be supplied by the user in order to start the method. Should any step fail to obtain a decrease in either the nonlinear least-squares objective or its gradient, τ_1 is decreased, and the calculation is repeated without changing x_0. Theorems on descent conditions and convergence are presented. Salane argues that his continuation method allows direct selection of the Levenberg-Marquardt parameter λ in (5.5.8), because λ may be chosen so that the term $\lambda(1-\tau)D^{\mathrm{T}}D$ behaves somewhat like the second-order terms that have been neglected in the Hessian of $\Phi(x;\tau)$. However, no mechanism is suggested for automatic choice of λ, and $\lambda = \|f(x_0)\|_2$ is used in the tests.

Salane gives test results for a version of his algorithm on a set of nine problems (all of which are included in our set). A comparison is made to results obtained from MINPACK, and also to the results reported by DeVilliers and Glasser [1981] for two of the test problems. He concludes that the performance of the method compares favorably with that of MINPACK, and is superior to the DeVilliers and Glasser continuation method on the relevant problems. The matrix D in (5.5.8) is taken to be the identity matrix throughout the tests, and for one test problem a type of variable scaling is used. No information is given concerning scaling for the MINPACK tests. The results that are presented correspond to

several different values of τ_1, although the criterion used in choosing this value is not given. Test results in which the value of τ_1 is varied are included for three of the problems for the purpose of showing that performance is sensitive to the specification of the continuation parameter.

5.5.6 Methods for Special Problem Classes

Algorithms have also been formulated to treat some special cases of the nonlinear least-squares problem. For example, there is a vast literature concerning methods specific to nonlinear equations that we shall make no attempt to survey here.

In some nonlinear least-squares problems, the vector x can be separated into two sets of variables, say

$$x = \begin{pmatrix} y \\ z \end{pmatrix}$$

where it is relatively easy to minimize the sum of squares as a function of y alone. A fairly common situation of this type is one in which y is the set of variables that occur linearly in all of the residuals, so that

$$\min_y \left\| f\begin{pmatrix} y \\ z \end{pmatrix} \right\|_2^2$$

is a linear least-squares problem. For example, exponential fitting problems (see Varah [1985]) fall into this category. Methods that deal with separable nonlinear least-squares problems are reviewed and extended in Ruhe and Wedin [1980]. They describe three basic algorithms, all of which use Gauss-Newton to minimize the sum of squares as a function of y. The methods differ in the definition of the quadratic model function for minimization with respect to z. The Jacobian and Hessian of the nonlinear least-squares objective can be partitioned as follows:

$$J = \begin{pmatrix} J_y & J_z \end{pmatrix}$$

$$\nabla^2\left(\frac{1}{2}f^{\mathrm{T}}f\right) \equiv G = \begin{pmatrix} G_{yy} & G_{zy}^{\mathrm{T}} \\ G_{zy} & G_{zz} \end{pmatrix}$$

$$= J^{\mathrm{T}}J + B = \begin{pmatrix} J_y^{\mathrm{T}}J_y & J_y^{\mathrm{T}}J_z \\ J_z^{\mathrm{T}}J_y & J_z^{\mathrm{T}}J_z \end{pmatrix} + \begin{pmatrix} B_{yy} & B_{zy}^{\mathrm{T}} \\ B_{zy} & B_{zz} \end{pmatrix}$$

so that

$$\nabla_z f\begin{pmatrix} y \\ z \end{pmatrix} = J_z^{\mathrm{T}}f,$$

$$\nabla_{zz} f\begin{pmatrix} y \\ z \end{pmatrix} = G_{zz} - G_{zy}^{\mathrm{T}} G_{yy}^{-1} G_{zy}$$
$$= (J_z^{\mathrm{T}} J_z + B_{zz}) - (J_z^{\mathrm{T}} J_y + B_{zy})^{\mathrm{T}} (J_y^{\mathrm{T}} J_y + B_{yy})^{-1} (J_z^{\mathrm{T}} J_y + B_{zy}).$$

The approximate Hessians for the minimization as a function of z are

$$J_z^{\mathrm{T}} J_z - G_{zy}^{\mathrm{T}} (J_y^{\mathrm{T}} J_y)^{-1} G_{zy}, \tag{5.5.9}$$

$$J_z^{\mathrm{T}} J_z - J_y^{\mathrm{T}} J_z (J_y^{\mathrm{T}} J_y)^{-1} J_z^{\mathrm{T}} J_y, \tag{5.5.10}$$

and

$$J_z^{\mathrm{T}} J_z. \tag{5.5.11}$$

The algorithms based on (5.5.9) and (5.5.10) are shown to converge at a faster rate than the conventional Gauss-Newton method, while the asymptotic convergence rate for (5.5.11) may be much slower. On the other hand, of the three quadratic models, it is least expensive to compute solutions with the approximate Hessian (5.5.12), and most expensive to compute them from (5.5.9). Use of (5.5.10) costs about the same as a conventional Gauss-Newton method. Tests on four sample problems are given to illustrate rates of convergence.

Davidon [1976] introduced a quasi-Newton method for problems in which $m \gg n$, location of the minimum is not very sensitive to weighting of the residuals, and rapid approach to a minimum is more important than convergence to it. A new estimate of the minimum is computed after each individual residual and its gradient are evaluated, rather than after evaluating the entire block of m residuals. Davidon gives an analogy to time-dependent measurements of experimental data, in which quantities calculated from the measurements are updated each time a new observation is made. Starting from an initial quadratic approximation

$$q_0(x) = f(x_0)^{\mathrm{T}} f(x_0) + (x - x_0)^{\mathrm{T}} H_0^{-1} (x - x_0),$$

with H_0 positive-definite, the algorithm that determines the next iterate is equivalent to minimizing a quadratic function of the form

$$q_{k+1}(x) = [\phi_j(x_k) + (x - x_k)^{\mathrm{T}} \nabla \phi_j(x_k)]^2 + \lambda_k q_k(x),$$

where λ_k is in $(0, 1]$. It is suggested that the choice of $\{\lambda_k\}$ should be problem-dependent, and a number of alternatives are proposed. Davidon tests the method on a set of four

problems in which he varies the size of the problem, the initial estimate of the solution, and the sequence $\{\lambda_k\}$. He observes that the method tends to oscillate about a minimum rather than converging to it, but that it often reduces the sum of squares more rapidly than other methods.

Further computational experiments with Davidon's method are reported in Cornwell, Kocman, and Prosser [1980]. On a set of fifteen zero-residual problems, they test the method with various fixed values of λ_k. They obtain overflow in most cases for small values, but otherwise find that the efficiency of the method decreases as λ_k is increased. In one case, the method cycled through a sequence of points that was not near-optimal. On the basis of these observations, they implement a new version that attempts to use a fixed, relatively small value of λ_k, restarting from the initial vector with a larger value if it is determined that overflow would otherwise occur. They find that this modified implementation of Davidon's method is competitive with the computer program LMCHOL from Argonne National Laboratory based on Fletcher's [1971] Levenberg-Marquardt algorithm (which has since been superseded by the MINPACK routine LMDER [Moré, Garbow, and Hillstrom (1980)]).

5.6 Numerical Results

In this section numerical results are presented for particular implementations of the methods discussed in sections 2, 3, and 4 of this chapter. The tests were performed using the following software (described in more detail in the next three subsections) :

method	derivatives	subroutine	source
Levenberg-Marquardt	first	LMDER	MINPACK
corrected Gauss-Newton	second	LSQSDN/E04HEF	NPL/NAG
corrected Gauss-Newton	first	LSQFDQ/E04GBF	NPL/NAG
special quasi-Newton	first	DN2G/NL2SOL	PORT/ACM

In the tables, we include the quantity

$$\frac{\|f^*\|_2^2 - \|f_{best}\|_2^2}{1 + \|f_{best}\|_2^2}, \tag{5.6.1}$$

where f^* is the value of f at the point of termination, and $\|f_{best}\|_2$ is the best available estimate of the norm of the solution, in order to get some idea of the error in $\|f^*\|_2$. For those problems that have nonzero residuals, the value of $\|f_{best}\|_2$ is given to six figures of accuracy, rounded down.

For further details on the numerical tests, see Section 1.3, as well as the individual description of each method that follows. For information on the test problems, see the Appendix.

These results are discussed in Section 5.7, where they are compared with the unconstrained methods of Chapter 2, and the Gauss-Newton methods of Chapter 4.

5.6.1 Levenberg-Marquardt Method (MINPACK LMDER)

5.6.1.1 Software and Algorithm

The results were obtained using the MINPACK subroutine LMDER, which implements a Levenberg-Marquardt method using exact derivative information. A subproblem of the form

$$\min_{p \in \Re^n} Q_k(p) \equiv 2\bar{g}_k^{\mathrm{T}} p + p^{\mathrm{T}} J_k^{\mathrm{T}} J_k p$$

$$\text{subject to} \quad \|D_k p\|_2 \leq \delta_k$$

is solved at each iteration for the step p_k to the next iterate, where D_k is a diagonal scaling matrix.

5.6.1.2 Parameters

The results were obtained using the MINPACK subroutine LMDER, with the following input parameters :

XTOL	-	varied, see tables	accuracy in x
FTOL	-	varied, see tables	accuracy in sum of squares
GTOL	-	0.00	gradient norm tolerance
MAXFEV	-	min $\{9999, 1000 * n\}$	function evaluation limit
MODE	-	1	specifies internal scaling
FACTOR	-	100. (default)	initial step magnification

† In some cases the default FACTOR = 100.0 was too large and overflow occurred during function evaluation. These cases are indicated in the table by giving the lower value of FACTOR that was subsequently used to obtain the results.

For details about these parameters, Moré, Garbow, and Hillstrom [1980].

5.6.1.3 Convergence Criteria

The following quantities will be used in describing the convergence criteria :

residual vector	:	$f(x_k)$
ith residual gradient	:	$\nabla\phi_i(x_k)$
Jacobian matrix	:	$J(x_k)$
objective function	:	$\mathcal{F}(x_k) \equiv f(x_k)^{\mathrm{T}} f(x_k)$
objective gradient	:	$\bar{g}_k = \nabla\mathcal{F}(x_k) \equiv 2J(x_k)^{\mathrm{T}} f(x_k)$
current step	:	p_k, the minimizer of the subproblem
predicted reduction	:	$\rho_P = \dfrac{\|f_k\|_2 - \|f_k + J_k p_k\|_2}{\|f_k\|_2} = \dfrac{-\mathcal{Q}_k(p_k)}{\|f_k\|_2}$
actual reduction	:	$\rho_A = \dfrac{\|f(x_k)\|_2 - \|f(x_k + p_k)\|_2}{\|f_k\|_2} = \dfrac{\mathcal{F}(x_k) - \mathcal{F}(x_k + p_k)}{\|f_k\|_2}$

Criteria for termination of LMDER at x_k are as follows :

- *$\mathcal{F}$ convergence.* Both actual and predicted reductions in the sum of squares are at most FTOL.

$$|\rho_A| \leq \texttt{FTOL} \text{ and } \rho_P \leq \texttt{FTOL} \text{ and } \rho_A \leq 2\rho_P \tag{5.6.1}$$

This attempts to guarantee that

$$\|f_k\|_2 \leq (1 + \texttt{FTOL})\, \|f(x^*)\|_2 .$$

- *x convergence.* Relative error between two consecutive iterates is at most XTOL.

$$\delta_{k+1} \leq \texttt{XTOL}\, \|x_k + p_k\|_2 \tag{5.6.2}$$

This attempts to guarantee that

$$\|D_k(x_k - x^*)\|_2 \leq \texttt{XTOL}\, \|D_k(x^*)\|_2 .$$

- The cosine of the angle between f_k and any column of J_k is at most GTOL in absolute value.

$$\max_{1 \leq i \leq m} \frac{|\nabla\phi_i(x_k)^{\mathrm{T}} f_k|}{\|\nabla\phi_i(x_k)\|_2 \, \|f_k\|_2} \leq \texttt{GTOL} \tag{5.6.3}$$

This approximates the necessary condition $\bar{g}(x_k) = 0$.

- FTOL is too small. No further reduction in the sum of squares is possible.

$$|\rho_A| \leq \epsilon_M \text{ and } \rho_P \leq \epsilon_M \text{ and } \rho_A \leq 2\rho_P \tag{5.6.4}$$

- XTOL is too small. No further improvement in the approximate solution x_k is possible.

$$\delta_{k+1} \leq \epsilon_M \|x_k + p_k\|_2 \tag{5.6.5}$$

- GTOL is too small. f_k is orthogonal to the columns of J_k to machine precision.

$$\max_{1 \leq i \leq n} \frac{|\nabla \phi_i(x_k)^T f_k|}{\|\nabla \phi_i(x_k)\|_2 \|f_k\|_2} \leq \epsilon_M \tag{5.6.6}$$

Except for test (5.6.3), tests for convergence are performed only when

$$\rho_A < 0.0001 \rho_P. \tag{5.6.7}$$

The convergence criteria are described in more detail in Moré, Garbow, and Hillstrom [1980]. The following abbreviations are used in the tables to describe the conditions under which the algorithm terminates :

F	-	(5.6.1) and (5.6.7)
X	-	(5.6.2) and (5.6.7)
X, F	-	(5.6.1) and (5.6.2) and (5.6.7)
G	-	(5.6.6) and (5.6.7)
F LIM.	-	function evalutaion limit reached

Numerical Results for LMDER

	n	m	TOL	max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-8}		22	16	1.41	10^{-16}	10^{-15}	10^{-32}	x
			10^{-12}		22	16	1.41	10^{-16}	10^{-15}	10^{-32}	x
2.^{0}	2	2	10^{-8}		14	8	11.4	10^{1}	10^{-3}	10^{1}	REL. F
			10^{-12}		21	13	11.4	10^{1}	10^{-5}	10^{1}	REL. F
3.^{0}	2	2	10^{-8}		19	17	9.11	10^{-16}	10^{-11}	10^{-32}	x
			10^{-12}		19	17	9.11	10^{-16}	10^{-11}	10^{-32}	x
4.^{0}	2	3	10^{-8}		40	39	10^{6}	10^{-3}	10^{-2}	10^{-5}	x
			10^{-12}		54	53	10^{6}	10^{-7}	10^{-7}	10^{-14}	x
5.^{0}	2	3	10^{-8}		9	7	3.04	10^{-16}	10^{-15}	10^{-32}	x
			10^{-12}		10	8	3.04	10^{-16}	10^{-15}	10^{-32}	x
6.	2	10	10^{-8}		21	12	.365	10^{1}	10^{-2}	10^{-6}	REL. F
			10^{-12}		28	16	.365	10^{1}	10^{-4}	10^{-6}	REL. F
7.^{0}	3	3	10^{-8}		11	8	1.00	10^{-16}	10^{-15}	10^{-32}	x
			10^{-12}		12	9	1.00	10^{-32}	10^{-31}	10^{-64}	x
8.	3	15	10^{-8}		6	5	2.60	10^{-1}	10^{-9}	10^{-8}	REL. F
			10^{-12}		7	6	2.60	10^{-1}	10^{-11}	10^{-8}	REL. F
9.	3	15	10^{-8}		4	3	1.08	10^{-4}	10^{-16}	10^{-14}	x
			10^{-12}		5	4	1.08	10^{-4}	10^{-16}	10^{-14}	x, REL. F
10.	3	16	10^{-8}		126	116	10^{4}	10^{1}	10^{0}	10^{-6}	x, REL. F
			10^{-12}		126	116	10^{4}	10^{1}	10^{0}	10^{-6}	x
11.^{0}	3	10	10^{-8}		(3000)	(2956)	239.	10^{-2}	10^{18}	10^{-5}	F LIM.
			10^{-12}		(3000)	(2956)	239.	10^{-2}	10^{18}	10^{-5}	F LIM.
12.^{0}	3	10	10^{-8}		7	6	10.1	10^{-16}	10^{-16}	10^{-32}	x
			10^{-12}		8	7	10.1	10^{-16}	10^{-16}	10^{-32}	x
13.^{0}	4	4	10^{-8}		65	60	10^{-17}	10^{-34}	10^{-50}	10^{-67}	G
			10^{-12}		65	60	10^{-17}	10^{-34}	10^{-50}	10^{-67}	G
14.^{0}	4	6	10^{-8}		70	64	2.00	0.00	0.00	0.00	x
			10^{-12}		70	64	2.00	0.00	0.00	0.00	x
15.	4	11	10^{-8}		18	16	.328	10^{-2}	10^{-7}	10^{-9}	REL. F
			10^{-12}		28	26	.328	10^{-2}	10^{-9}	10^{-9}	x
16.	4	20	10^{-8}		264	245	17.6	10^{2}	10^{0}	10^{-7}	REL. F
			10^{-12}		356	329	17.6	10^{2}	10^{-2}	10^{-8}	REL. F
17.	5	33	10^{-8}		18	15	2.46	10^{-2}	10^{-8}	10^{-11}	REL. F
			10^{-12}		19	16	2.46	10^{-2}	10^{-9}	10^{-11}	REL. F
18.^{0}	6	13	10^{-8}		46	32	12.3	10^{-16}	10^{-15}	10^{-31}	x
			10^{-12}		46	32	12.3	10^{-16}	10^{-15}	10^{-31}	x
19.	11	65	10^{-8}		17	13	9.38	10^{-1}	10^{-7}	10^{-8}	REL. F
			10^{-12}		19	15	9.38	10^{-1}	10^{-9}	10^{-8}	REL. F

Numerical Results for LMDER

	n	m	TOL	max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-8}		8	7	2.44	10^{-2}	10^{-7}	10^{-10}	REL. F
			10^{-12}		10	9	2.44	10^{-2}	10^{-9}	10^{-10}	X, REL. F
20b.	9	31	10^{-8}		9	8	6.06	10^{-3}	10^{-13}	10^{-13}	X
			10^{-12}		10	9	6.06	10^{-3}	10^{-13}	10^{-13}	X, REL. F
20c.	12	31	10^{-8}		10	9	16.6	10^{-5}	10^{-13}	10^{-16}	X
			10^{-12}		12	10	16.6	10^{-5}	10^{-13}	10^{-16}	X
20d.	20	31	10^{-8}		18	14	247.	10^{-10}	10^{-12}	10^{-24}	X
			10^{-12}		23	15	247.	10^{-10}	10^{-12}	10^{-25}	X
21a.^{0}	10	10	10^{-8}		22	16	3.16	10^{-16}	10^{-14}	10^{-31}	X
			10^{-12}		22	16	3.16	10^{-16}	10^{-14}	10^{-31}	X
21b.^{0}	20	20	10^{-8}		22	16	4.47	10^{-16}	10^{-14}	10^{-31}	X
			10^{-12}		22	16	4.47	10^{-16}	10^{-14}	10^{-31}	X
22a.^{0}	12	12	10^{-8}		72	63	10^{-17}	10^{-34}	10^{-51}	10^{-68}	G
			10^{-12}		72	63	10^{-17}	10^{-34}	10^{-51}	10^{-68}	G
22b.^{0}	20	20	10^{-8}		69	60	10^{-17}	10^{-33}	10^{-49}	10^{-66}	X
			10^{-12}		69	60	10^{-17}	10^{-33}	10^{-49}	10^{-66}	G
23a.	4	5	10^{-8}		34	23	.500	10^{-3}	10^{-9}	10^{-10}	REL. F
			10^{-12}		44	28	.500	10^{-3}	10^{-11}	10^{-10}	REL. F
23b.	10	11	10^{-8}		84	67	.500	10^{-2}	10^{-8}	10^{-11}	REL. F
			10^{-12}		104	82	.500	10^{-2}	10^{-10}	10^{-11}	REL. F
24a.	4	8	10^{-8}		151	113	.759	10^{-3}	10^{-8}	10^{-11}	REL. F
			10^{-12}		156	116	.759	10^{-3}	10^{-11}	10^{-11}	REL. F
24b.	10	20	10^{-8}		80	62	.598	10^{-2}	10^{-7}	10^{-9}	REL. F
			10^{-12}		88	67	.598	10^{-2}	10^{-10}	10^{-9}	REL. F
25a.^{0}	10	12	10^{-8}		11	10	3.16	10^{-15}	10^{-14}	10^{-30}	X
			10^{-12}		12	11	3.16	10^{-16}	10^{-14}	10^{-31}	X
25b.^{0}	20	22	10^{-8}		13	12	4.47	10^{-15}	10^{-13}	10^{-30}	X
			10^{-12}		14	13	4.47	10^{-15}	10^{-14}	10^{-31}	X
26a.^{0}	10	10	10^{-8}		28	16	.328	10^{-2}	10^{-7}	10^{-5}	REL. F
			10^{-12}		37	21	.328	10^{-2}	10^{-9}	10^{-5}	REL. F
26b.^{0}	20	20	10^{-8}		57	40	.228	10^{-3}	10^{-8}	10^{-6}	REL. F
			10^{-12}		71	45	.228	10^{-3}	10^{-10}	10^{-6}	REL. F
27a.^{0}	10	10	10^{-8}		15	13	3.18	10^{-15}	10^{-14}	10^{-31}	X
			10^{-12}		15	13	3.18	10^{-15}	10^{-14}	10^{-31}	X
27b.^{0}	20	20	10^{-8}		5	2	19.7	10^{0}	10^{-12}	10^{0}	REL. F
			10^{-12}		18	15	4.47	10^{-14}	10^{-13}	10^{-28}	X
28a.^{0}	10	10	10^{-8}		5	4	.412	10^{-17}	10^{-16}	10^{-33}	X
			10^{-12}		5	4	.412	10^{-17}	10^{-16}	10^{-33}	X
28b.^{0}	20	20	10^{-8}		5	4	.571	10^{-17}	10^{-16}	10^{-33}	X
			10^{-12}		5	4	.571	10^{-17}	10^{-16}	10^{-32}	X

Numerical Results for LMDER

	n	m	TOL	max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
29a.^{0}	10	10	10^{-8}		5	4	.412	10^{-17}	10^{-16}	10^{-33}	x
			10^{-12}		5	4	.412	10^{-17}	10^{-16}	10^{-33}	x
29b.^{0}	20	20	10^{-8}		5	4	.571	10^{-16}	10^{-16}	10^{-32}	x
			10^{-12}		5	4	.571	10^{-16}	10^{-16}	10^{-32}	x
30a.^{0}	10	10	10^{-8}		6	5	2.05	10^{-16}	10^{-15}	10^{-31}	x
			10^{-12}		7	6	2.05	10^{-16}	10^{-15}	10^{-31}	x
30b.^{0}	20	20	10^{-8}		6	5	3.04	10^{-15}	10^{-14}	10^{-30}	x
			10^{-12}		7	6	3.04	10^{-15}	10^{-14}	10^{-30}	x
31a.^{0}	10	10	10^{-8}		7	6	1.80	10^{-16}	10^{-15}	10^{-31}	x
			10^{-12}		8	7	1.80	10^{-16}	10^{-15}	10^{-31}	x
31b.^{0}	20	20	10^{-8}		7	6	2.66	10^{-15}	10^{-14}	10^{-30}	x
			10^{-12}		8	7	2.66	10^{-15}	10^{-14}	10^{-30}	x
32.^{L}	10	20	10^{-8}		3	2	3.16	10^{0}	10^{-14}	10^{-17}	X, REL. F
			10^{-12}		3	2	3.16	10^{0}	10^{-14}	10^{-17}	X, REL. F
33.^{L}	10	20	10^{-8}		3	2	470.	10^{0}	10^{-7}	10^{-6}	REL. F
			10^{-12}		8	2	470.	10^{0}	10^{-7}	10^{-6}	REL. F
34.^{L}	10	20	10^{-8}		3	2	381.	10^{0}	10^{-8}	10^{-6}	REL. F
			10^{-12}		7	3	428.	10^{0}	10^{-9}	10^{-6}	REL. F
35a.	8	8	10^{-8}		40	21	1.65	10^{-1}	10^{-5}	10^{-9}	REL. F
			10^{-12}		53	27	1.65	10^{-1}	10^{-7}	10^{-9}	REL. F
35b.^{0}	9	9	10^{-8}		12	9	1.73	10^{-16}	10^{-15}	10^{-32}	x
			10^{-12}		13	10	1.73	10^{-16}	10^{-15}	10^{-32}	x
35c.	10	10	10^{-8}		25	12	1.81	10^{-1}	10^{-6}	10^{-3}	REL. F
			10^{-12}		34	17	1.81	10^{-1}	10^{-7}	10^{-3}	REL. F
36a.^{0}	4	4	10^{-8}		(4000)	(3985)	27.9	10^{-7}	10^{-7}	10^{-13}	F LIM.
			10^{-12}		(4000)	(3985)	27.9	10^{-7}	10^{-7}	10^{-13}	F LIM.
36b.^{0}	9	9	10^{-8}		(5310)	(5292)	30.8	10^{-7}	10^{-7}	10^{-14}	TIME
			10^{-12}		(5330)	(5312)	30.8	10^{-7}	10^{-7}	10^{-14}	TIME
36c.^{0}	9	9	10^{-8}		29	28	1.73	10^{-17}	10^{-16}	10^{-33}	x
			10^{-12}		40	33	1.73	10^{-17}	10^{-17}	10^{-34}	x
36d.^{0}	9	9	10^{-8}		(9000)	(8982)	39.2	10^{-7}	10^{-7}	10^{-14}	F LIM.
			10^{-12}		(9000)	(8982)	39.2	10^{-7}	10^{-7}	10^{-14}	F LIM.
37.	2	16	10^{-8}		15	14	8.85	10^{1}	10^{-1}	10^{-6}	REL. F
			10^{-12}		21	20	8.85	10^{1}	10^{-3}	10^{-6}	REL. F
38.	3	16	10^{-8}		18	16	26.1	10^{1}	10^{-2}	10^{-6}	REL. F
			10^{-12}		28	26	26.1	10^{1}	10^{-4}	10^{-6}	REL. F

Numerical Results for LMDER

	n	m	TOL	max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
39a.	2	3	10^{-8}		5	4	10^{-6}	10^{-1}	10^{-7}	10^{-7}	REL. F
			10^{-12}		6	5	10^{-7}	10^{-1}	10^{-8}	10^{-7}	REL. F
39b.	2	3	10^{-8}		14	13	10^{-5}	10^{-1}	10^{-5}	10^{-7}	REL. F
			10^{-12}		21	20	10^{-6}	10^{-1}	10^{-7}	10^{-7}	REL. F
39c.	2	3	10^{-8}		18	10	10^{-5}	10^{-1}	10^{-5}	10^{-7}	REL. F
			10^{-12}		25	13	10^{-7}	10^{-1}	10^{-8}	10^{-7}	REL. F
39d.	2	3	10^{-8}		20	13	10^{-5}	10^{-1}	10^{-5}	10^{-7}	REL. F
			10^{-12}		28	18	10^{-7}	10^{-1}	10^{-7}	10^{-7}	REL. F
39e.	2	3	10^{-8}		28	19	10^{-5}	10^{-1}	10^{-4}	10^{-7}	REL. F
			10^{-12}		44	31	10^{-8}	10^{-1}	10^{-7}	10^{-7}	REL. F
39f.	2	3	10^{-8}		31	23	10^{-6}	10^{-1}	10^{-4}	10^{-7}	REL. F
			10^{-12}		44	33	10^{-8}	10^{-1}	10^{-6}	10^{-7}	REL. F
39g.	2	3	10^{-8}		39	29	10^{-8}	10^{-1}	10^{-4}	10^{-7}	REL. F
			10^{-12}		44	31	10^{-9}	10^{-1}	10^{-6}	10^{-7}	REL. F
40a.	3	4	10^{-8}		6	5	10^{-5}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		9	8	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
40b.	3	4	10^{-8}		14	8	10^{-4}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		17	10	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
40c.	3	4	10^{-8}		16	8	10^{-5}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		22	12	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
40d.	3	4	10^{-8}		26	17	10^{-5}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		40	27	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
40e.	3	4	10^{-8}		90	76	10^{-5}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		146	125	10^{-6}	10^{0}	10^{-5}	10^{-7}	REL. F
40f.	3	4	10^{-8}		180	158	10^{-5}	10^{0}	10^{-3}	10^{-7}	REL. F
			10^{-12}		272	241	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F
40g.	3	4	10^{-8}		206	184	10^{-5}	10^{0}	10^{-2}	10^{-7}	REL. F
			10^{-12}		319	287	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F
41a.	5	10	10^{-8}		4	3	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
			10^{-12}		4	3	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
41b.	5	10	10^{-8}		4	3	10^{-6}	10^{0}	10^{-7}	10^{-7}	REL. F
			10^{-12}		5	4	10^{-6}	10^{0}	10^{-9}	10^{-7}	REL. F
41c.	5	10	10^{-8}		6	5	10^{-6}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		8	7	10^{-6}	10^{0}	10^{-7}	10^{-7}	REL. F
41d.	5	10	10^{-8}		15	11	10^{-5}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		22	16	10^{-7}	10^{0}	10^{-6}	10^{-7}	REL. F
41e.	5	10	10^{-8}		29	18	10^{-5}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		38	24	10^{-7}	10^{0}	10^{-6}	10^{-7}	REL. F
41f.	5	10	10^{-8}		57	46	10^{-5}	10^{0}	10^{-3}	10^{-7}	REL. F
			10^{-12}		89	74	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F
41g.	5	10	10^{-8}		84	71	10^{-5}	10^{0}	10^{-3}	10^{-7}	REL. F
			10^{-12}		144	123	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F

Numerical Results for LMDER

	n	m	TOL	max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-8}		18	15	60.8	10^{-13}	10^{-11}	10^{-26}	x
			10^{-12}		19	16	60.8	10^{-13}	10^{-11}	10^{-26}	x
42b.^{0}	4	24	10^{-8}	0.001	48	42	61.3	10^{-13}	10^{-11}	10^{-25}	x
			10^{-12}	0.001	49	43	61.3	10^{-13}	10^{-11}	10^{-25}	x
42c.^{0}	4	24	10^{-8}	0.1	20	17	60.3	10^{-13}	10^{-10}	10^{-26}	x
			10^{-12}	0.1	20	17	60.3	10^{-13}	10^{-10}	10^{-26}	x
42d.^{0}	4	24	10^{-8}	0.1	15	14	60.3	10^{-13}	10^{-11}	10^{-26}	x
			10^{-12}	0.1	16	15	60.3	10^{-13}	10^{-11}	10^{-26}	x
43a.^{0}	5	16	10^{-8}	0.1	14	11	54.0	10^{-14}	10^{-12}	10^{-27}	x
			10^{-12}	0.1	15	12	54.0	10^{-14}	10^{-12}	10^{-28}	x
43b.^{0}	5	16	10^{-8}	0.1	18	15	54.0	10^{-14}	10^{-12}	10^{-28}	x
			10^{-12}	0.1	18	15	54.0	10^{-14}	10^{-12}	10^{-28}	x
43c.^{0}	5	16	10^{-8}	0.1	11	10	54.0	10^{-14}	10^{-11}	10^{-27}	x
			10^{-12}	0.1	11	10	54.0	10^{-14}	10^{-11}	10^{-27}	x
43d.^{0}	5	16	10^{-8}	0.1	22	18	54.0	10^{-14}	10^{-11}	10^{-27}	x
			10^{-12}	0.1	23	19	54.0	10^{-14}	10^{-12}	10^{-28}	x
43e.^{0}	5	16	10^{-8}	0.1	12	11	54.0	10^{-14}	10^{-11}	10^{-27}	x
			10^{-12}	0.1	13	12	54.0	10^{-14}	10^{-12}	10^{-28}	x
43f.^{0}	5	16	10^{-8}		12	9	54.0	10^{-14}	10^{-12}	10^{-27}	x
			10^{-12}		13	10	54.0	10^{-14}	10^{-12}	10^{-27}	x
44a.^{0}	6	6	10^{-8}		37	30	4.06	10^{-15}	10^{-13}	10^{-30}	x
			10^{-12}		38	31	4.06	10^{-15}	10^{-13}	10^{-30}	x
44b.^{0}	6	6	10^{-8}		5	4	3.52	10^{-15}	10^{-13}	10^{-29}	x
			10^{-12}		6	5	3.52	10^{-15}	10^{-13}	10^{-29}	x
44c.^{0}	6	6	10^{-8}		108	98	20.6	10^{-14}	10^{-11}	10^{-29}	x
			10^{-12}		109	99	20.6	10^{-15}	10^{-11}	10^{-30}	x
44d.^{0}	6	6	10^{-8}		98	88	15.3	10^{-15}	10^{-11}	10^{-29}	x
			10^{-12}		99	89	15.3	10^{-15}	10^{-11}	10^{-29}	x
44e.^{0}	6	6	10^{-8}		82	71	9.27	10^{-14}	10^{-11}	10^{-28}	x
			10^{-12}		83	72	9.27	10^{-14}	10^{-11}	10^{-28}	x
45a.^{0}	8	8	10^{-8}		47	35	4.06	10^{-16}	10^{-15}	10^{-29}	x
			10^{-12}		48	36	4.06	10^{-16}	10^{-15}	10^{-29}	x
45b.^{0}	8	8	10^{-8}		5	4	3.56	10^{-15}	10^{-13}	10^{-29}	x
			10^{-12}		6	5	3.56	10^{-15}	10^{-13}	10^{-29}	x
45c.^{0}	8	8	10^{-8}		164	148	20.6	10^{-14}	10^{-11}	10^{-28}	x
			10^{-12}		165	149	20.6	10^{-15}	10^{-11}	10^{-29}	x
45d.^{0}	8	8	10^{-8}		144	133	15.3	10^{-15}	10^{-13}	10^{-30}	x
			10^{-12}		145	134	15.3	10^{-16}	10^{-13}	10^{-31}	x
45e.^{0}	8	8	10^{-8}		130	119	9.31	10^{-14}	10^{-12}	10^{-29}	x
			10^{-12}		131	120	9.31	10^{-14}	10^{-12}	10^{-29}	x

5.6.2 Corrected Gauss-Newton Methods (NPL/NAG LSQSDN and LSQFDQ)

5.6.2.1 Software and Algorithms

The results were obtained using subroutines LSQSDN and LSQFDQ implementing corrected Gauss-Newton methods from the National Physical Laboratory, which are available at Stanford Linear Accelerator Center. A subproblem of the form

$$\min_{p \in \Re^n} \bar{g}_k^T p + \frac{1}{2} p^T (J_k^T J_k + B_k) p$$

$$\text{subject to} \quad J_k p \approx -f_k,$$

is solved for a search direction p_k, where $\approx$ is interpreted in a least-squares sense using the singular-value decomposition (see Chapters 3 and 4). Subroutine LSQSDN requires exact second derivatives for the term B_k that involves the second derivatives of the residuals, while LSQFDQ uses a quasi-Newton approximation. The linesearch algorithm used within the subroutines requires both function and gradient information (see Gill and Murray [1974], for details). These subroutines are similar to those available in the NAG Library [1984] for solving nonlinear least-squares problems : LSQSDN corresponds to NAG subroutine E04HEF and LSQFDQ to NAG subroutine E04GBF.

5.6.2.2 Parmeters

LSQSDN and LSQFDQ have the same set of input parameters as the corresponding software from the NAG Library [1984]. The values chosen are listed below.

MAXCAL	-	$\min\{9999, 1000 * n\}$	function evaluation limit
XTOL	-	varied; see tables	accuracy in x
ETA	-	0.5	linesearch accuracy
STEPMX	-	usually 10^6 (default) †	maximum step for linesearch

† In some cases the default STEPMX $= 10^6$ was too large and overflow occurred during function evaluation in the linesearch. These cases are indicated in the table by giving the lower value of STEPMX that was subsequently used to obtain the results.

See the NAG [1984] manual for details concerning the parameters.

5.6.2.3 Convergence Criteria

The following quantities will be used in describing the convergence criteria :

objective function	:	$\mathcal{F}_k = f_k^T f_k$
objective gradient	:	$\bar{g}_k = \nabla \mathcal{F}_k = 2J_k^T f_k$
search direction	:	p_k, the minimizer of the subproblem
steplength	:	α_k, determined by the linesearch

An iterate is determined to be optimal by `LSQSDN` and `LSQFDQ` if

$$\mathcal{F}_k < \epsilon_M^2 \tag{5.6.8}$$

or

$$\|g_k\|_2 < \epsilon_M \|f_k\|_2 \tag{5.6.9}$$

or if the following three conditions hold :

$$\alpha_k \|p_k\|_2 < (\texttt{XTOL} + \epsilon_M)(1 + \|x_k\|_2) \tag{5.6.10}$$

and

$$\mathcal{F}(x_{k-1}) - \mathcal{F}_k < (\texttt{XTOL} + \epsilon_M)^2(1 + |\mathcal{F}_k|) \tag{5.6.11}$$

and

$$\|g_k\|_2 < \epsilon_M^{1/3}(1 + |\mathcal{F}_k|). \tag{5.6.12}$$

Conditions (5.6.10) and (5.6.11) are meant to ensure that the sequence x_k has converged, while conditions (5.6.9) and (5.6.12) are intended to test whether the necessary condition that the gradient vanish at a minimum is approximately satisfied at x_k. Condition (5.6.9) allows the algorithm to accept a point as a local mimimum if a more restrictive test on the necessary condition than (5.6.12) is met, even if conditions (5.6.10) and (5.6.11) do not hold. For the zero-residual case, condition (5.6.8) specifies that the method may also terminate when $\|f_k\|_2$ is no larger than the relative machine precision. For a detailed discussion of convergence criteria similar to these, see Sections 8.2 and 8.5 of Gill, Murray, and Wright [1981]. In particular, Section 8.5.1.3 treats special considerations relevant to nonlinear least squares.

The following abbreviations are used in the tables to describe the conditions under which the algorithm terminates :

OPT.	-	optimal point found
*	-	current point cannot be improved †
F LIM.	-	function evalutaion limit reached
TIME	-	time limit exceeded

† A '*' corresponds to the situation in which the algorithm terminates due to failure in the linesearch to find an acceptable step at the current iteration.

Numerical Results for LSQSDN

	n	m		max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-8}		34	13	1.41	0.00	0.00	0.00	OPT.
			10^{-12}		34	13	1.41	0.00	0.00	0.00	OPT.
2.^{0}	2	2	10^{-8}		35	8	11.4	10^{1}	10^{-6}	10^{1}	*
			10^{-12}		35	8	11.4	10^{1}	10^{-6}	10^{1}	*
3.^{0}	2	2	10^{-8}		45	24	9.11	10^{-13}	10^{-8}	10^{-26}	OPT.
			10^{-12}		45	24	9.11	10^{-13}	10^{-8}	10^{-26}	OPT.
4.^{0}	2	3	10^{-8}		59	21	10^{6}	10^{-10}	10^{-4}	10^{-19}	*
			10^{-12}		59	21	10^{6}	10^{-10}	10^{-4}	10^{-19}	*
5.^{0}	2	3	10^{-8}		10	7	3.04	10^{-14}	10^{-13}	10^{-28}	OPT.
			10^{-12}		10	7	3.04	10^{-14}	10^{-13}	10^{-28}	OPT.
6.	2	10	10^{-8}	5.0	36	10	.365	10^{1}	10^{-5}	10^{-6}	*
			10^{-12}	5.0	36	10	.365	10^{1}	10^{-5}	10^{-6}	*
7.^{0}	3	3	10^{-8}		14	10	1.00	10^{-11}	10^{-10}	10^{-23}	OPT.
			10^{-12}		14	10	1.00	10^{-11}	10^{-10}	10^{-23}	OPT.
8.	3	15	10^{-8}		6	5	2.60	10^{-1}	10^{-9}	10^{-8}	OPT.
			10^{-12}		6	5	2.60	10^{-1}	10^{-9}	10^{-8}	OPT.
9.	3	15	10^{-8}		3	2	1.08	10^{-4}	10^{-12}	10^{-14}	OPT.
			10^{-12}		3	2	1.08	10^{-4}	10^{-12}	10^{-14}	OPT.
10.	3	16	10^{-8}		17	10	10^{4}	10^{1}	10^{2}	10^{-6}	*
			10^{-12}		17	10	10^{4}	10^{1}	10^{2}	10^{-6}	*
11.^{0}	3	10	10^{-8}		(3001)	(1528)	118.	10^{-2}	10^{1}	10^{-5}	F LIM.
			10^{-12}		(3001)	(1528)	118.	10^{-2}	10^{1}	10^{-5}	F LIM.
12.^{0}	3	10	10^{-8}		8	6	10.1	10^{-10}	10^{-10}	10^{-19}	OPT.
			10^{-12}		12	8	10.1	10^{-10}	10^{-10}	10^{-19}	*
13.^{0}	4	4	10^{-8}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
			10^{-12}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
14.^{0}	4	6	10^{-8}		93	42	2.00	10^{-7}	10^{-6}	10^{-14}	OPT.
			10^{-12}		99	45	2.00	10^{-7}	10^{-6}	10^{-14}	*
15.	4	11	10^{-8}		19	8	.328	10^{-2}	10^{-16}	10^{-9}	OPT.
			10^{-12}		19	8	.328	10^{-2}	10^{-16}	10^{-9}	OPT.
16.	4	20	10^{-8}		33	17	17.6	10^{2}	10^{-5}	10^{-8}	*
			10^{-12}		33	17	17.6	10^{2}	10^{-5}	10^{-8}	*
17.	5	33	10^{-8}		14	9	2.46	10^{-2}	10^{-9}	10^{-11}	OPT.
			10^{-12}		18	11	2.46	10^{-2}	10^{-9}	10^{-11}	*
18.^{0}	6	13	10^{-8}	10.0	115	30	12.3	10^{-13}	10^{-13}	10^{-27}	OPT.
			10^{-12}	10.0	115	30	12.3	10^{-13}	10^{-13}	10^{-27}	OPT.
19.	11	65	10^{-8}		19	10	9.38	10^{-1}	10^{-9}	10^{-8}	OPT.
			10^{-12}		19	10	9.38	10^{-1}	10^{-9}	10^{-8}	OPT.

Numerical Results for LSQSDN

	n	m		max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|g^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-8}		9	7	2.44	10^{-2}	10^{-8}	10^{-10}	*
			10^{-12}		9	7	2.44	10^{-2}	10^{-8}	10^{-10}	*
20b.	9	31	10^{-8}		6	5	6.06	10^{-3}	10^{-11}	10^{-13}	OPT.
			10^{-12}		6	5	6.06	10^{-3}	10^{-11}	10^{-13}	OPT.
20c.	12	31	10^{-8}		6	5	16.6	10^{-5}	10^{-13}	10^{-16}	OPT.
			10^{-12}		6	5	16.6	10^{-5}	10^{-13}	10^{-16}	OPT.
20d.	20	31	10^{-8}		11	7	247.	10^{-10}	10^{-11}	10^{-24}	OPT.
			10^{-12}		13	8	247.	10^{-10}	10^{-11}	10^{-24}	OPT.
21a.^{0}	10	10	10^{-8}		34	13	3.16	0.00	0.00	0.00	OPT.
			10^{-12}		34	13	3.16	0.00	0.00	0.00	OPT.
21b.^{0}	20	20	10^{-8}		34	13	4.47	0.00	0.00	0.00	OPT.
			10^{-12}		34	13	4.47	0.00	0.00	0.00	OPT.
22a.^{0}	12	12	10^{-8}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
			10^{-12}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
22b.^{0}	20	20	10^{-8}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
			10^{-12}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
23a.	4	5	10^{-8}		47	27	.500	10^{-3}	10^{-10}	10^{-10}	OPT.
			10^{-12}		47	27	.500	10^{-3}	10^{-10}	10^{-10}	OPT.
23b.	10	11	10^{-8}		73	34	.500	10^{-2}	10^{-14}	10^{-11}	OPT.
			10^{-12}		73	34	.500	10^{-2}	10^{-14}	10^{-11}	OPT.
24a.	4	8	10^{-8}		176	106	.759	10^{-3}	10^{-12}	10^{-11}	OPT.
			10^{-12}		176	106	.759	10^{-3}	10^{-12}	10^{-11}	OPT.
24b.	10	20	10^{-8}		153	89	.598	10^{-2}	10^{-13}	10^{-9}	OPT.
			10^{-12}		153	89	.598	10^{-2}	10^{-13}	10^{-9}	OPT.
25a.^{0}	10	12	10^{-8}		12	10	3.16	10^{-8}	10^{-6}	10^{-15}	OPT.
			10^{-12}		17	12	3.16	10^{-8}	10^{-6}	10^{-15}	OPT.
25b.^{0}	20	22	10^{-8}		14	12	4.47	10^{-8}	10^{-7}	10^{-17}	OPT.
			10^{-12}		19	14	4.47	10^{-8}	10^{-7}	10^{-17}	*
26a.^{0}	10	10	10^{-8}		18	9	.306	10^{-11}	10^{-11}	10^{-22}	OPT.
			10^{-12}		22	11	.306	10^{-11}	10^{-11}	10^{-22}	*
26b.^{0}	20	20	10^{-8}		25	11	.222	10^{-11}	10^{-11}	10^{-22}	OPT.
			10^{-12}		29	13	.222	10^{-11}	10^{-11}	10^{-22}	*
27a.^{0}	10	10	10^{-8}	10^2	21	7	3.18	10^{-14}	10^{-13}	10^{-27}	OPT.
			10^{-12}	10^2	21	7	3.18	10^{-14}	10^{-13}	10^{-27}	OPT.
27b.^{0}	20	20	10^{-8}	10^2	24	7	4.47	10^{-12}	10^{-12}	10^{-24}	OPT.
			10^{-12}	10^2	24	7	4.47	10^{-12}	10^{-12}	10^{-24}	OPT.
28a.^{0}	10	10	10^{-8}		4	3	.412	10^{-15}	10^{-16}	10^{-31}	OPT.
			10^{-12}		4	3	.412	10^{-15}	10^{-16}	10^{-31}	OPT.
28b.^{0}	20	20	10^{-8}		4	3	.571	10^{-16}	10^{-16}	10^{-32}	OPT.
			10^{-12}		4	3	.571	10^{-16}	10^{-16}	10^{-32}	OPT.

Numerical Results for LSQSDN

	n	m		max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
29a.$^{0,-}$	10	10	10^{-8}		6	4	.412	10^{-14}	10^{-14}	10^{-29}	OPT.
			10^{-12}		6	4	.412	10^{-14}	10^{-14}	10^{-29}	OPT.
29b.^{0}	20	20	10^{-8}		6	4	.571	10^{-14}	10^{-14}	10^{-28}	OPT.
			10^{-12}		6	4	.571	10^{-14}	10^{-14}	10^{-28}	OPT.
30a.^{0}	10	10	10^{-8}		7	5	2.05	10^{-9}	10^{-9}	10^{-18}	OPT.
			10^{-12}		11	7	2.05	10^{-9}	10^{-9}	10^{-18}	*
30b.^{0}	20	20	10^{-8}		7	5	3.04	10^{-9}	10^{-9}	10^{-18}	OPT.
			10^{-12}		11	7	3.04	10^{-9}	10^{-9}	10^{-18}	*
31a.^{0}	10	10	10^{-8}		8	6	1.80	10^{-8}	10^{-7}	10^{-16}	OPT.
			10^{-12}		12	8	1.80	10^{-8}	10^{-7}	10^{-16}	*
31b.^{0}	20	20	10^{-8}		8	6	2.66	10^{-8}	10^{-7}	10^{-16}	OPT.
			10^{-12}		12	8	2.66	10^{-8}	10^{-7}	10^{-16}	*
32.^{L}	10	20	10^{-8}		2	1	3.16	10^{0}	10^{-14}	10^{-17}	OPT.
			10^{-12}		2	1	3.16	10^{0}	10^{-14}	10^{-17}	OPT.
33.^{L}	10	20	10^{-8}		2	1	1.46	10^{0}	10^{-9}	10^{-6}	OPT.
			10^{-12}		2	1	1.46	10^{0}	10^{-9}	10^{-6}	OPT.
34.^{L}	10	20	10^{-8}		2	1	1.78	10^{0}	10^{-9}	10^{-6}	OPT.
			10^{-12}		2	1	1.78	10^{0}	10^{-9}	10^{-6}	OPT.
35a.	8	8	10^{-8}		99	24	1.65	10^{-1}	10^{-7}	10^{-9}	*
			10^{-12}		99	24	1.65	10^{-1}	10^{-7}	10^{-9}	*
35b.^{0}	9	9	10^{-8}		38	12	1.73	10^{-10}	10^{-9}	10^{-19}	OPT.
			10^{-12}		42	14	1.73	10^{-10}	10^{-9}	10^{-19}	*
35c.	10	10	10^{-8}	10.0	58	11	1.81	10^{-1}	10^{-10}	10^{-3}	OPT.
			10^{-12}	10.0	58	11	1.81	10^{-1}	10^{-10}	10^{-3}	OPT.
36a.^{0}	4	4	10^{-8}		96	39	50.0	10^{-11}	10^{-9}	10^{-22}	OPT.
			10^{-12}		102	42	50.0	10^{-11}	10^{-9}	10^{-22}	*
36b.^{0}	9	9	10^{-8}		93	38	50.0	10^{-10}	10^{-8}	10^{-19}	OPT.
			10^{-12}		97	40	50.0	10^{-10}	10^{-8}	10^{-19}	OPT.
36c.^{0}	9	9	10^{-8}		28	27	1.73	10^{-17}	10^{-16}	10^{-33}	OPT.
			10^{-12}		28	27	1.73	10^{-17}	10^{-16}	10^{-33}	OPT.
36d.^{0}	9	9	10^{-8}		834	201	233.	10^{-10}	10^{-8}	10^{-20}	OPT.
			10^{-12}		836	202	233.	10^{-10}	10^{-8}	10^{-20}	OPT.
37.	2	16	10^{-8}		10	7	8.85	10^{1}	10^{-6}	10^{-6}	*
			10^{-12}		10	7	8.85	10^{1}	10^{-6}	10^{-6}	*
38.	3	16	10^{-8}		13	9	26.1	10^{1}	10^{-8}	10^{-6}	*
			10^{-12}		13	9	26.1	10^{1}	10^{-8}	10^{-6}	*

Numerical Results for LSQSDN

	n	m		max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
39a.	2	3	10^{-8}		4	3	10^{-6}	10^{-1}	10^{-9}	10^{-7}	OPT.
			10^{-12}		4	3	10^{-6}	10^{-1}	10^{-9}	10^{-7}	OPT.
39b.	2	3	10^{-8}		6	5	10^{-7}	10^{-1}	10^{-15}	10^{-7}	OPT.
			10^{-12}		6	5	10^{-7}	10^{-1}	10^{-15}	10^{-7}	OPT.
39c.	2	3	10^{-8}		9	5	10^{-7}	10^{-1}	10^{-10}	10^{-7}	OPT.
			10^{-12}		9	5	10^{-7}	10^{-1}	10^{-10}	10^{-7}	OPT.
39d.	2	3	10^{-8}		12	7	10^{-7}	10^{-1}	10^{-10}	10^{-7}	OPT.
			10^{-12}		12	7	10^{-7}	10^{-1}	10^{-10}	10^{-7}	OPT.
39e.	2	3	10^{-8}		12	7	10^{-8}	10^{-1}	10^{-10}	10^{-7}	OPT.
			10^{-12}		12	7	10^{-8}	10^{-1}	10^{-10}	10^{-7}	OPT.
39f.	2	3	10^{-8}		24	10	10^{-9}	10^{-1}	10^{-10}	10^{-7}	OPT.
			10^{-12}		24	10	10^{-9}	10^{-1}	10^{-10}	10^{-7}	OPT.
39g.	2	3	10^{-8}		39	15	10^{-10}	10^{-1}	10^{-7}	10^{-7}	*
			10^{-12}		39	15	10^{-10}	10^{-1}	10^{-7}	10^{-7}	*
40a.	3	4	10^{-8}		5	4	10^{-6}	10^{0}	10^{-15}	10^{-7}	OPT.
			10^{-12}		5	4	10^{-6}	10^{0}	10^{-15}	10^{-7}	OPT.
40b.	3	4	10^{-8}		6	4	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
			10^{-12}		6	4	10^{-6}	10^{0}	10^{-10}	10^{-7}	OPT.
40c.	3	4	10^{-8}		11	6	10^{-7}	10^{0}	10^{-14}	10^{-7}	OPT.
			10^{-12}		11	6	10^{-7}	10^{0}	10^{-14}	10^{-7}	OPT.
40d.	3	4	10^{-8}		13	6	10^{-7}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-12}		13	6	10^{-7}	10^{0}	10^{-12}	10^{-7}	OPT.
40e.	3	4	10^{-8}		45	16	10^{-7}	10^{0}	10^{-13}	10^{-7}	OPT.
			10^{-12}		45	16	10^{-7}	10^{0}	10^{-13}	10^{-7}	OPT.
40f.	3	4	10^{-8}		79	24	10^{-8}	10^{0}	10^{-14}	10^{-7}	OPT.
			10^{-12}		79	24	10^{-8}	10^{0}	10^{-14}	10^{-7}	OPT.
40g.	3	4	10^{-8}		87	27	10^{-9}	10^{0}	10^{-11}	10^{-7}	OPT.
			10^{-12}		87	27	10^{-9}	10^{0}	10^{-11}	10^{-7}	OPT.
41a.	5	10	10^{-8}		4	3	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-12}		4	3	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
41b.	5	10	10^{-8}		4	3	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		4	3	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
41c.	5	10	10^{-8}		5	4	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		5	4	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
41d.	5	10	10^{-8}		9	7	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-12}		9	7	10^{-6}	10^{0}	10^{-12}	10^{-7}	OPT.
41e.	5	10	10^{-8}		14	10	10^{-7}	10^{0}	10^{-11}	10^{-7}	OPT.
			10^{-12}		14	10	10^{-7}	10^{0}	10^{-11}	10^{-7}	OPT.
41f.	5	10	10^{-8}		16	12	10^{-7}	10^{0}	10^{-10}	10^{-7}	OPT.
			10^{-12}		16	12	10^{-7}	10^{0}	10^{-10}	10^{-7}	OPT.
41g.	5	10	10^{-8}		21	15	10^{-8}	10^{0}	10^{-12}	10^{-7}	OPT.
			10^{-12}		21	15	10^{-8}	10^{0}	10^{-12}	10^{-7}	OPT.

Numerical Results for LSQSDN

	n	m		max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
42a.[0]	4	24	10^{-8}	5.0	67	41	63.5	10^{-6}	10^{-3}	10^{-11}	*
			10^{-12}	5.0	67	41	63.5	10^{-6}	10^{-3}	10^{-11}	*
42b.[0]	4	24	10^{-8}	2.0	76	67	61.9	10^{-7}	10^{-4}	10^{-13}	*
			10^{-12}	2.0	76	67	61.9	10^{-7}	10^{-4}	10^{-13}	*
42c.[0]	4	24	10^{-8}	5.0	49	28	61.8	10^{-11}	10^{-8}	10^{-22}	OPT.
			10^{-12}	5.0	49	28	61.8	10^{-11}	10^{-8}	10^{-22}	OPT.
42d.[0]	4	24	10^{-8}	5.0	27	19	60.3	10^{-5}	10^{-3}	10^{-10}	*
			10^{-12}	5.0	27	19	60.3	10^{-5}	10^{-3}	10^{-10}	*
43a.[0]	5	16	10^{-8}	5.0	29	16	54.0	10^{-9}	10^{-7}	10^{-18}	OPT.
			10^{-12}	5.0	37	20	54.0	10^{-9}	10^{-7}	10^{-18}	*
43b.[0]	5	16	10^{-8}	10.0	32	18	54.0	10^{-7}	10^{-5}	10^{-14}	*
			10^{-12}	10.0	32	18	54.0	10^{-7}	10^{-5}	10^{-14}	*
43c.[0]	5	16	10^{-8}	10.0	25	14	54.0	10^{-8}	10^{-6}	10^{-16}	OPT.
			10^{-12}	10.0	33	18	54.0	10^{-8}	10^{-6}	10^{-16}	*
43d.[0]	5	16	10^{-8}	10.0	30	16	54.0	10^{-9}	10^{-7}	10^{-18}	OPT.
			10^{-12}	10.0	38	20	54.0	10^{-9}	10^{-7}	10^{-18}	*
43e.[0]	5	16	10^{-8}	10.0	28	14	54.0	10^{-8}	10^{-6}	10^{-15}	*
			10^{-12}	10.0	28	14	54.0	10^{-8}	10^{-6}	10^{-15}	*
43f.[0]	5	16	10^{-8}		39	17	54.0	10^{-10}	10^{-7}	10^{-19}	OPT.
			10^{-12}		47	21	54.0	10^{-10}	10^{-7}	10^{-19}	*
44a.[0]	6	6	10^{-8}		100	27	4.06	10^{-14}	10^{-12}	10^{-27}	OPT.
			10^{-12}		100	27	4.06	10^{-14}	10^{-12}	10^{-27}	OPT.
44b.[0]	6	6	10^{-8}		6	4	3.52	10^{-8}	10^{-6}	10^{-15}	OPT.
			10^{-12}		10	6	3.52	10^{-8}	10^{-6}	10^{-15}	*
44c.[0]	6	6	10^{-8}		59	20	20.6	10^{-12}	10^{-8}	10^{-23}	OPT.
			10^{-12}		59	20	20.6	10^{-12}	10^{-8}	10^{-23}	OPT.
44d.[0]	6	6	10^{-8}		41	17	15.3	10^{-7}	10^{-3}	10^{-14}	*
			10^{-12}		41	17	15.3	10^{-7}	10^{-3}	10^{-14}	*
44e.[0]	6	6	10^{-8}		79	26	9.27	10^{-10}	10^{-7}	10^{-20}	OPT.
			10^{-12}		83	28	9.27	10^{-10}	10^{-7}	10^{-20}	*
45a.[0]	8	8	10^{-8}		100	27	4.06	10^{-14}	10^{-12}	10^{-27}	OPT.
			10^{-12}		100	27	4.06	10^{-14}	10^{-12}	10^{-27}	OPT.
45b.[0]	8	8	10^{-8}		6	4	3.56	10^{-8}	10^{-6}	10^{-15}	OPT.
			10^{-12}		12	7	3.56	10^{-8}	10^{-6}	10^{-15}	*
45c.[0]	8	8	10^{-8}		59	20	20.6	10^{-12}	10^{-8}	10^{-23}	OPT.
			10^{-12}		59	20	20.6	10^{-12}	10^{-8}	10^{-23}	OPT.
45d.[0]	8	8	10^{-8}		43	18	15.3	10^{-7}	10^{-3}	10^{-13}	*
			10^{-12}		43	18	15.3	10^{-7}	10^{-3}	10^{-13}	*
45e.[0]	8	8	10^{-8}		79	26	9.31	10^{-10}	10^{-7}	10^{-20}	OPT.
			10^{-12}		81	27	9.31	10^{-10}	10^{-7}	10^{-20}	*

Numerical Results for LSQFDQ

	n	m		max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-8}		34	13	1.41	0.00	0.00	0.00	OPT.
			10^{-12}		34	13	1.41	0.00	0.00	0.00	OPT.
2.^{0}	2	2	10^{-8}		44	17	11.4	10^{1}	10^{-7}	10^{1}	*
			10^{-12}		44	17	11.4	10^{1}	10^{-7}	10^{1}	*
3.^{0}	2	2	10^{-8}		49	26	9.11	10^{-13}	10^{-8}	10^{-26}	*
			10^{-12}		49	26	9.11	10^{-13}	10^{-8}	10^{-26}	*
4.^{0}	2	3	10^{-8}		65	21	10^{6}	10^{-9}	10^{-3}	10^{-19}	*
			10^{-12}		65	21	10^{6}	10^{-9}	10^{-3}	10^{-19}	*
5.^{0}	2	3	10^{-8}		14	9	3.04	10^{-14}	10^{-13}	10^{-28}	*
			10^{-12}		14	9	3.04	10^{-14}	10^{-13}	10^{-28}	*
6.	2	10	10^{-8}		52	24	.365	10^{1}	10^{-5}	10^{-6}	*
			10^{-12}		52	24	.365	10^{1}	10^{-5}	10^{-6}	*
7.^{0}	3	3	10^{-8}		20	13	1.00	10^{-11}	10^{-10}	10^{-23}	*
			10^{-12}		20	13	1.00	10^{-11}	10^{-10}	10^{-23}	*
8.	3	15	10^{-8}		14	12	2.60	10^{-1}	10^{-9}	10^{-8}	*
			10^{-12}		14	12	2.60	10^{-1}	10^{-9}	10^{-8}	*
9.	3	15	10^{-8}		3	2	1.08	10^{-4}	10^{-12}	10^{-14}	OPT.
			10^{-12}		3	2	1.08	10^{-4}	10^{-12}	10^{-14}	OPT.
10.	3	16	10^{-8}		18	10	10^{4}	10^{1}	10^{3}	10^{-6}	*
			10^{-12}		18	10	10^{4}	10^{1}	10^{3}	10^{-6}	*
11.^{0}	3	10	10^{-8}		61	16	60.8	10^{-2}	10^{-8}	10^{-4}	*
			10^{-12}		61	16	60.8	10^{-2}	10^{-8}	10^{-4}	*
12.^{0}	3	10	10^{-8}		12	8	10.1	10^{-10}	10^{-10}	10^{-19}	*
			10^{-12}		12	8	10.1	10^{-10}	10^{-10}	10^{-19}	*
13.^{0}	4	4	10^{-8}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
			10^{-12}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
14.^{0}	4	6	10^{-8}		114	48	2.00	10^{-10}	10^{-8}	10^{-19}	*
			10^{-12}		114	48	2.00	10^{-10}	10^{-8}	10^{-19}	*
15.	4	11	10^{-8}		35	22	.328	10^{-2}	10^{-9}	10^{-9}	OPT.
			10^{-12}		36	23	.328	10^{-2}	10^{-10}	10^{-9}	OPT.
16.	4	20	10^{-8}		44	25	17.6	10^{2}	10^{-3}	10^{-8}	*
			10^{-12}		44	25	17.6	10^{2}	10^{-3}	10^{-8}	*
17.	5	33	10^{-8}		19	11	2.46	10^{-2}	10^{-7}	10^{-11}	*
			10^{-12}		19	11	2.46	10^{-2}	10^{-7}	10^{-11}	*
18.^{0}	6	13	10^{-8}	10.0	75	24	12.3	10^{-11}	10^{-11}	10^{-22}	*
			10^{-12}	10.0	75	24	12.3	10^{-11}	10^{-11}	10^{-22}	*
19.	11	65	10^{-8}		34	22	9.38	10^{-1}	10^{-8}	10^{-8}	*
			10^{-12}		34	22	9.38	10^{-1}	10^{-8}	10^{-8}	*

Numerical Results for LSQFDQ

	n	m		max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
20a.	6	31	10^{-8}		31	22	2.44	10^{-2}	10^{-8}	10^{-10}	*
			10^{-12}		31	22	2.44	10^{-2}	10^{-8}	10^{-10}	*
20b.	9	31	10^{-8}		6	5	6.06	10^{-3}	10^{-11}	10^{-13}	OPT.
			10^{-12}		6	5	6.06	10^{-3}	10^{-11}	10^{-13}	OPT.
20c.	12	31	10^{-8}		6	5	16.6	10^{-5}	10^{-13}	10^{-16}	OPT.
			10^{-12}		6	5	16.6	10^{-5}	10^{-13}	10^{-16}	OPT.
20d.	20	31	10^{-8}		19	11	247.	10^{-10}	10^{-11}	10^{-24}	*
			10^{-12}		19	11	247.	10^{-10}	10^{-11}	10^{-24}	*
21a.^{0}	10	10	10^{-8}		34	13	3.16	0.00	0.00	0.00	OPT.
			10^{-12}		34	13	3.16	0.00	0.00	0.00	OPT.
21b.^{0}	20	20	10^{-8}		34	13	4.47	0.00	0.00	0.00	OPT.
			10^{-12}		34	13	4.47	0.00	0.00	0.00	OPT.
22a.^{0}	12	12	10^{-8}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
			10^{-12}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
22b.^{0}	20	20	10^{-8}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
			10^{-12}		18	17	10^{-5}	10^{-9}	10^{-13}	10^{-18}	OPT.
23a.	4	5	10^{-8}		67	39	.500	10^{-3}	10^{-12}	10^{-10}	OPT.
			10^{-12}		67	39	.500	10^{-3}	10^{-12}	10^{-10}	OPT.
23b.	10	11	10^{-8}		88	42	.500	10^{-2}	10^{-10}	10^{-11}	OPT.
			10^{-12}		88	42	.500	10^{-2}	10^{-10}	10^{-11}	OPT.
24a.	4	8	10^{-8}		598	290	.759	10^{-3}	10^{-10}	10^{-11}	OPT.
			10^{-12}		598	290	.759	10^{-3}	10^{-10}	10^{-11}	OPT.
24b.	10	20	10^{-8}	10^2	533	231	.598	10^{-2}	10^{-9}	10^{-9}	*
			10^{-12}	10^2	533	231	.598	10^{-2}	10^{-9}	10^{-9}	*
25a.^{0}	10	12	10^{-8}		16	12	3.16	10^{-8}	10^{-6}	10^{-15}	*
			10^{-12}		16	12	3.16	10^{-8}	10^{-6}	10^{-15}	*
25b.^{0}	20	22	10^{-8}		18	14	4.47	10^{-8}	10^{-7}	10^{-17}	*
			10^{-12}		18	14	4.47	10^{-8}	10^{-7}	10^{-17}	*
26a.^{0}	10	10	10^{-8}		22	11	.306	10^{-11}	10^{-11}	10^{-22}	*
			10^{-12}		22	11	.306	10^{-11}	10^{-11}	10^{-22}	*
26b.^{0}	20	20	10^{-8}		29	13	.222	10^{-11}	10^{-11}	10^{-22}	*
			10^{-12}		29	13	.222	10^{-11}	10^{-11}	10^{-22}	*
27a.^{0}	10	10	10^{-8}	10^2	26	10	3.16	10^{-12}	10^{-12}	10^{-24}	*
			10^{-12}	10^2	26	10	3.16	10^{-12}	10^{-12}	10^{-24}	*
27b.^{0}	20	20	10^{-8}	10^2	30	11	4.47	10^{-11}	10^{-10}	10^{-22}	*
			10^{-12}	10^2	30	11	4.47	10^{-11}	10^{-10}	10^{-22}	*
28a.^{0}	10	10	10^{-8}		4	3	.412	10^{-15}	10^{-16}	10^{-31}	OPT.
			10^{-12}		4	3	.412	10^{-15}	10^{-16}	10^{-31}	OPT.
28b.^{0}	20	20	10^{-8}		4	3	.571	10^{-16}	10^{-16}	10^{-32}	OPT.
			10^{-12}		4	3	.571	10^{-16}	10^{-16}	10^{-32}	OPT.

Numerical Results for LSQFDQ

	n	m		max. step	f, J evals.	iters.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
29a.$^{0,-}$	10	10	10^{-8}		10	6	.412	10^{-14}	10^{-14}	10^{-29}	*
			10^{-12}		10	6	.412	10^{-14}	10^{-14}	10^{-29}	*
29b.^{0}	20	20	10^{-8}		10	6	.571	10^{-14}	10^{-14}	10^{-28}	*
			10^{-12}		10	6	.571	10^{-14}	10^{-14}	10^{-28}	*
30a.^{0}	10	10	10^{-8}		11	7	2.05	10^{-9}	10^{-9}	10^{-18}	*
			10^{-12}		11	7	2.05	10^{-9}	10^{-9}	10^{-18}	*
30b.^{0}	20	20	10^{-8}		11	7	3.04	10^{-9}	10^{-9}	10^{-18}	*
			10^{-12}		11	7	3.04	10^{-9}	10^{-9}	10^{-18}	*
31a.^{0}	10	10	10^{-8}		12	8	1.80	10^{-8}	10^{-7}	10^{-16}	*
			10^{-12}		12	8	1.80	10^{-8}	10^{-7}	10^{-16}	*
31b.^{0}	20	20	10^{-8}		12	8	2.66	10^{-8}	10^{-7}	10^{-16}	*
			10^{-12}		12	8	2.66	10^{-8}	10^{-7}	10^{-16}	*
32.^{L}	10	20	10^{-8}		2	1	3.16	10^{0}	10^{-14}	10^{-17}	OPT.
			10^{-12}		2	1	3.16	10^{0}	10^{-14}	10^{-17}	OPT.
33.^{L}	10	20	10^{-8}		2	1	1.46	10^{0}	10^{-9}	10^{-6}	OPT.
			10^{-12}		2	1	1.46	10^{0}	10^{-9}	10^{-6}	OPT.
34.^{L}	10	20	10^{-8}		2	1	1.78	10^{0}	10^{-9}	10^{-6}	OPT.
			10^{-12}		2	1	1.78	10^{0}	10^{-9}	10^{-6}	OPT.
35a.	8	8	10^{-8}	10.0	87	31	1.65	10^{-1}	10^{-8}	10^{-9}	*
			10^{-12}	10.0	87	31	1.65	10^{-1}	10^{-8}	10^{-9}	*
35b.^{0}	9	9	10^{-8}		42	14	1.73	10^{-9}	10^{-9}	10^{-18}	*
			10^{-12}		42	14	1.73	10^{-9}	10^{-9}	10^{-18}	*
35c.	10	10	10^{-8}		85	25	1.81	10^{-1}	10^{-8}	10^{-3}	*
			10^{-12}		85	25	1.81	10^{-1}	10^{-8}	10^{-3}	*
36a.^{0}	4	4	10^{-8}		102	42	50.0	10^{-11}	10^{-9}	10^{-22}	*
			10^{-12}		102	42	50.0	10^{-11}	10^{-9}	10^{-22}	*
36b.^{0}	9	9	10^{-8}		99	41	50.0	10^{-10}	10^{-8}	10^{-19}	*
			10^{-12}		99	41	50.0	10^{-10}	10^{-8}	10^{-19}	*
36c.^{0}	9	9	10^{-8}		28	27	1.73	10^{-17}	10^{-16}	10^{-31}	OPT.
			10^{-12}		28	27	1.73	10^{-17}	10^{-16}	10^{-31}	OPT.
36d.^{0}	9	9	10^{-8}		816	211	232.	10^{-10}	10^{-10}	10^{-19}	*
			10^{-12}		816	211	232.	10^{-10}	10^{-10}	10^{-19}	*
37.	2	16	10^{-8}		13	8	8.85	10^{1}	10^{-6}	10^{-6}	*
			10^{-12}		13	8	8.85	10^{1}	10^{-6}	10^{-6}	*
38.	3	16	10^{-8}		17	12	26.1	10^{1}	10^{-5}	10^{-6}	*
			10^{-12}		17	12	26.1	10^{1}	10^{-5}	10^{-6}	*

Numerical Results for LSQFDQ

	n	m		max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
39a.	2	3	10^{-8}		9	6	10^{-6}	10^{-1}	10^{-9}	10^{-7}	OPT.
			10^{-12}		9	6	10^{-6}	10^{-1}	10^{-9}	10^{-7}	OPT.
39b.	2	3	10^{-8}		17	14	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
			10^{-12}		17	14	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
39c.	2	3	10^{-8}		11	6	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
			10^{-12}		11	6	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
39d.	2	3	10^{-8}		15	8	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
			10^{-12}		15	8	10^{-7}	10^{-1}	10^{-9}	10^{-7}	OPT.
39e.	2	3	10^{-8}		15	8	10^{-8}	10^{-1}	10^{-10}	10^{-7}	OPT.
			10^{-12}		15	8	10^{-8}	10^{-1}	10^{-10}	10^{-7}	OPT.
39f.	2	3	10^{-8}		27	10	10^{-9}	10^{-1}	10^{-9}	10^{-7}	OPT.
			10^{-12}		27	10	10^{-9}	10^{-1}	10^{-9}	10^{-7}	OPT.
39g.	2	3	10^{-8}		38	15	10^{-10}	10^{-1}	10^{-7}	10^{-7}	*
			10^{-12}		38	15	10^{-10}	10^{-1}	10^{-7}	10^{-7}	*
40a.	3	4	10^{-8}		12	9	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		12	9	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
40b.	3	4	10^{-8}		19	17	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		19	17	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
40c.	3	4	10^{-8}		27	22	10^{-7}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-12}		27	22	10^{-7}	10^{0}	10^{-8}	10^{-7}	OPT.
40d.	3	4	10^{-8}		18	8	10^{-7}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		18	8	10^{-7}	10^{0}	10^{-9}	10^{-7}	OPT.
40e.	3	4	10^{-8}		51	18	10^{-7}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		51	18	10^{-7}	10^{0}	10^{-9}	10^{-7}	OPT.
40f.	3	4	10^{-8}		96	32	10^{-8}	10^{0}	10^{-7}	10^{-7}	OPT.
			10^{-12}		98	33	10^{-8}	10^{0}	10^{-7}	10^{-7}	*
40g.	3	4	10^{-8}		103	32	10^{-9}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		103	32	10^{-9}	10^{0}	10^{-9}	10^{-7}	OPT.
41a.	5	10	10^{-8}		8	7	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
			10^{-12}		8	7	10^{-6}	10^{0}	10^{-9}	10^{-7}	OPT.
41b.	5	10	10^{-8}		16	13	10^{-6}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-12}		16	13	10^{-6}	10^{0}	10^{-8}	10^{-7}	OPT.
41c.	5	10	10^{-8}		19	18	10^{-6}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-12}		19	18	10^{-6}	10^{0}	10^{-8}	10^{-7}	OPT.
41d.	5	10	10^{-8}		27	14	10^{-6}	10^{0}	10^{-8}	10^{-7}	OPT.
			10^{-12}		27	14	10^{-6}	10^{0}	10^{-8}	10^{-7}	OPT.
41e.	5	10	10^{-8}		35	17	10^{-7}	10^{0}	10^{-8}	10^{-7}	*
			10^{-12}		35	17	10^{-7}	10^{0}	10^{-8}	10^{-7}	*
41f.	5	10	10^{-8}		44	23	10^{-7}	10^{0}	10^{-8}	10^{-7}	*
			10^{-12}		44	23	10^{-7}	10^{0}	10^{-8}	10^{-7}	*
41g.	5	10	10^{-8}		48	25	10^{-8}	10^{0}	10^{-7}	10^{-7}	*
			10^{-12}		48	25	10^{-8}	10^{0}	10^{-7}	10^{-7}	*

Numerical Results for LSQFDQ

	n	m		max. step	f, J evals.	iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-8}	10.0	80	42	63.5	10^{-11}	10^{-8}	10^{-22}	*
			10^{-12}	10.0	80	42	63.5	10^{-11}	10^{-8}	10^{-22}	*
42b.^{0}	4	24	10^{-8}	2.0	98	68	61.9	10^{-7}	10^{-4}	10^{-13}	*
			10^{-12}	2.0	98	68	61.9	10^{-7}	10^{-4}	10^{-13}	*
42c.^{0}	4	24	10^{-8}	5.0	47	27	60.3	10^{-7}	10^{-5}	10^{-14}	*
			10^{-12}	5.0	47	27	60.3	10^{-7}	10^{-5}	10^{-14}	*
42d.^{0}	4	24	10^{-8}	5.0	27	19	60.3	10^{-5}	10^{-3}	10^{-10}	*
			10^{-12}	5.0	27	19	60.3	10^{-5}	10^{-3}	10^{-10}	*
43a.^{0}	5	16	10^{-8}	10.0	33	18	54.0	10^{-9}	10^{-6}	10^{-17}	*
			10^{-12}	10.0	33	18	54.0	10^{-9}	10^{-6}	10^{-17}	*
43b.^{0}	5	16	10^{-8}	10.0	32	18	54.0	10^{-7}	10^{-5}	10^{-14}	*
			10^{-12}	10.0	32	18	54.0	10^{-7}	10^{-5}	10^{-14}	*
43c.^{0}	5	16	10^{-8}	10.0	33	18	54.0	10^{-8}	10^{-6}	10^{-16}	*
			10^{-12}	10.0	33	18	54.0	10^{-8}	10^{-6}	10^{-16}	*
43d.^{0}	5	16	10^{-8}	10.0	38	20	54.0	10^{-9}	10^{-7}	10^{-18}	*
			10^{-12}	10.0	38	20	54.0	10^{-9}	10^{-7}	10^{-18}	*
43e.^{0}	5	16	10^{-8}	10.0	28	14	54.0	10^{-8}	10^{-6}	10^{-15}	*
			10^{-12}	10.0	28	14	54.0	10^{-8}	10^{-6}	10^{-15}	*
43f.^{0}	5	16	10^{-8}		47	21	54.0	10^{-10}	10^{-7}	10^{-19}	*
			10^{-12}		47	21	54.0	10^{-10}	10^{-7}	10^{-19}	*
44a.^{0}	6	6	10^{-8}		104	29	4.03	10^{-14}	10^{-12}	10^{-27}	*
			10^{-12}		104	29	4.03	10^{-14}	10^{-12}	10^{-27}	*
44b.^{0}	6	6	10^{-8}		10	6	3.52	10^{-8}	10^{-6}	10^{-15}	*
			10^{-12}		10	6	3.52	10^{-8}	10^{-6}	10^{-15}	*
44c.^{0}	6	6	10^{-8}		63	22	20.6	10^{-12}	10^{-8}	10^{-23}	*
			10^{-12}		63	22	20.6	10^{-12}	10^{-8}	10^{-23}	*
44d.^{0}	6	6	10^{-8}		41	17	15.3	10^{-7}	10^{-3}	10^{-14}	*
			10^{-12}		41	17	15.3	10^{-7}	10^{-3}	10^{-14}	*
44e.^{0}	6	6	10^{-8}		83	28	9.27	10^{-10}	10^{-7}	10^{-20}	*
			10^{-12}		83	28	9.27	10^{-10}	10^{-7}	10^{-20}	*
45a.^{0}	8	8	10^{-8}		104	29	4.06	10^{-14}	10^{-12}	10^{-27}	*
			10^{-12}		104	29	4.06	10^{-14}	10^{-12}	10^{-27}	*
45b.^{0}	8	8	10^{-8}		12	7	3.56	10^{-8}	10^{-6}	10^{-15}	*
			10^{-12}		12	7	3.56	10^{-8}	10^{-6}	10^{-15}	*
45c.^{0}	8	8	10^{-8}		63	22	20.6	10^{-12}	10^{-8}	10^{-23}	*
			10^{-12}		63	22	20.6	10^{-12}	10^{-8}	10^{-23}	*
45d.^{0}	8	8	10^{-8}		45	18	15.3	10^{-7}	10^{-3}	10^{-14}	*
			10^{-12}		45	18	15.3	10^{-7}	10^{-3}	10^{-14}	*
45e.^{0}	8	8	10^{-8}		85	29	9.31	10^{-10}	10^{-7}	10^{-20}	*
			10^{-12}		85	29	9.31	10^{-10}	10^{-7}	10^{-20}	*

5.6.3 Adaptive Method

(PORT/ACM DN2G/NL2SOL)

5.6.3.1 Software and Algorithm

The results were obtained using subroutine DN2G, a double precision version of the ACM algorithm NL2SOL available in the PORT Library [1984]. A subproblem of the form

$$\min_{p \in \Re^n} Q_k(p) \equiv \bar{g}_k^T p + \frac{1}{2} p^T (J_k^T J_k + B_k) p$$

$$\text{subject to} \quad \|D_k p\|_2 \leq \delta_k$$

is solved at each iteration for the step p_k to the next iterate, where D_k is a diagonal scaling matrix. The method is adaptive, so that B_k is sometimes null and sometimes a scaled quasi-Newton approximation to the part of the Hessian involving the second derivatives of f.

5.6.3.2 Parameters

Parameters were kept at their default values with the following exceptions :

IV(MXFCAL) -	min $\{9999, 1000 * n\}$	function evaluation limit
IV(MXITER) -	min $\{9999, 1000 * n\}$	iteration limit
V(AFCTOL) -	TOL*TOL (varied; see tables)	absolute function convergence tolerance
V(RFCTOL) -	TOL (varied; see tables)	relative function convergence tolerance
V(SCTOL) -	ϵ_M	singular convergence tolerance
V(XCTOL) -	TOL (varied; see tables)	x convergence tolerance
V(XFTOL) -	ϵ_M	false convergence tolerance
V(LMAX0) -	usually 1.0 (default) †	initial trust-region diameter
V(LMAXS) -	1.0 (default)	step bound for singular convergence test
V(TUNER1) -	0.1 (default)	reduction test coefficient

† In some cases the default V(LMAX0) = 1.0 for the initial diameter of the trust-region was too large and overflow occurred during function evaluation. These cases are indicated in the table by giving the lower value of V(LMAX0) that was subsequently used to obtain the results in the column labeled "init. diam.".

See Dennis, Gay, and Welsch [1981a, 1981b], Gay [1983], and PORT [1984] for details concerning the parameters.

5.6.3.3 Convergence Criteria

The following quantities will be used in describing the convergence criteria :

$$\begin{aligned}
\text{objective function} &: \mathcal{F}_k \equiv f_k^{\mathrm{T}} f_k \\
\text{objective gradient} &: \bar{g}_k = \nabla \mathcal{F}_k \equiv 2 J_k^{\mathrm{T}} f_k \\
\text{current step} &: p_k, \text{ the minimizer of the subproblem} \\
\text{Newton step} &: p_N \begin{cases} H_k^{-1} g & \text{if } H_k \text{ is positive definite;} \\ \textit{undefined} & \text{otherwise.} \end{cases} \\
\text{Newton reduction} &: \rho_N = \begin{cases} -\mathcal{Q}_k(p_N) & \text{if } H_k \text{ is positive definite;} \\ 0 & \text{otherwise.} \end{cases} \\
\text{predicted reduction} &: \rho_P = -\mathcal{Q}_k(p_k) \\
\text{actual reduction} &: \rho_A = \mathcal{F}_k - \mathcal{F}(x_k + p_k) \\
\text{scaled distance} &: \nu(x, y, D) = \frac{\max_{1 \le i \le n} \{|(D(x-y))^i|\}}{\max_{1 \le i \le n} \{|(Dx)^i| + |(Dy)^i|\}},
\end{aligned}$$

† Here v_i denotes the ith component of the vector v. There is a provision for the user to replace the function ν; we used the default in all of the tests.

The convergence criteria used in DN2G are as follows :

- *Absolute function convergence* occurs at x_k if

$$|\mathcal{F}_k| < \mathrm{V}(\mathtt{AFCTOL}).$$

- *Relative function convergence* is intended to approximate the condition

$$\mathcal{F}_k - \mathcal{F}(x_*) \le \mathrm{V}(\mathtt{RFCTOL})\,|\mathcal{F}_k|\,.$$

The test actually used is

$$\rho_N \le \mathrm{V}(\mathtt{RFCTOL})\,|\mathcal{F}_k|\,. \tag{5.6.13}$$

- x *convergence* is intended to approximate the condition

$$\nu(x_k, x^*, D_k) \le \mathrm{V}(\mathtt{XCTOL}),$$

The test actually used is

$$p_k = p_N \quad \text{and} \quad \nu(x_k, x_k + p_k, D_k) \le \mathrm{V}(\mathtt{XCTOL}). \tag{5.6.14}$$

- *Singular convergence* is intended to approximate the condition

$$\mathcal{F}_k - \min\,\{\mathcal{F}(y) \mid \|D_k(y - x_k)\| \le \mathrm{V}(\mathtt{LMAXS})\} < \mathrm{V}(\mathtt{SCTOL})\,|\mathcal{F}_k|\,,$$

where D_k is the diagonal scaling matrix at the kth iterate — when none of the convergence criteria listed above hold. It is meant to indicate relative function convegence when the Hessian in the subproblem is singular.

The actual test is

$$\mathcal{F}_k - \min\{\mathcal{Q}_k(y) \mid \|D_k(y - x_k)\| \leq \texttt{V(LMAXS)}\} < \texttt{V(SCTOL)}\,|\mathcal{F}_k|\,. \tag{5.6.15}$$

Under certain conditions, the test is repeated for a step of length V(LMAXS).

• *False convergence* is returned if none of the other convergence criteria are satisfied and a trial step no larger than V(XFCTOL) is rejected. This usually indicates either an error in computing the objective gradient, or a discontinuity (in $\mathcal{F}$ or g) near the current iterate, or that one or more of the convergence tolerances (V(RFCTOL), V(XCTOL), and V(AFCTOL)) are too small relative the accuracy to which the objective is computed.

The test actually used is

$$\mathcal{F}_k - \mathcal{F}(x_k + p_k) \leq \texttt{V(TUNER1)}\rho_P \quad \text{and} \quad \nu(x_k, x_k + p_k, D_k) \leq \texttt{V(XFTOL)}, \tag{5.6.16}$$

where the parameter V(TUNER1) is adjustable, although in these tests the default value 0.1 is used throughout.

Except for test (5.6.13), tests for convergence are performed only when

$$\rho_A \leq 2\rho_P. \tag{5.6.17}$$

See Dennis, Gay, and Welsch [1981a, 1981b], Gay [1983], and PORT [1984] for more discussion of the convergence criteria.

The following abbreviations are used in the tables to describe the conditions under which the algorithm terminates :

ABS. F -	(5.6.13)
REL. F -	(5.6.14) and (5.6.17)
X -	(5.6.15) and (5.6.17)
X, F -	(5.6.14) and (5.6.15) and (5.6.17)
SING. -	(5.6.16) and (5.6.17)
FALSE -	(5.6.17) and (5.6.17)
F LIM. -	function evaluation limit reached
TIME -	time limit exceeded
LOOP -	subroutine appears to loop

The total number of Jacobian evaluations is either equal to the total number of iterations of the method, or it is one more than the number of iterations. The number in the column labeled "iters. / J evals." is followed by a "+" if an extra Jacobian evaluation was used in the computation.

Numerical Results for DN2G

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
1.^{0}	2	2	10^{-8}		14	11+	1.41	10^{-16}	10^{-15}	10^{-32}	ABS. F
			10^{-12}		14	11+	1.41	10^{-16}	10^{-15}	10^{-32}	ABS. F
2.^{0}	2	2	10^{-8}		10	8+	11.4	10^{1}	10^{-4}	10^{1}	REL. F
			10^{-12}		12	10	11.4	10^{1}	10^{-6}	10^{1}	REL. F
3.^{0}	2	2	10^{-8}		64	50+	9.11	10^{-9}	10^{-4}	10^{-19}	ABS. F
			10^{-12}		65	51+	9.11	10^{-17}	10^{-12}	10^{-33}	ABS. F
4.^{0}	2	3	10^{-8}		40	33+	10^{6}	10^{-2}	10^{-2}	10^{-4}	X
			10^{-12}		53	46+	10^{6}	10^{-6}	10^{-6}	10^{-12}	X
5.^{0}	2	3	10^{-8}		9	7+	3.04	10^{-11}	10^{-10}	10^{-21}	ABS. F
			10^{-12}		10	8+	3.04	0.00	0.00	0.00	ABS. F
6.	2	10	10^{-8}		14	10+	.365	10^{1}	10^{-2}	10^{-6}	REL. F
			10^{-12}		16	12+	.365	10^{1}	10^{-7}	10^{-6}	REL. F
7.^{0}	3	3	10^{-8}		13	8+	1.00	10^{-11}	10^{-9}	10^{-21}	ABS. F
			10^{-12}		14	9+	1.00	10^{-23}	10^{-21}	10^{-45}	ABS. F
8.	3	15	10^{-8}		7	6+	2.60	10^{-1}	10^{-9}	10^{-8}	REL. F
			10^{-12}		8	7+	2.60	10^{-1}	10^{-11}	10^{-8}	REL. F
9.	3	15	10^{-8}		3	2+	1.08	10^{-4}	10^{-12}	10^{-14}	X
			10^{-12}		5	4	1.08	10^{-4}	10^{-16}	10^{-14}	X, REL. F
10.	3	16	10^{-8}		132	120+	10^{4}	10^{1}	10^{-3}	10^{-6}	X, REL. F
			10^{-12}		133	121	10^{4}	10^{1}	10^{-3}	10^{-6}	X, REL. F
11.^{0}	3	10	10^{-8}		(3000)	(2953)	239.	10^{-2}	10^{18}	10^{-5}	F LIM.
			10^{-12}		(3000)	(2953)	239.	10^{-2}	10^{18}	10^{-5}	F LIM.
12.^{0}	3	10	10^{-8}		8	5+	10.1	10^{-10}	10^{-10}	10^{-19}	ABS. F
			10^{-12}		9	6+	10.1	10^{-16}	10^{-16}	10^{-32}	ABS. F
13.^{0}	4	4	10^{-8}		19	16+	10^{-4}	10^{-8}	10^{-12}	10^{-17}	ABS. F
			10^{-12}		25	22+	10^{-6}	10^{-12}	10^{-17}	10^{-24}	ABS. F
14.^{0}	4	6	10^{-8}		52	40+	2.00	10^{-9}	10^{-7}	10^{-17}	ABS. F
			10^{-12}		53	41+	2.00	0.00	0.00	0.00	ABS. F
15.	4	11	10^{-8}		11	9+	.328	10^{-2}	10^{-9}	10^{-9}	REL. F
			10^{-12}		12	10+	.328	10^{-2}	10^{-9}	10^{-9}	REL. F
16.	4	20	10^{-8}		21	14+	17.6	10^{2}	10^{-2}	10^{-8}	REL. F
			10^{-12}		22	15+	17.6	10^{2}	10^{-4}	10^{-8}	REL. F
17.	5	33	10^{-8}		26	20+	2.46	10^{-2}	10^{-8}	10^{-11}	REL. F
			10^{-12}		27	21+	2.46	10^{-2}	10^{-10}	10^{-11}	REL. F
18.^{0}	6	13	10^{-8}		45	32+	12.3	10^{-9}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		46	33+	12.3	10^{-16}	10^{-16}	10^{-31}	ABS. F
19.	11	65	10^{-8}		20	13+	9.38	10^{-1}	10^{-7}	10^{-8}	REL. F
			10^{-12}		22	15+	9.38	10^{-1}	10^{-9}	10^{-8}	REL. F

Numerical Results for DN2G

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
20a.	6	31	10^{-8}		13	9+	2.44	10^{-2}	10^{-9}	10^{-10}	REL. F
			10^{-12}		13	9+	2.44	10^{-2}	10^{-9}	10^{-10}	REL. F
20b.	9	31	10^{-8}		12	9+	6.06	10^{-3}	10^{-11}	10^{-13}	REL. F
			10^{-12}		15	12	6.06	10^{-3}	10^{-14}	10^{-13}	X, REL. F
20c.	12	31	10^{-8}		14	11+	16.6	10^{-5}	10^{-13}	10^{-16}	REL. F
			10^{-12}		14	11+	16.6	10^{-5}	10^{-13}	10^{-16}	REL. F
20d.	20	31	10^{-8}		8	6+	1.11	10^{-8}	10^{-13}	10^{-16}	ABS. F
			10^{-12}		(471)	(137+)	1.18	10^{-8}	10^{-14}	10^{-16}	LOOP
21a.^{0}	10	10	10^{-8}		27	17+	3.16	10^{-16}	10^{-14}	10^{-31}	ABS. F
			10^{-12}		27	17+	3.16	10^{-16}	10^{-14}	10^{-31}	ABS. F
21b.^{0}	20	20	10^{-8}		16	12+	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F
			10^{-12}		16	12+	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F
22a.^{0}	12	12	10^{-8}		19	16+	10^{-4}	10^{-8}	10^{-12}	10^{-17}	ABS. F
			10^{-12}		26	23+	10^{-6}	10^{-12}	10^{-18}	10^{-24}	ABS. F
22b.^{0}	20	20	10^{-8}		20	17+	10^{-4}	10^{-8}	10^{-12}	10^{-17}	ABS. F
			10^{-12}		26	23+	10^{-6}	10^{-12}	10^{-18}	10^{-24}	ABS. F
23a.	4	5	10^{-8}		36	26+	.500	10^{-3}	10^{-10}	10^{-10}	REL. F
			10^{-12}		37	27+	.500	10^{-3}	10^{-10}	10^{-10}	REL. F
23b.	10	11	10^{-8}		61	46+	.500	10^{-2}	10^{-7}	10^{-11}	REL. F
			10^{-12}		68	50+	.500	10^{-2}	10^{-10}	10^{-11}	REL. F
24a.	4	8	10^{-8}		139	110+	.759	10^{-3}	10^{-7}	10^{-11}	REL. F
			10^{-12}		142	113+	.759	10^{-3}	10^{-11}	10^{-11}	REL. F
24b.	10	20	10^{-8}		129	101+	.598	10^{-2}	10^{-7}	10^{-9}	REL. F
			10^{-12}		138	108+	.598	10^{-2}	10^{-8}	10^{-9}	REL. F
25a.^{0}	10	12	10^{-8}		15	10+	3.16	10^{-12}	10^{-11}	10^{-24}	ABS. F
			10^{-12}		16	11+	3.16	10^{-15}	10^{-14}	10^{-31}	X
25b.^{0}	20	22	10^{-8}		19	12+	4.47	10^{-15}	10^{-13}	10^{-29}	X
			10^{-12}		19	12+	4.47	10^{-15}	10^{-13}	10^{-29}	ABS. F
26a.^{0}	10	10	10^{-8}		11	7+	.306	10^{-9}	10^{-10}	10^{-19}	ABS. F
			10^{-12}		12	8+	.306	10^{-15}	10^{-15}	10^{-30}	ABS. F
26b.^{0}	20	20	10^{-8}		39	25+	.228	10^{-3}	10^{-9}	10^{-6}	REL. F
			10^{-12}		42	27	.228	10^{-3}	10^{-10}	10^{-6}	REL. F
27a.^{0}	10	10	10^{-8}		8	6+	3.18	10^{-10}	10^{-10}	10^{-21}	ABS. F
			10^{-12}		9	7+	3.18	10^{-15}	10^{-14}	10^{-29}	ABS. F
27b.^{0}	20	20	10^{-8}		11	8+	4.47	10^{-8}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		12	9+	4.47	10^{-14}	10^{-13}	10^{-27}	ABS. F
28a.^{0}	10	10	10^{-8}		4	3+	.412	10^{-15}	10^{-16}	10^{-31}	ABS. F
			10^{-12}		4	3+	.412	10^{-15}	10^{-16}	10^{-31}	ABS. F
28b.^{0}	20	20	10^{-8}		3	2+	.571	10^{-8}	10^{-9}	10^{-16}	ABS. F
			10^{-12}		4	3+	.571	10^{-16}	10^{-16}	10^{-32}	ABS. F

Numerical Results for DN2G

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
29a.^{0}	10	10	10^{-8}		4	3+	.412	10^{-14}	10^{-14}	10^{-29}	ABS. F
			10^{-12}		4	3+	.412	10^{-14}	10^{-14}	10^{-29}	ABS. F
29b.^{0}	20	20	10^{-8}		4	3+	.571	10^{-14}	10^{-14}	10^{-28}	ABS. F
			10^{-12}		4	3+	.571	10^{-14}	10^{-14}	10^{-28}	ABS. F
30a.^{0}	10	10	10^{-8}		8	5+	2.05	10^{-9}	10^{-9}	10^{-18}	ABS. F
			10^{-12}		9	6+	2.05	10^{-16}	10^{-15}	10^{-31}	ABS. F
30b.^{0}	20	20	10^{-8}		8	5+	3.04	10^{-8}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		9	6+	3.04	10^{-15}	10^{-14}	10^{-30}	ABS. F
31a.^{0}	10	10	10^{-8}		10	7+	1.80	10^{-12}	10^{-11}	10^{-23}	ABS. F
			10^{-12}		11	8+	1.80	10^{-16}	10^{-15}	10^{-31}	X
31b.^{0}	20	20	10^{-8}		10	7+	2.66	10^{-10}	10^{-10}	10^{-21}	ABS. F
			10^{-12}		11	8+	2.66	10^{-15}	10^{-14}	10^{-30}	ABS. F
32.^{L}	10	20	10^{-8}		5	2	3.16	10^{0}	10^{-14}	10^{-16}	X, REL. F
			10^{-12}		5	2	3.16	10^{0}	10^{-14}	10^{-16}	X, REL. F
33.^{L}	10	20	10^{-8}		18	5	20.2	10^{0}	10^{-11}	10^{-6}	SING.
			10^{-12}		18	5	20.2	10^{0}	10^{-11}	10^{-6}	SING.
34.^{L}	10	20	10^{-8}		13	5	7.24	10^{0}	10^{-10}	10^{-6}	SING.
			10^{-12}		13	5	7.24	10^{0}	10^{-10}	10^{-6}	SING.
35a.	8	8	10^{-8}		23	14+	1.65	10^{-1}	10^{-7}	10^{-9}	REL. F
			10^{-12}		24	15+	1.65	10^{-1}	10^{-8}	10^{-9}	REL. F
35b.^{0}	9	9	10^{-8}		11	7+	1.73	10^{-12}	10^{-12}	10^{-24}	ABS. F
			10^{-12}		11	7+	1.73	10^{-12}	10^{-12}	10^{-24}	ABS. F
35c.	10	10	10^{-8}		17	12+	1.81	10^{-1}	10^{-7}	10^{-3}	REL. F
			10^{-12}		19	14+	1.81	10^{-1}	10^{-8}	10^{-3}	REL. F
36a.^{0}	4	4	10^{-8}		(4000)	(3988)	27.7	10^{-7}	10^{-7}	10^{-13}	F LIM.
			10^{-12}		(4000)	(3988)	27.7	10^{-7}	10^{-7}	10^{-13}	F LIM.
36b.^{0}	9	9	10^{-8}		(9000)	(8977)	35.5	10^{-7}	10^{-8}	10^{-14}	F LIM.
			10^{-12}		(9000)	(8977)	35.5	10^{-7}	10^{-8}	10^{-14}	F LIM.
36c.^{0}	9	9	10^{-8}		16	15+	1.73	10^{-9}	10^{-8}	10^{-17}	ABS. F
			10^{-12}		22	21+	1.73	10^{-12}	10^{-12}	10^{-24}	ABS. F
36d.^{0}	9	9	10^{-8}		(9000)	(8966)	38.4	10^{-7}	10^{-7}	10^{-14}	F LIM.
			10^{-12}		(9000)	(8966)	38.4	10^{-7}	10^{-7}	10^{-14}	F LIM.
37.	2	16	10^{-8}		10	8+	8.85	10^{1}	10^{-3}	10^{-6}	REL. F
			10^{-12}		11	9+	8.85	10^{1}	10^{-5}	10^{-6}	REL. F
38.	3	16	10^{-8}		10	8+	26.1	10^{1}	10^{-2}	10^{-6}	REL. F
			10^{-12}		12	10+	26.1	10^{1}	10^{-7}	10^{-6}	REL. F

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
39a.	2	3	10^{-8}		5	4+	10^{-6}	10^{-1}	10^{-10}	10^{-7}	REL. F
			10^{-12}		5	4+	10^{-7}	10^{-1}	10^{-10}	10^{-7}	REL. F
39b.	2	3	10^{-8}		6	5+	10^{-7}	10^{-1}	10^{-7}	10^{-7}	REL. F
			10^{-12}		7	6+	10^{-7}	10^{-1}	10^{-9}	10^{-7}	REL. F
39c.	2	3	10^{-8}		7	4+	10^{-7}	10^{-1}	10^{-7}	10^{-7}	REL. F
			10^{-12}		8	5+	10^{-7}	10^{-1}	10^{-10}	10^{-7}	REL. F
39d.	2	3	10^{-8}		7	5+	10^{-7}	10^{-1}	10^{-6}	10^{-7}	REL. F
			10^{-12}		8	6+	10^{-7}	10^{-1}	10^{-10}	10^{-7}	REL. F
39e.	2	3	10^{-8}		9	6+	10^{-8}	10^{-1}	10^{-7}	10^{-7}	REL. F
			10^{-12}		10	7+	10^{-8}	10^{-1}	10^{-10}	10^{-7}	REL. F
39f.	2	3	10^{-8}		14	10+	10^{-8}	10^{-1}	10^{-6}	10^{-7}	REL. F
			10^{-12}		15	11+	10^{-9}	10^{-1}	10^{-9}	10^{-7}	REL. F
39g.	2	3	10^{-8}		18	12+	10^{-7}	10^{-1}	10^{-4}	10^{-7}	REL. F
			10^{-12}		20	14+	10^{-9}	10^{-1}	10^{-6}	10^{-7}	REL. F
40a.	3	4	10^{-8}		7	6	10^{-6}	10^{0}	10^{-11}	10^{-7}	REL. F
			10^{-12}		7	6	10^{-6}	10^{0}	10^{-11}	10^{-7}	REL. F
40b.	3	4	10^{-8}		7	5+	10^{-6}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		11	9	10^{-6}	10^{0}	10^{-14}	10^{-7}	REL. F
40c.	3	4	10^{-8}		9	6+	10^{-7}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		10	7+	10^{-7}	10^{0}	10^{-9}	10^{-7}	REL. F
40d.	3	4	10^{-8}		9	7+	10^{-7}	10^{0}	10^{-8}	10^{-7}	REL. F
			10^{-12}		9	7+	10^{-7}	10^{0}	10^{-8}	10^{-7}	REL. F
40e.	3	4	10^{-8}		10	9+	10^{-7}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		11	10+	10^{-7}	10^{0}	10^{-6}	10^{-7}	REL. F
40f.	3	4	10^{-8}		13	10+	10^{-8}	10^{0}	10^{-6}	10^{-7}	REL. F
			10^{-12}		14	11+	10^{-8}	10^{0}	10^{-7}	10^{-7}	REL. F
40g.	3	4	10^{-8}		23	16+	10^{-8}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		25	18+	10^{-9}	10^{0}	10^{-7}	10^{-7}	REL. F
41a.	5	10	10^{-8}		4	3+	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
			10^{-12}		4	3+	10^{-6}	10^{0}	10^{-10}	10^{-7}	REL. F
41b.	5	10	10^{-8}		4	3+	10^{-6}	10^{0}	10^{-7}	10^{-7}	REL. F
			10^{-12}		5	4+	10^{-6}	10^{0}	10^{-11}	10^{-7}	REL. F
41c.	5	10	10^{-8}		6	5+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
			10^{-12}		6	5+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
41d.	5	10	10^{-8}		9	7+	10^{-6}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		11	9+	10^{-6}	10^{0}	10^{-8}	10^{-7}	REL. F
41e.	5	10	10^{-8}		17	13+	10^{-6}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		20	16+	10^{-7}	10^{0}	10^{-8}	10^{-7}	REL. F
41f.	5	10	10^{-8}		24	19+	10^{-6}	10^{0}	10^{-4}	10^{-7}	REL. F
			10^{-12}		27	22+	10^{-7}	10^{0}	10^{-7}	10^{-7}	REL. F
41g.	5	10	10^{-8}		29	22+	10^{-7}	10^{0}	10^{-5}	10^{-7}	REL. F
			10^{-12}		30	23+	10^{-8}	10^{0}	10^{-6}	10^{-7}	REL. F

Numerical Results for DN2G

	n	m	TOL	init. diam.	f evals.	iters./ J evals.	$\|\|x^*\|\|_2$	$\|\|f^*\|\|_2$	$\|\|\bar{g}^*\|\|_2$	est. err.	conv.
42a.^{0}	4	24	10^{-8}	0.9	29	19+	60.8	10^{-13}	10^{-10}	10^{-25}	x
			10^{-12}	0.9	29	19+	60.8	10^{-13}	10^{-10}	10^{-25}	ABS. F
42b.^{0}	4	24	10^{-8}	0.001	74	48+	61.3	10^{-13}	10^{-11}	10^{-25}	x
			10^{-12}	0.001	74	48+	61.3	10^{-13}	10^{-11}	10^{-25}	ABS. F
42c.^{0}	4	24	10^{-8}	0.01	32	19+	60.3	10^{-13}	10^{-10}	10^{-26}	x
			10^{-12}	0.01	32	19+	60.3	10^{-13}	10^{-10}	10^{-26}	ABS. F
42d.^{0}	4	24	10^{-8}		23	18+	60.3	10^{-10}	10^{-7}	10^{-19}	ABS. F
			10^{-12}		24	19+	60.3	10^{-13}	10^{-10}	10^{-26}	x
43a.^{0}	5	16	10^{-8}		31	22+	54.0	10^{-12}	10^{-10}	10^{-24}	ABS. F
			10^{-12}		32	23+	54.0	10^{-14}	10^{-11}	10^{-28}	x
43b.^{0}	5	16	10^{-8}		20	13+	54.0	10^{-12}	10^{-10}	10^{-24}	ABS. F
			10^{-12}		20	13+	54.0	10^{-12}	10^{-10}	10^{-24}	ABS. F
43c.^{0}	5	16	10^{-8}		34	26+	53.6	10^{-1}	10^{-7}	10^{-2}	REL. F
			10^{-12}		41	33+	53.6	10^{-1}	10^{-10}	10^{-2}	REL. F
43d.^{0}	5	16	10^{-8}		17	11+	54.0	10^{-14}	10^{-11}	10^{-27}	x
			10^{-12}		17	11+	54.0	10^{-14}	10^{-11}	10^{-27}	ABS. F
43e.^{0}	5	16	10^{-8}		28	18+	54.0	10^{-11}	10^{-9}	10^{-22}	ABS. F
			10^{-12}		29	19+	54.0	10^{-14}	10^{-12}	10^{-27}	x
43f.^{0}	5	16	10^{-8}		20	15+	54.0	10^{-12}	10^{-10}	10^{-25}	ABS. F
			10^{-12}		20	15+	54.0	10^{-12}	10^{-10}	10^{-25}	ABS. F
44a.^{0}	6	6	10^{-8}		58	41+	4.06	10^{-10}	10^{-8}	10^{-20}	ABS. F
			10^{-12}		59	42+	4.06	10^{-15}	10^{-13}	10^{-30}	ABS. F
44b.^{0}	6	6	10^{-8}		7	6+	3.52	10^{-14}	10^{-12}	10^{-28}	x
			10^{-12}		7	6+	3.52	10^{-14}	10^{-12}	10^{-28}	ABS. F
44c.^{0}	6	6	10^{-8}		93	84+	20.6	10^{-10}	10^{-6}	10^{-19}	ABS. F
			10^{-12}		94	85+	20.6	10^{-15}	10^{-11}	10^{-29}	ABS. F
44d.^{0}	6	6	10^{-8}		97	81+	15.3	10^{-11}	10^{-8}	10^{-22}	ABS. F
			10^{-12}		98	82+	15.3	10^{-14}	10^{-11}	10^{-28}	x
44e.^{0}	6	6	10^{-8}		83	72+	9.27	10^{-14}	10^{-12}	10^{-29}	x
			10^{-12}		83	72+	9.27	10^{-14}	10^{-12}	10^{-29}	ABS. F
45a.^{0}	8	8	10^{-8}		65	45+	4.06	10^{-11}	10^{-9}	10^{-22}	ABS. F
			10^{-12}		66	46+	4.06	10^{-17}	10^{-15}	10^{-33}	ABS. F
45b.^{0}	8	8	10^{-8}		8	7+	3.56	10^{-13}	10^{-12}	10^{-26}	ABS. F
			10^{-12}		8	7+	3.56	10^{-13}	10^{-12}	10^{-26}	ABS. F
45c.^{0}	8	8	10^{-8}		129	123+	20.6	10^{-8}	10^{-4}	10^{-16}	ABS. F
			10^{-12}		130	124+	20.6	10^{-15}	10^{-10}	10^{-29}	ABS. F
45d.^{0}	8	8	10^{-8}		168	144+	15.3	10^{-14}	10^{-11}	10^{-29}	x
			10^{-12}		168	144+	15.3	10^{-14}	10^{-11}	10^{-29}	ABS. F
45e.^{0}	8	8	10^{-8}		173	165+	9.31	10^{-15}	10^{-12}	10^{-29}	x
			10^{-12}		173	165+	9.31	10^{-15}	10^{-12}	10^{-29}	ABS. F

5.7 Discussion and Summary of Numerical Results for Chapters 2, 4, and 5

In this section, we briefly summarize the numerical results obtained for unconstrained optimization methods in Chapter 2, and nonlinear least-squares methods in Chapters 4 and 5; more detailed results are tabulated in Sections 2.6, 4.7, and 5.6. The tests were performed using the following software :

subroutine	source	problem type	derivatives
DMNG/SUMSOL	PORT	unconstrained optimization	first
NPSOL	SOL / NAG	unconstrained optimization	first
DMNH/HUMSOL	PORT	unconstrained optimization	second
MNA	NPL / NAG	unconstrained optimization	second
G-N	uses SOL / NAG LSSOL	nonlinear least squares	first
LMDER	MINPACK	nonlinear least squares	first
DN2G/NL2SOL	PORT	nonlinear least squares	first
LSQFDQ	NPL / NAG	nonlinear least squares	first
LSQSDN	NPL / NAG	nonlinear least squares	second

Information about the individual test problems is given in the Appendix. The number of function evaluations required by each subroutine is listed in the tables below. In addition, the following symbols are used :

- ^{0} - zero-residual problem
- ^{L} - linear least-squares problem
- — - failure to achieve an approximate solution
- $\sim$ - appears to be unable to terminate at an approximate solution
- ^{l} - local minimum
- ^{t} - termination criteria satisfied at a point away from a local minimum
- ^{s} - failed with default step length or trust-region size

Two columns of figures corresponding to two different values of a single parameter are given for each subroutine. For the Gauss-Newton methods, the parameter affects rank estimation; for all of the other methods, the parameter affects termination criteria. See the detailed tables of numerical results in the relevant chapters for information about the precise choices that were made. The wide variability in the numerical results makes it difficult to draw definitive conclusions about the relative performance of the software, because observations

of small samples could result in misleading generalizations. The sources of this variability are discussed below in some detail.

First, the number of function evaluations may not be an adequate basis for comparison. The routines vary in the number of gradient evaluations performed per function evaluation, and second-derivative methods require evaluation of the Hessian matrix. Moreover, when function evaluations are relatively inexpensive, costs could be dominated by other portions of the computation. Another difficulty in making comparisons is that the definition of an acceptable minimum varies from routine to routine. For example, the norm of the gradient of the nonlinear least-squares objective, $\|\bar{g}\|$, at an alleged solution x^* may differ considerably for different software, although $\bar{g}(x^*) = 0$ is a necessary condition for a minimum. (On problem 10., LMDER terminates at a point for which $\|\bar{g}\|$ is of order 10^0, while DN2G terminates at a point for which $\|\bar{g}\|$ is of order 10^{-3}.) Most algorithms do not attempt to reduce $\|\bar{g}\|$ directly, but convergence criteria may include a threshold on $\|\bar{g}(x^*)\|$. Depending on how this threshold is used in relation to other criteria, some routines may spend more function evaluations in anticipation of a reduction in $\|\bar{g}\|$ than others. A small value of $\|\bar{g}\|$ means greater certainty that a minimum has actually been obtained, but may be unreasonably expensive to achieve in practice.

Second, aside from design choices that define a particular implementation of an algorithm, the user is permitted to specify certain parameters that may affect performance. In Chapter 4, we saw that Gauss-Newton methods may be sensitive to rank estimation criteria (see, for example, problems 35b., 36a., and 20d. that were discussed in Section 4.5). For problems on which an algorithm is linearly convergent, small changes in tolerances that are used to define convergence criteria can mean substantial differences in the amount of computation required in order to obtain a point satisfying conditions for convergence (see, for example, DMNG on 24b., LMDER on 40., and NPSOL on 45e.). Selection of a maximum steplength or an initial trust-region radius can also be critical factor in the performance of a method. In these tests, the default values for these parameters were altered only in cases where a method was initially observed to fail by attempting to evaluate problem functions outside the region in which they are numerically defined (see, for example, the DeVilliers and Glasser test problems 42. and 43.). In NPSOL, the mechanism for dealing with this type of difficulty is to put bounds on variables rather than adjusting maximum step length. The version of MNA that is available in the NAG Library (E04LBF) also provides for bounds on

variables, and there are alternative versions of all of the PORT software used in these tests (DMNH, DMNG, and DN2G) that allow bounds to be specified. Unfortunately, when bounds on the variables are included in the formulation, local minima at which the bounds are active may be found rather than local minima for the nonlinear least-squares problem (see the results for NPSOL on the DeVilliers and Glasser test problems 42. and 43.).

Third, the performance of any given method over the set of test problems is by no means uniform, and it is not easy to separate the problems into classes for which the behavior of an algorithm can be categorized. One reason for this is that many of the test problems recur in the literature precisely because they have certain distinguishing properties. Powell's singular function and variants (13. and 22.) are zero-residual problems in which the Jacobian becomes singular at the solution. The McKeown test problems (39., 40., and 41.) are chosen so that the Jacobian is well-conditioned everywhere, and the rate of convergence for Gauss-Newton can be controlled by varying a single parameter (the parameter can also be chosen so that Gauss-Newton diverges). Both Powell's singular function and McKeown's test problems are constructed analytically rather than derived from data-fitting applications. The matrix square root problems (36.) are examples of small, dense, nonlinear systems of equations requiring a very accurate solution. Watson's function (20.) comes from polynomial interpolation, and has multiple local minima with small, but nonzero, residuals. It also has the feature that the Jacobian becomes increasingly ill-conditioned as the problem size is increased (see Section 4.5). The Gulf Research and Development function (11.) has discontinuties in the derivative of each residual on a one-dimensional subspace and hence violates the assumption (made in developing all of the algorithms we have discussed) that the sum of squares has continuous second derivatives. The DeVilliers and Glasser test problems (42. and 43.) illustrate variability in performance due to the use of different starting values. More generally, considerable differences in performance may occur for a given type of residual function over several sets of defining data of similar magnitude, as shown by the Dennis, Gay, and Vu test problems (44. and 45.).

Finally, there is considerable variability in performance among the routines tested, and few generalizations are possible. Our data generally supports the use of nonlinear least-squares software over that designed for general unconstrained minimization, but there are some exceptions (see, for example, the McKeown test problems 39. - 41.). Of the nonlinear least-squares routines, DN2G (NL2SOL) is often the best (the Dennis, Gay, and Vu test

problems 44. and 45. are examples of exceptions). When second derivatives are relatively cheap to obtain, the use of an unconstrained optimization method that uses exact second derivatives may be a reasonable alternative to a nonlinear least-squares method (see, for example, the penalty functions 23. and 24.). Our tests do not indicate overall superiority of any particular method over the others; in situations in which a variety of problems are being solved, we conclude that it is desirable to to have the flexibilty to choose from among several methods.

Summary of Results for Chapters 2, 4, and 5

(number of function evaluations)

Summary of Results : Unconstrained Optimization Methods

(number of function evaluations)

Moré, Garbow, and Hillstrom Test Problems

	n	m	DMNG		NPSOL		DMNH		MNA	
1.^{0}	2	2	40	42	57	57	32	32	15	15
2.^{0}	2	2	12^l	12^l	14^l	16^l	10^l	10^l	8	8
3.^{0}	2	2	217	220	–	–	130	132	129	129
4.^{0}	2	3	66	67	–	–	22	23	1	1
5.^{0}	2	3	16	17	23	24	11	12	47^s	47^s
6.	2	10	33	34	5^{sl}	5^{sl}	11	11	12	12
7.^{0}	3	3	28	30	38	40	16	17	12	12
8.	3	15	19	22	30	31	9	10	11	11
9.	3	15	8	12	8	8	4	4	3	3
10.	3	16	465	467	–	–	382	388	274	274
11.^{0}	3	10	4^l	327	2^l	2^l	290	292	532	532
12.^{0}	3	10	43	45	–	–	24	24	41	41
13.^{0}	4	4	62^t	89	86^t	86^t	27^t	38	23^t	23^t
14.^{0}	4	6	100	102	90	91	42	49	50	50
15.	4	11	35	36	33	35	11	12	20	20
16.	4	20	46	47	42	43	11	13	9	11
17.	5	33	69	72	–st	–st	46^s	47^s	43	44
18.^{0}	6	13	45	47	48	48	–	–	31	31
19.	11	65	69	72	81	83	23	24	7^s	8^s
20a.	6	31	35	37	53	54	15	15	15	15
20b.	9	31	76	79	94	98	20	22	15	15
20c.	12	31	89^l	148^l	86^l	251^l	24	24	15	15
20d.	20	31	110^l	134^l	76^l	200^l	50^l	~	–	–
21a.^{0}	10	10	120	125	138	139	25	26	15	15
21b.^{0}	20	20	189	193	210	218	27	27	15	15
22a.^{0}	12	12	143^t	235	133^t	171^t	28^t	40	23^t	23^t
22b.^{0}	20	20	187^t	344	193^t	232^t	29^t	40	24^t	24^t
23a.	4	5	77	78	73	75	42	43	47	47
23b.	10	11	80	81	212	225	44	45	46	46
24a.	4	8	364	472	21^l	920	126	128	159	159
24b.	10	20	475	632	350	368	158	162	133	133

Summary of Results : Nonlinear Least-Squares Methods

(number of function evaluations)

Moré, Garbow, and Hillstrom Test Problems

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
1.^{0}	34	34	22	22	14	14	34	34	34	34
2.^{0}	138	225	14^l	21^l	10^l	12^l	44^l	44^l	35^l	35^l
3.^{0}	31	44	19	19	64	65	49	49	45	45
4.^{0}	56	56	40^t	54	40^t	53	65	65	59	59
5.^{0}	8	8	9	10	9	10	14	14	10	10
6.	1180^s	–	21	28	14	16	52	52	36^s	36^s
7.^{0}	13	13	11	12	13	14	20	20	14	20
8.	7	7	6	7	7	8	14	14	6	6
9.	3	3	4	5	3	5	3	3	3	3
10.	–	30	126	126	132	133	18	18	17	17
11.^{0}	–	–	–	–	–	–	61^l	61^l	–	–
12.^{0}	7	7	7	8	8	9	12	12	8	12
13.^{0}	16^t	16^t	65	65	19^t	25	18^t	18^t	18^t	18^t
14.^{0}	96	96	70	70	52	53	114	114	93	99
15.	26	26	18	28	11	12	35	36	19	19
16.	3484	3484	264	356	21	22	44	44	33	33
17.	13	13	18	19	26	27	19	19	14	18
18.^{0}	–^{s}	–	46	46	45	46	75^s	75^s	115^s	115^s
19.	24	24	17	19	20	22	34	34	19	19
20a.	12	12	8	10	13	13	31	31	9	9
20b.	6	6	9	10	12	15	6	6	6	6
20c.	6	6	10	12	14	14	6	6	6	6
20d.	6^l	6	18	23	7^l	~	19	19	11	13
21a.^{0}	34	34	22	22	27	27	34	34	34	34
21b.^{0}	34	34	22	22	16	16	34	34	34	34
22a.^{0}	16^t	16^t	72	72	20^t	26	18^t	18^t	18^t	18^t
22b.^{0}	16^t	16^t	69	69	19^t	26	18^t	18^t	18^t	18^t
23a.	86	86	34	44	36	37	67	67	47	47
23b.	99	99	84	104	61	68	88	88	73	73
24a.	781	781	151	156	139	142	598	598	176	176
24b.	–	–	80	88	129	138	533^s	533^s	153	153

Summary of Results : Unconstrained Optimization Methods

(number of function evaluations)

Moré, Garbow, and Hillstrom Test Problems (continued)

	n	m	DMNG		NPSOL		DMNH		MNA	
25a.^{0}	10	12	20	21	26	26	15	15	14	14
25b.^{0}	20	22	25	26	31	32	18	19	17	18
26a.^{0}	10	10	34^l	37^l	37^l	39^l	11^l	12^l	22	23
26b.^{0}	20	20	62^l	65^l	86^l	95^l	20^l	20^l	27	27
27a.^{0}	10	10	13	16	19	20	9	10	22	22
27b.^{0}	20	20	15	18	18^s	21^s	11	12	30	30
28a.^{0}	10	10	31	34	30	33	4	4	4	4
28b.^{0}	20	20	60	64	54	60	4	4	4	4
29a.^{0}	10	10	8	10	7	8	4	5	4	4
29b.^{0}	20	20	8	10	7	8	4	5	4	4
30a.^{0}	10	10	51	57	42	44	6	7	7	7
30b.^{0}	20	20	65	88	61	62	6	7	7	7
31a.^{0}	10	10	46	60	47	49	9	9	9	9
31b.^{0}	20	20	47	63	76	78	9	9	9	9
32.^{L}	10	20	6	6	2	2	6	6	4	4
33.^{L}	10	20	4	4	4	4	5	5	27	27
34.^{L}	10	20	5	5	4	4	6	6	20	20
35a.	8	8	34	38	33	35	14	14	46	46
35b.^{0}	9	9	44	46	29	33	17	18	94	94
35c.	10	10	41^l	45^l	37^l	40^l	19	20	59^s	60^s

Matrix Square Root Test Problems

	n	m	DMNG		NPSOL		DMNH		MNA	
36a.^{0}	4	4	–	–	–^{t}	–^{t}	–	–	–	–
36b.^{0}	9	9	–	–	–^{t}	–^{t}	–	–	65^l	–
36c.^{0}	9	9	69	101	3	3	31	35	–	–
36d.^{0}	9	9	–	–	–^{t}	–^{t}	–	–	–	–

Summary of Results : Nonlinear Least-Squares Methods

(number of function evaluations)

Moré, Garbow, and Hillstrom Test Problems (continued)

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
25a.[0]	11	11	11	12	15	16	16	16	12	17
25b.[0]	13	13	13	14	19	19	18	18	14	19
26a.[0]	16	16	28[l]	37[l]	11	12	22	22	18	22
26b.[0]	25	25	57[l]	71[l]	39	42	29	29	25	29
27a.[0]	21	21	15	15	8	9	26[s]	26[s]	21[s]	21[s]
27b.[0]	22[s]	22[s]	5[l]	18	11	12	30[s]	30[s]	24[s]	24[s]
28a.[0]	4	4	5	5	4	4	4	4	4	4
28b.[0]	4	4	5	5	3	4	4	4	4	4
29a.[0]	4	4	5	5	4	4	10	10	6	6
29b.[0]	4	4	5	5	4	4	10	10	6	6
30a.[0]	6	6	6	7	8	9	11	11	7	11
30b.[0]	6	6	6	7	8	9	11	11	7	11
31a.[0]	7	7	7	8	10	11	12	12	8	12
31b.[0]	7	7	7	8	10	11	12	12	8	12
32.[z]	2	2	3	3	5	5	2	2	2	2
33.[z]	3	–	3	8	18	18	2	2	2	2
34.[z]	3	3	3	7	13	13	2	2	2	2
35a.	222	–[t]	40	53	23	24	87[s]	87[s]	99	99
35b.[0]	107	–[t]	12	13	11	11	42	42	38	42
35c.	–[t]	–[t]	25	34	17	19	85	85	58[s]	58[s]

Matrix Square Root Test Problems

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
36a.[0]	–	95	–	–	–	–	102	102	96	102
36b.[0]	4[l]	879	9	10	16[l]	–	–	–	1116[l]	1116[l]
36c.[0]	20	20	29	40	16	22	28	28	28	28
36d.[0]	2	2	2	2	4	4	2	2	2	2

Summary of Results : Unconstrained Optimization Methods

(number of function evaluations)

Hanson Test Problems

	n	m	DMNG		NPSOL		DMNH		MNA	
37.	2	16	22	22	16	18	16	17	6	6
38.	3	16	31	32	$-^{st}$	$-^{st}$	14	14	12	12

McKeown Test Problems

	n	m	DMNG		NPSOL		DMNH		MNA	
39a.	2	3	9	11	8	8	4	4	4	4
39b.	2	3	9	10	9	10	4	4	4	4
39c.	2	3	6	7	6	8	4	5	4	4
39d.	2	3	8	9	11	12	6	6	6	6
39e.	2	3	11	12	20	21	8	8	8	8
39f.	2	3	11	11	26	27	11	11	11	11
39g.	2	3	17	18	28	29	13	14	14	14
40a.	3	4	11	12	13	14	4	4	4	4
40b.	3	4	10	12	10	11	4	5	4	4
40c.	3	4	9	10	9	10	5	5	5	5
40d.	3	4	10	11	15	16	5	6	6	6
40e.	3	4	11	13	23	25	7	7	8	8
40f.	3	4	14	16	40	41	10	10	10	10
40g.	3	4	18	20	45	45	13	13	13	13
41a.	5	10	11	13	12	12	4	4	4	4
41b.	5	10	11	13	12	13	4	4	4	4
41c.	5	10	13	14	12	14	8	8	8	8
41d.	5	10	17	20	17	19	8	9	9	9
41e.	5	10	24	26	47	50	11	12	12	12
41f.	5	10	27	31	73	76	14	14	14	14
41g.	5	10	32	35	83	84	17	17	17	17

Summary of Results : Nonlinear Least-Squares Methods

(number of function evaluations)

Hanson Test Problems

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
37.	37	37	15	21	10	11	13	13	10	10
38.	31	31	18	28	10	12	17	17	13	13

McKeown Test Problems

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
39a.	8	8	5	6	5	5	9	9	4	4
39b.	10	10	14	21	6	7	17	17	6	6
39c.	23	23	18	25	7	8	11	11	9	9
39d.	699	699	20	28	7	8	15	15	12	12
39e.	1962	1962	28	44	9	10	15	15	12	12
39f.	–	–	31	44	14	15	27	27	24	24
39g.	–	–	39	44	18	20	38	38	39	39
40a.	13	13	6	9	7	7	12	12	5	5
40b.	16	16	14	17	7	11	19	19	6	6
40c.	381	381	16	22	9	10	27	27	11	11
40d.	2695	2695	26	40	9	9	18	18	13	13
40e.	–	–	90	146	10	11	51	51	45	45
40f.	–	–	180	272	13	14	96	98	79	79
40g.	–	–	206	319	23	25	103	103	87	87
41a.	5	5	4	4	4	4	8	8	4	4
41b.	6	6	4	5	4	5	16	16	4	4
41c.	12	12	6	8	6	6	19	19	5	5
41d.	31	31	15	22	9	11	27	27	9	9
41e.	154	154	29	38	17	20	35	35	14	14
41f.	812	812	57	89	24	27	44	44	16	16
41g.	2137	2137	84	144	29	30	48	48	21	21

Summary of Results : Unconstrained Optimization Methods

(number of function evaluations)

DeVilliers and Glasser Test Problems

	n	m	DMNG		NPSOL		DMNH		MNA	
42a.[0]	4	24	53	56	3[el]	3[el]	28	28	15[e]	15[e]
42b.[0]	4	24	103[e]	104[e]	3[el]	3[el]	35	36	16	16
42c.[0]	4	24	76	78	–[et]	–[et]	30	31	6	6
42d.[0]	4	24	61	64	–[et]	–[et]	30	30	6	6
43a.[0]	5	16	49	51	–[et]	–[et]	22	22	28[e]	28[e]
43b.[0]	5	16	58	60	–[et]	–[et]	26	27	24[e]	24[e]
43c.[0]	5	16	41	44	–[et]	–[et]	21	21	22[e]	22[e]
43d.[0]	5	16	57	60	–[et]	–[et]	27[e]	28[e]	41[e]	41[e]
43e.[0]	5	16	51	53	–[et]	–[et]	28[e]	29[e]	36	36
43f.[0]	5	16	45	48	–[et]	–[et]	17	18	87	87

Dennis, Gay, and Vu Test Problems

	n	m	DMNG		NPSOL		DMNH		MNA	
44a.[0]	6	6	441	444	379	388	179	180	181	181
44b.[0]	6	6	31	34	25	27	9	10	49	49
44c.[0]	6	6	3726	3731	–	–	194	195	908	909
44d.[0]	6	6	–[t]	3865	–	–	187	188	917	918
44e.[0]	6	6	–[t]	2815	3430	3586	219	220	501	502
45a.[0]	8	8	284	288	307	312	63	64	170	170
45b.[0]	8	8	36	40	40	41	15	16	31	31
45c.[0]	8	8	6197	6200	–	–	321	322	1380	1381
45d.[0]	8	8	7929	7934	–	–	328	329	1431	1432
45e.[0]	8	8	3341	3346	2821	3147	351	352	1512	1513

Summary of Results : Nonlinear Least-Squares Methods

(number of function evaluations)

DeVilliers and Glasser Test Problems

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
42a.^{0}	67^s	67^s	18	19	29^s	29^s	80^s	80^s	67^s	67^s
42b.^{0}	611^s	–	48^s	49^s	74^s	74^s	98^s	98^s	76^s	76^s
42c.^{0}	33^s	33^s	20^s	20^s	32^s	32^s	47^s	47^s	49^s	49^s
42d.^{0}	27^s	27^s	15^s	16^s	23	24	27^s	27^s	27^s	27^s
43a.^{0}	22^s	22^s	14^s	15^s	31	32	33^s	33^s	29^s	37^s
43b.^{0}	1167^{s,l}	1167^{s,l}	18^s	18^s	20	20	32^s	32^s	32^s	32^s
43c.^{0}	23^s	23^s	11^s	11^s	34^l	41^l	33^s	33^s	25^s	33^s
43d.^{0}	19^s	19^s	22^s	23^s	17	17	38^s	38^s	30^s	38^s
43e.^{0}	37^s	37^s	12^s	13^s	28	29	28^s	28^s	28^s	28^s
43f.^{0}	20^s	20^s	12	13	20	20	47	47	39	47

Dennis, Gay, and Vu Test Problems

	G-N		LMDER		DN2G		LSQFDQ		LSQSDN	
44a.^{0}	125	125	37	38	58	59	104	104	100	100
44b.^{0}	5	5	5	6	7	7	10	10	6	10
44c.^{0}	52	52	108	109	93	94	63	63	59	59
44d.^{0}	36	36	98	99	97	98	41	41	41	41
44e.^{0}	70	70	82	83	83	83	83	83	79	83
45a.^{0}	125	125	47	48	65	66	104	104	100	100
45b.^{0}	5	5	5	6	8	8	12	12	6	12
45c.^{0}	52	52	164	165	129	130	63	63	59	59
45d.^{0}	36	36	144	145	168	168	45	45	43	43
45e.^{0}	70	70	130	131	173	173	85	85	79	81

5.8 Bibliography

Al-Baali, M., "Methods for Nonlinear Least Squares", Ph. D. Thesis, Department of Mathematical Sciences, University of Dundee (1984).

Al-Baali, M., and R. Fletcher, "Variational Methods for Non-Linear Least Squares", *Journal of the Operational Research Society*, Vol. 36 No. 5 (May 1985) 405–421.

Al-Baali, M., and R. Fletcher, "An Efficient Line Search for Nonlinear Least Squares", *Journal of Optimization Theory and Applications*, Vol. 48 No. 3 (March 1986) 359–377.

Bard, Y., "Comparison of Gradient Methods for the Solution of Parameter Estimation Problems", SIAM *Journal on Numerical Analysis*, Vol. 7 No. 1 (March 1970) 157–186.

Bartels, R. H., G. H. Golub, and M. A. Saunders, "Numerical Techniques in Mathematical Programming", in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter (eds.), Academic Press (1970) 123–176.

Bartholomew-Biggs, M. C., "The Estimation of the Hessian Matrix in Nonlinear Least Squares Problems with Non-Zero Residuals", *Mathematical Programming*, 12 (1977) 67–80.

Betts, J. T., "Solving the Nonlinear Least Square Problem : Application of a General Method", *Journal of Optimization Theory and Applications*, Vol. 18 (1976) 469–484.

Brown, K. M., and J. E. Dennis, "A New Algorithm for Nonlinear Least Squares Curve Fitting", in *Mathematical Software*, J. R. Rice (ed.), Academic Press (1971).

Cornwell, L. W., M. G. Kocman, and M. F. Prosser, "Computational Experience with Davidon's Least-Square Algorithm", *Journal of Optimization Theory and Applications*, Vol. 31 No. 1 (May 1980) 27–40.

Davidon, W. C., "Optimally Conditioned Optimization Algorithms without Line Searches", *Mathematical Programming* 9 (1975) 1–30.

Davidon, W. C., "New Least Square Algorithms", *Journal of Optimization Theory and Applications*, Vol. 18 (1976) 187–197.

Dennis, J. E., "Some Computational Techniques for the Nonlinear Least Squares Problem", in *Numerical Solution of Systems of Nonlinear Algebraic Equations*, G. D. Byrne and C. A. Hall (eds.), Academic Press (1973) 157–183.

Dennis, J. E., "A Brief Survey of Convergence Results for Quasi-Newton Methods", in *Nonlinear Programming*, SIAM-AMS Proceedings Vol. IX, American Mathematical Society (1976) 185–199.

Dennis, J. E., and J. J. Moré, "Quasi-Newton Methods : Motivation and Theory", SIAM *Review*, Vol. 19 No. 1 (January 1977) 46–89.

Dennis, J. E., and R. E. Welsch, "Techniques for Nonlinear Least Squares and Robust Regression", *Communications in Statistics — Simulation and Computation*, B7 (1978) 345–359.

Dennis, J. E., D. M. Gay, and R. E. Welsch, "An Adaptive Nonlinear Least-Squares Algorithm", ACM *Transactions on Mathematical Software*, Vol. 7 No. 3 (September 1981a) 348–368.

Dennis, J. E., D. M. Gay, and R. E. Welsch, "ALGORITHM 573 NL2SOL : An Adaptive Nonlinear Least-Squares Algorithm", ACM *Transactions on Mathematical Software*, Vol. 7 No. 3 (September 1981b) 369–383.

Dennis, J. E., and H. F. Walker, "Convergence Theorems for Least-Change Secant Update Methods", SIAM *Journal on Numerical Analysis*, Vol. 18 No. 6 (December 1981) 949–987.

Dennis, J. E., D. M. Gay, and R. E. Welsch, "Remark on ALGORITHM 573", ACM *Transactions on Mathematical Software*, Vol. 9 No. 1 (March 1983) 139.

Dennis, J. E., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall (1983).

DeVilliers, N., and D. Glasser, "A Continuation Method for Nonlinear Regression", SIAM *Journal on Numerical Analysis*, Vol. 18 No. 6 (December 1981) 1139–1154.

Fletcher, R., "A Modified Marquardt Subroutine for Non-Linear Least Squares", Technical Report AERE- R 6799, Atomic Energy Research Establishment, Harwell (1971).

Fletcher, R., *Practical Methods of Optimization : Volume I, Unconstrained Optimization*, Wiley (1980).

Fletcher, R., and C. Xu, "Hybrid Methods for Nonlinear Least Squares", Technical Report NA/92, Numerical Analysis Report, University of Dundee, Department of Mathematical Sciences (December 1985; revised July 1986).

Gay, D. M., "ALGORITHM 611 : Subroutines for Unconstrained Minimization Using a Model/Trust-Region Approach", ACM *Transactions on Mathematical Software*, Vol. 9 No. 4 (December 1983) 503–524.

Gill, P. E., and W. Murray, "Safeguarded Steplength Algorithms for Optimization using Descent Methods", Technical Report NAC 37, National Physical Laboratory, England.

Gill, P. E., and W. Murray, "Nonlinear Least Squares and Nonlinearly Constrained Optimization", in *Numerical Analysis — Proceedings Dundee 1975, Lecture Notes in Mathematics*, Vol. 506, Springer-Verlag (1976) 134–147.

Gill, P. E., and W. Murray, "Algorithms for the Solution of the Nonlinear Least-Squares Problem", SIAM *Journal on Numerical Analysis*, Vol. 15 No. 5 (October 1978) 977–992.

Gill, P. E., and W. Murray, "Algorithms for the Solution of the Nonlinear Least-Squares Problem", SIAM *Journal on Numerical Analysis*, Vol. 15 No. 5 (October 1978) 977–992.

Gill, P. E., W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press (1981).

Hebden, M. D., "An Algorithm for Minimization using Exact Second Derivatives", Technical Report T. P. 515, Atomic Energy Research Establishment, Harwell (1973).

Jones, A., "Spiral — A new Algorithm for Non-linear Parameter Estimation using Least Squares", *Computer Journal*, Vol. 13 No. 3 (August 1970) 301–308.

Levenberg, K., "An Method for the Solution of Certain Nonlinear Problems in Least Squares", *Quarterly of Applied Mathematics*, Vol. 2 (1944) 164–168.

Lindström, P., and P.-Å. Wedin, "A New Linesearch Algorithm for Nonlinear Least Squares Problems", *Mathematical Programming*, Vol. 29 (1984) 268–296.

Marquardt, D. W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters", *Journal of the Institute of Mathematics and its Applications*, Vol. 11 No. 2 (June 1963) 431–441.

Meyer, R. R., "Theoretical and Computational Aspects of Nonlinear Regression", in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter (eds.), Academic Press (1970) 465–486.

Moré, J. J., "The Levenberg-Marquardt Algorithm: Implementation and Theory", in *Numerical Analysis — Proceedings Dundee 1977, Lecture Notes in Mathematics*, Vol. 630, Springer-Verlag (1978) 105–116.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, " User Guide for MINPACK-1", Technical Report ANL-80-74, Argonne National Laboratory (1980).

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization Software", ACM *Transactions on Mathematical Software*, Vol. 7 No. 1 (March 1981) 17–41.

Moré, J. J., "Recent Developments in Algorithms and Software for Trust Region Methods", in *Mathematical Programming—The State of the Art—Bonn 1982*, A. Bachem, M. Grötschel, and B. Korte (eds.), Springer-Verlag (1983) 258–287.

Morrison, D. D., "Methods for Nonlinear Least Squares Problems and Convergence Proofs, Tracking Programs and Orbital Determinations", *Procedings of the Jet Propulsion Laboratiry Seminar* (1960) 1–9.

NAG Fortran Library Manual, Vol. 3, Numerical Algorithms Group, Oxford U. K., and Downers Grove, IL U. S. A. (January 1984).

Nazareth, L., "Some Recent Approaches to Solving Large Residual Nonlinear Least Squares Problems", SIAM *Review*, Vol. 22 No. 1 (January 1980) 1–11.

Nazareth, L., "An Adaptive Method for Minimzing a Sum of Squares of Nonlinear Functions", IIASA Report WP-83-99 (1983).

Oren, S. S., "On the Selection of Parameters in Self-Scaling Variable Metric Algorithms", *Mathematical Programming*, 7 (1974) 351–367.

Osborne, M. R., "Some Aspects of Non-linear Least Squares Calculations", in *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma (ed.), Academic Press (1972) 171–189.

Osborne, M. R., "Nonlinear Least Squares — the Levenberg Algorithm Revisted", *Journal of the Australian Mathematical Society* 19, Series B (1976) 343–357.

PORT Mathematical Software Library, A. T. & T. Bell Laboratories, Murray Hill, NJ U. S. A. (May 1984).

Powell, M. J. D., "A Hybrid Method for Nonlinear Equations", in *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz (ed.), Gordon and Breach (1970a) 87–114.

Powell, M. J. D., "A New Algorithm for Unconstrained Optimization", in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter (eds.) (1970b) 31–65.

Powell, M. J. D., "Convergence Properties of a Class of Minimization Algorithms", in *Nonlinear Programming 2*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), Academic Press (1975) 1–27.

Ramsin, H., and P-Å. Wedin, "A Comparison of some Algorithms for the Nonlinear Least Squares Problem", *BIT* 17 No. 1 (1977) 72–90.

Reinsch, C. H., "Smoothing by Spline Functions II", *Numerische Mathematik* 16 (1971) 451–454.

Ruhe, A., "Accelerated Gauss-Newton Algorithms for Nonlinear Least Squares Problems", *BIT*, Vol. 19 (1979) 356–367.

Ruhe, A., and P-Å. Wedin, "Algorithms for Separable Least Squares Problems", SIAM *Review*, Vol. 22 No. 3 (July 1980) 318–337.

Salane, D. E., "A Continuation Approach for Solving Large Residual Nonlinear Least Squares Problems", Sandia National Laboratories, Albuquerque, NM (1986) [to appear in SIAM *Journal on Scientific and Statistical Computing*].

Shanno, D. F., "Parameter Selection for Modified Newton Methods for Function Minimization", SIAM *Journal on Numerical Analysis*, Vol. 7 No. 3 (September 1970) 366–372.

Steen, N. M., and G. D. Byrne, "The Problem of Minimizing Nonlinear Functionals", in *Numerical Solution of Systems of Nonlinear Algebraic Equations*, G. D. Byrne and C. A. Hall (eds.), Academic Press (1973) 185–239.

Varah, J. M., "On Fitting Exponentials by Nonlinear Least Squares", SIAM *Journal on Numerical Analysis*, Vol. 6 No. 1 (January 1986) 30–44.

Wedin, P.-Å., "On the Gauss-Newton Method for the Non-Linear Least Squares Problem", Technical Report 24, The Swedish Institute of Applied Mathematics (1974).

6. Sequential Quadratic Programming (SQP) Methods

6.1 Overview

This chapter investigates some new methods for nonlinear least squares that find a search direction by solving a quadratic program (QP) of the form

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p \tag{6.1.1}$$

subject to

$$-b^L \leq Ap + c \leq b^U,$$

where

$$b^L \geq 0 \quad \text{and} \quad b^U \geq 0.$$

Recall that $\bar{g}$ denotes the gradient of the nonlinear least-squares objective :

$$\bar{g} \equiv \nabla \left(\frac{1}{2} f^T f \right) = J^T f.$$

The matrix H approximates the Hessian matrix of the nonlinear least-squares objective :

$$H \approx \nabla^2 \left(\frac{1}{2} f^T f \right) = J^T J + \sum_{i=1}^{m} \phi_i \nabla^2 \phi_i, \tag{6.1.2}$$

where ϕ_i is the ith component of the vector f.

In all cases we shall consider, the vector c in (6.1.1) is related to f, and the matrix A is related to the Jacobian J of f. In Chapter 2, we described algorithms for unconstrained optimization in which search directions minimize a quadratic function. For each iteration, these methods compute an approximation H to the Hessian matrix and solve

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p. \tag{6.1.3}$$

It was explained in Chapter 4 that nonlinear least-squares problems are distinguished from other unconstrained optimization problems in that some curvature information is available from the first derivatives of the residual functions (see (6.1.2)). In Chapters 4 and 5, we saw

that algorithms for nonlinear least squares typically attempt to exploit this feature by using special approximations for H in (6.1.3) that are based on the structure of the nonlinear least-squares Hessian (6.1.2). Our motivation for (6.1.1) is to introduce either an estimate or explicit information about the second derivatives of f by building a model for the curvature of $f^T f$ that separates the contribution of first derivatives of the residual functions from the Hessian of the sum of squares. Rather than propose alternatives for H, we approximate the full Hessian and include the additional information as constraints in the QP subproblem. We shall investigate options that are based on convergence properties of sequential quadratic programming (SQP) methods for constrained optimization, and on geometric considerations in nonlinear least squares.

Some background material on quadratic programs is reviewed in Section 6.2. In Section 6.3, we motivate the SQP approach to nonlinear least squares through the relationship between nonlinear least squares and nonlinearly constrained optimization. Section 6.4 discusses types of constraints that are consistent with our motivations. These include constraints based on information about the individual residuals, as well as constraints derived from the QR factorization of the Jacobian of f. Two different algorithmic frameworks that incorporate (6.1.1) are then presented in Section 6.5. In both approaches, a tentative set of constraints is formulated at the beginning of an iteration, after which the set is modified if necessary to take into account feasibility and restrictions on the size of the search direction. The first approach modifies the given set through the addition of perturbations to the constraints. The perturbations are defined by a special QP subproblem. Among other possibilities, this strategy leads to a generalization of Levenberg-Marquardt methods (see Section 5.2). The second approach uses the QP to select from among the constraints in the given set. Several possible algorithms are suggested, including a corrected Gauss-Newton method (see Section 5.4). Numerical examples are given thoughout Section 6.5, because — as we have seen in previous chapters — it is not possible to draw conclusions about the performance of a method without observing its behavior on a variety of problem types. Details of the numerical tests are given at the end of Section 6.5. Conclusions and suggestions as to how these methods might be extended are given in the final section.

6.1.1 Abbreviations

The following abbreviations are used throughout this chapter :

QP - quadratic program, or quadratic programming

SQP - sequential quadratic programming

6.2 Quadratic Programs

This section summarizes information about quadratic programs that is relevant to formulating SQP algorithms for nonlinear least squares. First, optimality conditions for quadratic programs are given, together with conditions sufficient to guarantee uniqueness of a minimum. For simplicity, equality-constrained and inequality-constrained quadratic programs are treated separately, but it is straightforward to generalize to cases that include both types of constraint. Next, we give a theorem stating a sufficient condition for the SQP search diirection to be a descent direction for the nonlinear least-squares objective. The section ends with a list of references concerning algorithms and software for quadratic programs.

6.2.1 Theoretical Properties

6.2.1.1 Equality Constrained Quadratic Programs

For an equality-constrained quadratic program (EQP)

$$\min_{x \in \Re^n} \mathcal{Q}(x) \equiv q^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} Q x \qquad (6.2.1)$$
$$\text{subject to} \quad Ax = b,$$

necessary conditions for optimality at $\hat{x}$ (see, for example, Chapter 9 of Fletcher [1981]) are

(1) $A\hat{x} = b$,

(2) $\nabla \mathcal{Q}(\hat{x}) = Q\hat{x} + q \in \mathcal{R}(A^{\mathrm{T}})$, or, equivalently, $Z^{\mathrm{T}}(Q\hat{x} + q) = 0$,

(3) $Z^{\mathrm{T}} Q Z$ is positive semi-definite,

where Z is any matrix whose columns form a basis for $\mathcal{N}(A)$. Condition (2) says that, at a solution $\hat{x}$, there is a vector of Lagrange multipliers $\hat{\lambda}$ such that $A^{\mathrm{T}}\hat{\lambda} = Q\hat{x} + q$. Hence conditions (1) and (2) imply that $(\hat{x}, \hat{\lambda})$ must be a solution to the system of equations

$$\begin{pmatrix} Q & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ -\lambda \end{pmatrix} = \begin{pmatrix} -q \\ b \end{pmatrix}. \tag{6.2.2}$$

The following theorem establishes the conditions under which (6.2.2) has a unique solution. (For a proof, as well as a more extensive treatment of optimality conditions for EQP, see Gould [1985]).

Theorem 6.2-1:

Define $M = \begin{pmatrix} Q & A^{\mathrm{T}} \\ A & 0 \end{pmatrix}$, *and let* Z *be a matrix whose columns are a basis for the null space of* A. *Then* M *is nonsingular if and only if* A *has full row rank and* $Z^{\mathrm{T}}QZ$ *is nonsingular.* ∎

In order for (6.2.1) to have a strong (global) minimum, it is sufficient for A to have full row rank, and $Z^{\mathrm{T}}QZ$ to be positive definite. Moreover, the necessary conditions do not imply uniqueness in x for (6.2.2). The vector $x_R = Yx_Y$ is completely determined by the equations $Ax = b$, where Y is a matrix whose columns form a basis for $\mathcal{R}(A^{\mathrm{T}})$. If A has row rank $< n$, then the set of vectors $x_R \in \mathcal{R}(A^{\mathrm{T}})$ satisfing the constraints is either an infinite set, or it is empty. The null-space component $x_N = Zx_Z = \hat{x} - Yx_Y$ minimizes

$$\left(q + x_Y^{\mathrm{T}}Y^{\mathrm{T}}Q\right)^{\mathrm{T}} Zx_Z + \frac{1}{2}x_Z^{\mathrm{T}}Z^{\mathrm{T}}QZx_Z$$

as a function of x_Z, and is found by solving the system of equations

$$Z^{\mathrm{T}}QZx_Z = -Z^{\mathrm{T}}q - Z^{\mathrm{T}}QYx_Y. \tag{6.2.3}$$

Hence x_N is completely determined by (6.2.3) whenever $Z^{\mathrm{T}}QZ$ is nonsingular. Moreover, if Q is positive definite and $Ax = b$ has a solution, then the solution of (6.2.2) is unique in x (but not in λ if A is row rank deficient).

6.2.1.2 Inequality Constrained Quadratic Programs

The situation for an inequality-constrained quadratic program (IQP)

$$\min_{x \in \mathbb{R}^n} \mathcal{Q}(x) \equiv q^T x + \frac{1}{2} x^T Q x \tag{6.2.4}$$

$$\text{subject to} \quad Ax \geq b$$

is somewhat more complicated than for an EQP. If $\hat{A}x = \hat{b}$ is the set of constraints that hold as equalities at a point $\hat{x}$, then necessary conditions for $\hat{x}$ to be a local minimum (see, for example, Chapter 9 of Fletcher [1981]) are

(1) $A\hat{x} \geq b$

(2) $\nabla \mathcal{Q}(\hat{x}) = Q\hat{x} + q = \hat{A}^T \hat{\lambda}$, with $\hat{\lambda} \geq 0$.

(3) $Z^T Q Z$ is positive semi-definite,

where Z is any matrix whose columns form a basis for the null space of $\hat{A}$. Solutions must satisfy an augmented system

$$\begin{pmatrix} Q & \hat{A}^T \\ \hat{A} & 0 \end{pmatrix} \begin{pmatrix} x \\ -\lambda \end{pmatrix} = \begin{pmatrix} -q \\ \hat{b} \end{pmatrix}, \tag{6.2.5}$$

which are the necessary conditions for a local minimum of the equality constrained quadratic program

$$\min_{x \in \mathbb{R}^n} \mathcal{Q}(x) \equiv q^T x + \frac{1}{2} x^T Q x$$

$$\text{subject to} \quad \hat{A}x = \hat{b}.$$

If Q is positive definite and the rows of A are linearly independent, then (6.2.4) is convex, and has a unique minimum. Finding a global minimum for a general quadratic program with inequality constraints is a combinatorial problem, because several nonsingular systems of the form (6.2.5) may be possible for a single QP. Practical algorithms for general quadratic programming therefore seek only a local minimum.

6.2.2 A Sufficient Condition for Descent

In formulating sequential quadratic programming methods for nonlinear least squares, we shall be interested in producing search directions p that are descent directions for $f^{\mathrm{T}}f$, that is, directions that satisfy $\bar{g}^{\mathrm{T}}p < 0$. The following theorem gives a sufficient condition for descent in the SQP algorithms :

Theorem 6.2-2 (sufficient condition for descent) :

Consider the quadratic program

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}Hp \tag{6.2.6}$$

subject to

$$-b^L \leq Ap + c \leq b^U,$$

where

$$b^L \geq 0 \quad \text{and} \quad b^U \geq 0.$$

If H is positive semi-definite, and if the feasible region includes a vector $\tilde{p}$ such that
(1) $\bar{g}^{\mathrm{T}}\tilde{p} < 0$,
and
(2) $\gamma\tilde{p}$ *is feasible for all* $\gamma \in (0, 1]$,
then solutions p_* to (6.2.6) *satisfy* $\bar{g}^{\mathrm{T}}p_* < 0$.

Proof:

Define

$$\bar{Q}(p) \equiv \bar{g}^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}Hp.$$

The hypotheses of the theorem imply the existence of a feasible vector p for which $\bar{Q}(p)$ is negative. Specifically, $\gamma > 0$ may be chosen sufficiently small so that $\bar{Q}(\gamma\tilde{p}) = \gamma\bar{g}^{\mathrm{T}}\tilde{p} + \frac{1}{2}\gamma^2\tilde{p}^{\mathrm{T}}H\tilde{p} < 0$. If p_* solves (6.2.6), then $\bar{Q}(p_*)$ must be at least as small as $\bar{Q}(\gamma\tilde{p})$. Since $p_*^{\mathrm{T}}Hp_* \geq 0$, it follows that $\bar{g}^{\mathrm{T}}p_* < 0$. ∎

An immediate corollary is that if $p = 0$ is an interior point of the feasible region, then a solution p_* will satisfy $\bar{g}^T p_* < 0$. This result will be used in defining constraint regions for SQP algorithms in later sections.

6.2.3 Algorithms and Software for Quadratic Programming

Some general discussion of quadratic programming is given in the texts by Fletcher [1981] and Gill, Murray, and Wright [1981]. Computational methods for quadratic programming are surveyed in Fletcher [1986]. Algorithms for the convex case can be found in Stoer [1971], Schittkowski and Stoer [1979], Sacher [1980], Han [1981], Haskell and Hanson [1981], Powell [1981], Goldfarb and Idnani [1983], Best [1984], Gill et al. [1984], Gill et al. [1986a], and Hanson [1986]. The software packages `LSEI` and `WNNLS` [Hanson and Haskell (1982)], and `LSSOL` [Gill et al. (1986a)], which use least-squares techniques [see Stoer (1971)], and `ZQPCVX` [Powell (1983b); (1985)], based on the method of Goldfarb and Idnani, are available for convex quadratic programming. For non-convex quadratic programming algorithms, see Gill and Murray [1978b], Benveniste [1979], Bunch and Kaufman [1980], and Hoyle [1986]. Software for the general case includes `QPSOL` [Gill et al. (1984)], a revised version of the Gill and Murray algorithm, and `IQP` [PORT (1984)], based on the method of Bunch and Kaufman.

6.3 Nonlinear Least Squares and Nonlinearly Constrained Optimization

In order to motivate the use of search directions based on QP subproblems with constraints, we describe some relationships between optimization subject to nonlinear constraints and nonlinear least-squares problems. We show that the set of applicable algorithms for nonlinear least squares can be expanded to include SQP methods related to those for general nonlinear programming, and explain why it may be advantageous to do so.

Consider the nonlinear programming problem

$$\min_{x \in \Re^n} \mathcal{F}(x) \tag{6.3.1}$$

subject to

$$c_E(x) = 0$$

$$c_I(x) \geq 0,$$

where it is assumed that $\mathcal{F}$, c_E, and c_I are smooth functions. Near a solution x^*, SQP algorithms for (6.3.1) solve either equality-constrained subproblems of the form

$$\min_{p \in \Re^n} \nabla \mathcal{F}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \mathcal{H} p \tag{6.3.2}$$

$$\text{subject to} \quad (\nabla \hat{c}) p = -\hat{c},$$

where $\hat{c}(x)$ is the vector of constraints that hold as equalities at x^*, or else they solve subproblems of the form

$$\min_{p \in \Re^n} \nabla \mathcal{F}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \mathcal{H} p \tag{6.3.3}$$

subject to

$$(\nabla c_E) p = -c_E$$

$$(\nabla c_I) p \geq -c_I,$$

that include inequality constraints. The matrix $\mathcal{H}$ approximates the Hessian (as a function of x) of the Lagrangian function

$$\mathcal{L}(x, \lambda^*) \equiv \mathcal{F}(x) - \hat{c}(x)^{\mathrm{T}} \lambda^* \tag{6.3.4}$$

in $\mathcal{N}(\nabla \hat{c}(x))$. The vector λ^* in (6.3.4) is the vector of Lagrange multipliers at the solution, and satisfies the relation

$$\nabla \mathcal{F}(x^*) = \nabla \hat{c}(x^*)^{\mathrm{T}} \lambda^*, \tag{6.3.5}$$

which is a necessary condition for a minimum at x^* when $\nabla \hat{c}(x^*)$ has full row rank (see, for example, Gill, Murray, and Wright [1981], Chapters 3 and 6). SQP methods based on (6.3.2) or (6.3.3) are superlinearly convergent whenever $\nabla \hat{c}(x^*)$ has full row rank, the projected Hessian of $\mathcal{L}$ in $\mathcal{N}(\nabla \hat{c})$ is positive definite at x^*, and the projection in $\mathcal{N}(\nabla \hat{c})$

of the approximation $\mathcal{H}$ is suffciently close to that of the exact Hessian of the Lagrangian function in a finite neighborhood of x^* (see, for example, Nocedal and Overton [1985]). Away from a solution, (6.3.2) and (6.3.3) may need to be modified to take into account infeasibility of the linearized constraints, and the need for the QP search direction to be a direction of descent for a merit function that reflects the aims of minimizing the objective and satisfying the constraints (see Murray and Wright [1982]). Recent references with extensive bibliographies on SQP algorithms for nonlinear programming include Gill et al. [1985, 1986c], Nocedal and Overton [1985], Stoer [1985], and Gurwitz [1986].

There are many ways to recast the nonlinear least-squares problem

$$\min_{x \in \Re^n} \frac{1}{2} f^{\mathrm{T}} f \tag{6.3.6}$$

$$f : \Re^n \rightarrow \Re^m,$$

as a constrained optimization problem. Given a solution x^* to (6.3.6), the following formulation subsumes a number of possibilities:

$$\min_{x \in \Re^n} \frac{1}{2} f^{\mathrm{T}} f, \tag{6.3.7}$$

subject to

$$\phi_i(x) = 0, \quad i \in \mathcal{E} \subseteq \mathcal{E}^*$$

$$\phi_i(x) \leq 0, \quad i \in \mathcal{I}_{\leq} \subseteq \mathcal{I}_{\leq}^*$$

$$\phi_i(x) \geq 0, \quad i \in \mathcal{I}_{\geq} \subseteq \mathcal{I}_{\geq}^*$$

where

$$\mathcal{E}^* \equiv \{i \mid \phi_i(x^*) = 0\}$$

$$\mathcal{I}_{\leq}^* \equiv \{i \mid \phi_i(x^*) \leq 0\}$$

$$\mathcal{I}_{\geq}^* \equiv \{i \mid \phi_i(x^*) \geq 0\}.$$

The objective in (6.3.7) need not necessarily include all of the residuals of f, but need only be a sum of squares of a subset of the components of f that includes the residuals not represented in $\mathcal{E}$. The QP subproblem associated with (6.3.7) would be

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p \tag{6.3.8}$$

$$(\nabla \phi_i)^T p = -\phi_i, \quad i \in \tilde{\mathcal{E}}$$

$$(\nabla \phi_i)^T p \leq -\phi_i, \quad i \in \tilde{\mathcal{I}}_{\leq}$$

$$(\nabla \phi_i)^T p \geq -\phi_i, \quad i \in \tilde{\mathcal{I}}_{\geq}$$

The sets of $\tilde{\mathcal{E}}$, $\tilde{\mathcal{I}}_{\leq}$, $\tilde{\mathcal{I}}_{\geq}$, and $\tilde{\mathcal{I}}$ of residuals defining each type of constraint may vary from iteration to iteration, and may be unrelated to $\mathcal{E}^*$, $\mathcal{I}^*_{\leq}$, and $\mathcal{I}^*_{\geq}$ away from a solution (see the discussion of QP constraints in the next section). The uncertainty in the form of the QP at points that are not especially close to a minimum is similar to the situation in SQP methods for nonlinearly constrained optimization. Asymptotic estimates of λ^* must be used in order to approximate the Hessian of the Lagrangian by $\mathcal{H}$ in (6.3.2) and (6.3.3). Moreover, when inequality constraints are present, $\hat{c}$ is usually unknown at nonoptimal points, which affects $\mathcal{H}$, as well as the QP constraints in methods that solve only equality-constrained subproblems (see Murray and Wright [1982] for a discussion of QP subproblems in SQP methods for constrained optimization). A fundamental aim of any scheme for formulating QP subproblems in an SQP method is that it should satisfy conditions for superlinear convergence near a solution. Usually this implies that the correct set of active constraints must be identified by the QP in a finite neighborhood of a minimum and that the active constraint gradients at a minimum must be linearly independent. In the case of (6.3.8), it suffices for H to approximate the Hessian matrix of the objective, because the vector of Lagrange multipliers is the zero vector, so that the Hessian of the Lagrangian and the Hessian of the objective in (6.3.7) are identical at a solution. Although convergence results that require nonzero Lagrange multipliers (strict complementarity) at a solution do not apply to (6.3.7) (see, for example, Robinson [1974]) the analysis of Nocedal and Overton [1985] implies that SQP methods for (6.3.7) are superlinearly convergent with QP subproblems of the form (6.3.8), provided the Jacobian matrix $\hat{J}$ of the active constraints has full row rank at a solution, and the projected Hessian of the objective in (6.3.7) is positive definite in $\mathcal{N}(\hat{J})$.

One reason to consider using methods that find search directions by solving more general quadratic programs (as opposed to minimizing quadratic functions) is that there is the

possibility of improvement in the asymptotic rate of convergence. If $Z(x)$ denotes the space orthogonal to the active constraint normals at x, then for superlinear convergence of an SQP method it is sufficient for the Hessian of $f^{\mathrm{T}}f$ to be positive definite only in $Z(x^*)$, provided the active constraint normals are linearly independent at x^*, and that the approximation to the Hessian of the Lagrangian is sufficiently good in $Z(x)$ in some finite neighborhood of a minimum. Most of the other methods we have discussed for nonlinear least squares require that the full Hessian be positive definite in order to achieve superlinear convergence. A possible exception is the class of corrected Gauss-Newton methods (see the discussion below).

As an example, consider an underdetermined nonlinear least-squares problem ($m < n$). This is a zero-residual problem in which the Hessian matrix is singular at a minimum, because there are fewer rows than columns in J. An application of such problems is in finding feasible points for nonlinear equality constraints. An equivalent constrained optimization problem is

$$\min_{x \in \Re^n} \frac{1}{2} f^{\mathrm{T}} f \tag{6.3.9}$$

$$\text{subject to} \quad \hat{f}(x) = 0,$$

where $\hat{f}$ is any subvector of f. Search directions in an SQP method for (6.3.9) solve subproblems of the form

$$\min_{x \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p \tag{6.3.10}$$

$$(\nabla \hat{f})^{\mathrm{T}} p = -\hat{f}$$

in the vicinity of a solution, and are superlinearly convergent when $\nabla \hat{f}$ has full row rank, $\nabla^2 \left(f^{\mathrm{T}} f\right)$ is positive definite in $\mathcal{N}(\nabla \hat{f})$, and H is a sufficiently close approximation to the Hessian of the nonlinear least-squares objective in $\mathcal{N}(\nabla \hat{f})$. If $\hat{f} = f$, then corrected Gauss-Newton methods (see Section 5.3) with $grade(J) = n$ solve QP subproblems of the form (6.3.10), so that they are potentially superlinearly convergent.

A related example views the nonlinear programming problem (6.3.1) as a nonlinear least-squares problem. Suppose that x^* is a solution to (6.3.1), and that $\hat{c}(x)$ is the vector of constraints that hold as equalities at x^*. Then x^* solves the nonlinear least-squares problem

$$\min_{x \in \Re^n} (\mathcal{F}(x) - \mathcal{F}^*)^2 + \hat{c}(x)^{\mathrm{T}} \hat{c}(x), \tag{6.3.11}$$

$$\mathcal{F}^* \equiv \mathcal{F}(x^*).$$

Algorithms for the solution of (6.3.1) that are based on (6.3.11) have been proposed by Morrison [1968] and Gill and Murray [1976] for the case when only equality constraints are present. Bartholomew-Biggs [1981] discusses the use of (6.3.11) as a merit function for sequential quadratic programming (SQP) methods for general nonlinear programming. Typically there would be fewer than n active constraints at a solution to (6.3.1) (otherwise the problem is overdetermined). If there are fewer than $n-1$ constraints, the Jacobian matrix

$$\mathcal{J}(x) \equiv \nabla \begin{pmatrix} \mathcal{F}(x) - \mathcal{F}^* \\ \hat{c}(x) \end{pmatrix} = \begin{pmatrix} \nabla \mathcal{F}(x)^{\mathrm{T}} \\ \nabla \hat{c}(x) \end{pmatrix}$$

cannot have full column rank because there are fewer rows than columns in the matrix. In any case $\mathcal{J}(x^*)$ has linearly dependent columns, because optimality conditions for (6.3.1) imply that $\nabla \mathcal{F}(x^*) \in \mathcal{R}(\nabla \hat{c}(x^*)^{\mathrm{T}})$ when $\nabla \hat{c}(x^*)$ has full row rank (see, for example, Gill, Murray, and Wright [1981], Chapter 3). For nearly all of the QP-based methods discussed in earlier chapters, the fact that the Hessian matrix of the objective in (6.3.11) is singular at a solution precludes superlinear convergence. For the corrected Gauss-Newton methods (Section 5.3), numerical tests show only linear convergence with $grade(\mathcal{J}) = rank(\mathcal{J})$ near a solution.

The splitting of the search direction into two orthogonal components that is allowed by SQP methods has potential computational advantages beyond favorable asymptotic convergence. This is true for nonlinear programming as well as for nonlinear least-squares, although the two cases are somewhat different, as explained below.

Before special techniques for handling linear constraints were available, the prevailing methods for nonlinear programming were based on transforming the constrained problem into a sequence of unconstrained problems (see, for example, Fiacco and McCormick [1968], Bertsekas [1982], and Fletcher [1983]). For example, augmented Lagrangian methods use such an approach. The augmented Lagrangian function is given by

$$\bar{\mathcal{L}}(x, \lambda^*, \rho) \equiv \mathcal{F}(x) - \hat{c}(x)^{\mathrm{T}} \lambda^* + \rho \hat{c}(x)^{\mathrm{T}} \hat{c}(x), \tag{6.3.12}$$

where $\hat{c}$ is the vector of active constraints at a minimum x^*, and λ^* is the corresponding vector of Lagrange multipliers. The function $\bar{\mathcal{L}}$ has a stationary point at x^* for any value of

ρ, and a local minimum at x^* when ρ is larger than some finite threshold $\bar{\rho}$. The parameter $\bar{\rho}$, like λ^* and $\hat{c}$, is generally unknown in advance, and must therefore be estimated during the course of the algorithm. A typical algorithm of this kind computes an approximate minimum of $\bar{\mathcal{L}}$ at each iterate by one or more inner iterations of the form

$$\min_{x \in \Re^n} \tilde{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \tilde{\mathcal{H}} p, \tag{6.3.13}$$

where

$$\tilde{g} \approx \nabla \bar{\mathcal{L}} \quad \text{and} \quad \tilde{\mathcal{H}} \approx \nabla^2 \bar{\mathcal{L}}$$

(see Chapter 2). A drawback of these methods is that the range of values of ρ for which the subproblems are well-conditioned may be very small (see, for example, Gill, Murray, and Wright [1981], Chapter 6). However, $\nabla^2 \bar{\mathcal{L}}(x^*, \lambda^*, \rho)$ and the Hessian of the Lagrangian $\nabla^2 \mathcal{L}(x^*, \lambda^*)$ (see (6.3.4)) have identical projections in $\mathcal{N}(\nabla \hat{c}(x^*))$. SQP methods compute the components of the search direction in $\mathcal{N}(\hat{c})$ and $\mathcal{N}(\hat{c})^{\perp}$ separately so that any ill-conditioning due to the penalty term is avoided. In an SQP method only the curvature in the null space of the active constraint normals is used to define the solution.

Recall that in nonlinear least squares, the Hessian matrix of the objective can be separated into the sum of two components involving different types of derivative information:

$$\nabla^2 \left(\frac{1}{2} f^{\mathrm{T}} f \right) = J^{\mathrm{T}} J + B,$$

where

$$B \equiv \sum_{i=1}^{m} \phi_i \nabla^2 \phi_i.$$

The corrected Gauss-Newton methods (Section 5.3) calculate a search direction that is separated into two orthogonal components when $0 < grade(J) < n$, and can be viewed as SQP methods. When $grade(J) = rank(J) < n$, the contributions of $J^{\mathrm{T}} J$ and of B (or of an approximation to B) are essentially decoupled because the contribution of $J^{\mathrm{T}} J$ in the projected Hessian is zero (see Section 5.3). No such separation is possible when $rank(J) = n$. In any case, $grade(J) < n$ may be selected based on the progress of the minimization as well as the singular values of J, so that partial separation of $J^{\mathrm{T}} J$ and B may occur between the extremes of Gauss-Newton ($grade(J) = rank(J)$), and a full Newton-type method ($grade(J) = 0$). The strategy of making a quasi-Newton approximation to B, which is then added to $J^{\mathrm{T}} J$ in a full Newton-type method, has not been successful outside a

neighborhood of the solution, unless it is combined with other techniques (see Section 5.4). *The approach taken here is to use a quasi-Newton approximation to the full Hessian, while separating out some of the contribution to the curvature due to $J^T J$ by including first-order information about the residuals as constraints.*

A final motivation is that the SQP framework allows considerable flexibility for variations in the type of nonlinear least-squares problem that is being solved. Information about the individual residuals, and about the interrelationships between residuals, can be used to construct the QP subproblems at any point in the domain. These aspects are discussed in detail in the next section.

6.4 Suitable QP Constraints

In this section we propose several types of constraints for QP subproblems and explain why and when they are appropriate for our application. In what follows, x refers to the current iterate, and x^* to a minimizer of the sum of squares.

6.4.1 Constraints Defined by Individual Residuals

6.4.1.1 Non-Ascent Constraints

When $|\phi_i(x)| \geq |\phi_i(x^*)|$, it would seem reasonable to try to achieve a decrease in the magnitude of ϕ_i at the next iterate. QP constraints consistent with this goal are of the form

$$\begin{aligned} b_i^L \leq \nabla\phi_i^T p \leq b_i^U \quad &\text{if } \phi_i < 0 \\ -b_i^L \leq \nabla\phi_i^T p \leq -b_i^U \quad &\text{if } \phi_i > 0, \end{aligned} \tag{6.4.1}$$

where $b_i^L \geq 0$ and $b_i^U \geq 0$. The theorem below characterizes search directions satisfying constraints of this type.

Theorem 6.4-1:

If $p \neq 0$ satisfies (6.4.1) for some nonzero residual ϕ_i, then either p is a direction of descent for $\phi_i(x)^2$, or p is orthogonal to $\nabla\phi_i$.

Proof:

The directional derivative of $\phi_i^2(x)$ along p is $2\phi_i(\nabla\phi_i^{\mathrm{T}}p)$. If $\nabla\phi_i^{\mathrm{T}}p \neq 0$, then the condition (6.4.1) requires $\nabla\phi_i^{\mathrm{T}}p$ to be opposite in sign from ϕ_i whenever ϕ_i is nonzero, so that $\phi_i(\nabla\phi_i^{\mathrm{T}}p)$ is negative. ∎

We call (6.4.1) a *non-ascent constraint* for ϕ_i.

A non-ascent constraint is equivalent to the following restriction on the directional derivative of ϕ_i^2 :

$$\begin{aligned} \phi_i b_i^L \leq \phi_i(\nabla\phi_i^{\mathrm{T}}p) \leq \quad \phi_i b_i^U \quad &\text{if } \phi_i < 0 \\ -\phi_i b_i^L \leq \phi_i(\nabla\phi_i^{\mathrm{T}}p) \leq -\phi_i b_i^U \quad &\text{if } \phi_i > 0, \end{aligned} \tag{6.4.2}$$

where $b_i^L \geq 0$ and $b_i^U \geq 0$. Treatment of zero-valued residuals is left undetermined in (6.4.2), because the constraint reduces to $0 \leq 0 \leq 0$ when $\phi_i = 0$. However, zero and near-zero residuals can be handled consistently in (6.4.1) by requiring the bounds b_i^L and b_i^U to approach zero as ϕ_i goes to zero. One possibility along these lines is to use equality constraints of the form

$$\nabla\phi_i^{\mathrm{T}}p = -\phi_i. \tag{6.4.3}$$

For small residuals, (6.4.3) is a sensible choice, because it defines a first-order step to a zero of ϕ_i. Moreover, (6.4.3) is precisely the type of QP constraint that would occur in an SQP method for a nonlinearly-constrained optimization problem if $\phi_i(x) = 0$ were among the constraints. Another reason for considering (6.4.3) is related to the structure of the nonlinear least-squares gradient as the sum of gradients of the individual residuals weighted by the residual values :

$$\nabla\left(\frac{1}{2}f^{\mathrm{T}}f\right) = J^{\mathrm{T}}f \equiv \bar{g} = \sum_{i=1}^{m}\phi_i\nabla\phi_i.$$

The contribution of relatively large residuals to the directional derivative

$$\bar{g}^{\mathrm{T}}p = \sum_{i=1}^{m}\phi_i\left(\nabla\phi_i^{\mathrm{T}}p\right) \tag{6.4.4}$$

should be large enough to force them to be decreased by the current step, but small enough to allow as many other residuals as possible to be decreased as well. The constraint (6.4.3) is reasonable for well-scaled problems, because the term in $\bar{g}^{\mathrm{T}}p$ corresponding to ϕ_i has the value $-\phi_i^2$.

There are a number of potential problems associated with having equality constraints

$$Ap = -c \tag{6.4.5}$$

in the QP subproblem. First, if the constraint gradients are linearly dependent and $c \neq 0$, there may be no feasible point. Such a situation could occur when $|\phi_i(x)| \geq |\phi_i(x^*)|$ for more than n residuals, and c is the vector of these residuals, with $A = \nabla c$. Moreover, inclusion of more than n constraints near a solution may be a poor strategy even if there is a feasible point, because superlinear convergence in an SQP method can be guaranteed only if the active constraint normals are linearly independent at a solution. The algorithms proposed in Section 6.5 include modification procedures for a given constraint set in order to ensure feasibility.

A second possible drawback in using equality constraints (6.4.5) in QP subproblems is that the method may yield poor search directions when the matrix A of constraint gradients is ill-conditioned. In order to compute a search direction from

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$$\text{subject to} \quad Ap = -c,$$

it is necessary to determine $\mathcal{R}(A^{\mathrm{T}})$, and $\mathcal{N}(A)$, which may be ill-defined when A is nearly rank-deficient (see Sections 4.2 and 6.2). Moreover, the resulting range-space component could be very large in magnitude, and involve significant computational error (see Chapter 3). A possible remedy is to use constraints of the form

$$\begin{aligned} 0 \;\; &\leq \nabla\phi_i^{\mathrm{T}} p \leq -\phi_i \qquad \text{if } \phi_i \leq 0 \\ -\phi_i \leq \nabla\phi_i^{\mathrm{T}} p &\leq \;\; 0 \qquad \text{if } \phi_i \geq 0 \end{aligned} \tag{6.4.6}$$

rather than (6.4.3). Like (6.4.3), (6.4.6) treats zero and near-zero residuals consistently. A constraint region bounded by constraints of type (6.4.6) always has $p = 0$ as a feasible point, although the origin may be the only feasible point when the constraint gradients are linearly dependent. The following theorem shows that with (6.4.6), there are always feasible directions that are of reasonable size whenever the feasible region is nontrivial, regardless of the condition number of the matrix of constraint gradients.

Theorem 6.4-2:

If $\bar{p} \neq 0$ satisfies

$$\min\{0, -c_i\} \leq a_i^{\mathrm{T}} p \leq \max\{0, -c_i\} \quad i = 1, 2, \ldots, k, \tag{6.4.7}$$

then $\gamma\bar{p}$ also satisfies (6.4.7) for all $\gamma \in [0, 1]$.

Proof:

A vector p is feasible in (6.4.7) if and only if the following two conditions hold :

$$|a_i^{\mathrm{T}} p| \leq |c_i|\,,$$

and either

$$a_i^{\mathrm{T}} p = 0 \quad \text{or} \quad sign(a_i^{\mathrm{T}} p) = -sign(c_i).$$

Hence $\gamma\bar{p}$ is feasible for all $\gamma \in [0, 1]$. ∎

A third disadvantage of equality constraints is that solutions to the QP subproblem may not be descent directions for $f^{\mathrm{T}} f$. If the constraints are of the form $(\nabla c)p = -c$, where c is a subvector of f, then the computation can proceed by using an alternative merit function

$$f^{\mathrm{T}} f + \rho \sum \phi_i^2, \tag{6.4.8}$$

where the sum is taken over any subset of nonzero residual components of c, since a feasible point p is a descent direction for (6.4.8) for some positive value of ρ. Other approaches that avoid this difficulty are used in the algorithms proposed in Section 6.5. The following theorem gives sufficient conditions for solutions to QP subproblems to be descent directions for $f^{\mathrm{T}} f$ when constraints are all of the form (6.4.7).

Theorem 6.4-3:

If there is a feasible point p satisfying $\bar{g}^{\mathrm{T}}p < 0$ and H is positive semi-definite, then the minimum p^* of the QP

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}Hp$$

$$\text{subject to} \quad \min\{0, -c_i\} \leq a_i^{\mathrm{T}}p \leq \max\{0, -c_i\}; \quad i = 1, 2, \ldots, k$$

is a direction of descent for $f^{\mathrm{T}}f$.

Proof:

The proof follows from Theorems (6.2-2) and (6.4-2). ∎

A drawback of (6.4.6) relative to (6.4.3) is that the presence of inequality constraints generally means that more than one iteration (and possibly many iterations) will be required to solve the resulting QP.

Another consideration in selecting bounds for (6.4.1), is that, for small residuals, imposing contraints of the form (6.4.3) or (6.4.6) may impede progress towards x^* when the distance from the current point to x^* is fairly large. When ϕ_i is small, such constraints force the SQP algorithm to follow the curve $\phi_i(x) = 0$ (which may be highly nonlinear) very closely. The effect is compounded when several small residuals are involved. Substantial gains might be made by temporarily permitting search directions to be directions of increase for small residuals. When only non-ascent constraints are allowed, the way to enable the search direction to be an initial direction of increase for small residuals is to omit constraints corresponding to that residual. Constraints that explicitly allow individual residual increases are discussed in the next subsection.

Before expanding the class of admissible constraints, we end this section with a discussion of a special class of non-ascent constraints called *orthogonality constraints*. These are equations of the form

$$\nabla\phi_i^{\mathrm{T}}p = 0, \tag{6.4.9}$$

requiring the search direction to be orthogonal to the gradient of the defining residual. Both (6.4.3) and (6.4.5) reduce to orthogonality constraints when $\phi_i = 0$. There are also several

motives for using this type of constraint for nonzero residuals. First, weaker conditions are required for descent with orthogonality constraints than with (6.4.3) or (6.4.6), as shown in the following theorem.

Theorem 6.4-4:

If $\bar{g}$ has a nonzero component that is orthogonal to A and H is positive semi-definite, then the minimum p^* of the QP

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$$\text{subject to} \quad Ap = 0$$

is a direction of descent for $f^{\mathrm{T}} f$.

Proof:

The set of feasible points in this case is the subspace $\mathcal{N}(A)$. If $\bar{p} \neq 0$ is the component of $\bar{g}$ onto $\mathcal{N}(A)$, it follows that $\bar{p}$ is a nonzero feasible point of the constraints. Moreover, since $\bar{p} \in \mathcal{N}(A)$ implies that $-\bar{p} \in \mathcal{N}(A)$, $-\bar{p}$ is also feasible. Hence there is a feasible descent direction (either $\bar{g}^{\mathrm{T}}\bar{p}$ or $\bar{g}^{\mathrm{T}}(-\bar{p})$ is negative), and the desired result follows from Theorem 6.2-2. ∎

The second motive for using orthogonality constraints is that there are always nonzero feasible points for

$$Ap = 0 \tag{6.4.10}$$

when rank of A is less than n, even when A has linearly dependent rows. On the other hand, $p = 0$ is the only feasible point for (6.4.10) when the rank is equal to n, whereas with (6.4.5) and (6.4.7) there could possibly be nonzero feasible points when $c \neq 0$. Third, imposition of an orthogonality constraint requires the corresponding residual to remain constant (to first order) during the iteration, enabling the search direction to favor reductions in other residuals. In the test problems used in this research (see the Appendix), most problems have residuals that are similar in magnitude at a minimum, regardless of whether or not they actually vanish there. When there is a wide variation in residual magnitudes, it may thus be advantageous to use orthogonality constraints for those that fall in the middle range. Finally, because orthogonality constraints are equations, only a single QP iteration is required to

resolve them. Orthogonality constraints have the same disadvantage for small residuals as their counterparts in (6.4.3) and (6.4.5) : the SQP method may force the iterates to stay close to a highly nonlinear surface at points away from the solution.

6.4.1.2 Ascent Bounds

In this section we consider constraints that allow the directional derivatives of the individual gradients to take on both positive and negative values. These constraints have the general form

$$\begin{aligned} -b_i^A \leq \nabla\phi_i^{\mathrm{T}} p \leq b_i^D \quad & \text{if } \phi_i < 0 \\ -b_i^D \leq \nabla\phi_i^{\mathrm{T}} p \leq b_i^A \quad & \text{if } \phi_i > 0, \end{aligned} \tag{6.4.11}$$

where $b_i^A \geq 0$ and $b_i^D \geq 0$. We shall refer to b_i^A and b_i^D as *ascent bounds* and *descent bounds*, respectively, for ϕ_i^2. Feasible vectors for a region of the form

$$-b^L \leq Ap \leq b^U,$$

with $b^L > 0$ and $b^U > 0$, need not be large in magnitude — regardless of whether or not constraint gradients are nearly linearly dependent — because the region includes a ball centered at the origin. Moreover, by Theorem 6.4-2, solutions to the QP

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to

$$-b^L \leq Ap \leq b^U,$$

with $b^L > 0$ and $b^U > 0$, are descent directions for $f^{\mathrm{T}} f$ when H is positive semi-definite. In (6.4.11), the inclusion of ascent bounds allows controlled increases for specific residuals.

We have already given reasons for choosing

$$b_i^D = |\phi_i| \tag{6.4.12}$$

in the preceding subsection. For small residuals, (6.4.12) restricts feasible points in (6.4.11) to be no larger than a first-order step to a zero of ϕ_i ; for large residuals, it prevents the search direction from disproportionately favoring descent for any particular residual at the expense of the rest. Definition (6.4.12) also makes it possible to extend the definition of descent

bounds to the case where $\phi_i = 0$. The choice of ascent bounds is not as straightforward. For consistent treatment of zero and near-zero residuals, it would seem that

$$b_i^A = |\phi_i| \tag{6.4.13}$$

is the proper choice, but there are some drawbacks. First, although we would normally want to favor decreasing the residuals that are the largest in magnitude, (6.4.13) means that there are large ascent bounds for large residuals. Second, reasons for introducing ascent bounds included the desire to allow increases in residual magnitude in two cases — when $|\phi_i(x)| < |\phi_i(x^*)|$, and for small residuals at points away from the solution in order to avoid following curved boundaries — both inconsistent with (6.4.13). These considerations motivate the use of different types of constraints for different residuals, as well as the addition of a mechanism to reject constraints after they are tried, in some of the algorithms proposed in Section 6.5.

6.4.2 Constraints Defined by the QR Factorization

It is also possible to formulate constraints for QP subproblems based on orthogonal factorizations of J. We shall limit our discussion to the QR factorization (see Sections 3.3.2 and 4.4.2), although it is equally possible to use the SVD (see Sections 3.3.3 and 4.4.1).

Recall that the QR factorization of J is given by

$$J = \begin{cases} Q(R \quad 0)P, & \text{if } m < n; \\ QRP, & \text{if } m = n; \\ Q\begin{pmatrix} R \\ 0 \end{pmatrix}P, & \text{if } m > n, \end{cases}$$

where R is upper triangular, Q and P are orthogonal, and P is a permutation of the columns of J. We assume that the diagonal elements of R are in non-increasing order of magnitude. Let

$$Q = (Q_1 \quad Q_2) \quad \text{and} \quad P = \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}$$

where Q_1 consists of the first $\min\{m,n\}$ columns of Q, and P_1 consists of the first $\min\{m,n\}$ rows of P. If

$$\bar{R} \equiv RP_1,$$

then

$$J = Q_1\bar{R},$$

so that

$$\bar{g}^{\mathrm{T}}p = f^{\mathrm{T}}Jp = f^{\mathrm{T}}Q_1\bar{R}p = \sum_{i=1}^{\min\{m,n\}} \bar{f}^{\mathrm{T}}(\bar{r}_i^{\mathrm{T}}p), \tag{6.4.14}$$

where

$$\bar{f} \equiv Q_1^{\mathrm{T}}f \equiv \begin{pmatrix} \bar{\phi}_1 \\ \vdots \\ \bar{\phi}_{\min\{m,n\}} \end{pmatrix},$$

and $\bar{r}_i^{\mathrm{T}}$ is the ith row of $\bar{R}$. The similarity between (6.4.14) and (6.4.4) suggests that it may be possible to substitute $\{\bar{\phi}_i\}$ for $\{\phi_i\}$, and $\{\bar{r}_i\}$ for $\{\nabla\phi_i\}$, in the constraints given in Section 6.4.1. An advantage is that no more than $\min\{m,n\}$ constraints need be considered. Moreover, the first $rank(J)$ components of $\bar{f}$ vanish at a minimum, because the first $rank(J)$ rows of Q form an orthonormal basis for $\mathcal{R}(J)$ (see Section 3.3.1). Hence there is no need to distinguish between zero and nonzero residuals in constraint-selection strategies except when J has nearly linearly dependent columns. A corrected Gauss-Newton method (see Section 5.3) can be obtained by taking

$$\bar{r}_i^{\mathrm{T}}p = -\bar{\phi}_i, \quad i = 1,2,\ldots,grade(J), \tag{6.4.15}$$

(analogous to (6.4.3)) as constraints, where $grade(J)$ is an integer approximating $rank(J)$. When $rank(J) = grade(J) = n$, the search direction is completely determined by the constraints in (6.1.1), and is a full-rank Gauss-Newton search direction. When $grade(J) < n$, part of the search direction depends on the objective in the QP subproblem. In our implementation, we use a QP to define $grade(J)$, rather than rely on the relative size of the diagonals, and the progress of the iteration, as is done in the corrected Gauss-Newton methods (see Section 6.5). With the following constraints,

$$\begin{aligned} 0 \;&\le \bar{r}_i^{\mathrm{T}}p \le -\bar{\phi}_i \qquad \text{if } \bar{\phi}_i \le 0 \\ -\bar{\phi}_i \;&\le \bar{r}_i^{\mathrm{T}}p \le \;\; 0 \qquad \text{if } \bar{\phi}_i \ge 0, \end{aligned} \tag{6.4.16}$$

which are analogous to (6.4.6), the SQP methods reduce to corrected Gauss-Newton methods near a solution because the first $rank(J)$ components of $\bar{f}$ vanish there. However, the asymptotic interpretation of (6.4.3) and (6.4.6) does not generally carry over to constraints based on $\bar{r}_i$ and $\bar{\phi}_i$, because $\bar{r}_i \neq \nabla\bar{\phi}_i$. The function $\bar{f}$ is not differentiable when J is column rank deficient, although otherwise it can be extended to a differentiable function (see Coleman and Sorensen [1984] and Goodman [1985]). Numerical tests show that the

corrected Gauss-Newton methods mentioned above are only linearly convergent when J is rank deficient.

6.5 SQP Algorithms

The motivation given so far allows considerable flexibility in formulating and developing sequential quadratic programming algorithms. In this section, we discuss some alternatives and examine their performance on a subset of the test problems. Two different approaches within the SQP framework are presented. Suppose that $\mathcal{T}$ is a tentative set of constraints (chosen from considerations detailed in the previous section), and that $QP^\star$ is the QP subproblem that ultimately determines the search direction. One approach uses a QP to select constraints in $\mathcal{T}$ in order to define $QP^\star$. In the other approach, the constraints in $QP^\star$ are defined by perturbing constraints in $\mathcal{T}$. The perturbations are either included as additional variables in $QP^\star$, or they are computed by solving an auxiliary QP. Although the two approaches are treated separately, they could be combined in future algorithms. A description of the numerical tests, including a complete listing of results, is given at the end of the section.

6.5.1 Algorithms that use a QP to Select Constraints

The algorithms treated in this section typically solve several related QP subproblems before deciding on a search direction. Algorithms of this type are characterized by the way in which they determine the next subproblem given the current QP, and also by the criterion for accepting or rejecting the solution to the most recent QP subproblem as a tentative search direction. The general form of an algorithm is given on the next page.

Although it might appear that solving many QP subproblems would result in an unjustifiably large amount of work per iteration, there are several reasons for considering such a strategy. First, starting the solution process for a QP with information about the solution of a related subproblem can often lead to significant savings in QP iterations (see, for example, Gill et al. [1985]). Second, when the cost of a function evaluation is much greater than the cost of a QP iteration, the effort involved in obtaining the search direction by solving more than one subproblem may be worthwhile if it results in a substantial reduction in the number of outer iterations.

Algorithm that uses a QP to select constraints

repeat

compute the solution p_{QN} to

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^\star \leftarrow p_{QN}$

select an initial set of constraints $\tilde{\mathcal{C}}$

loop

compute the solution $\tilde{p}$ to

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

decide whether to replace $p^\star$ by $\tilde{p}$

either add and/or delete constraints in $\tilde{\mathcal{C}}$ or else exit loop

forever

compute steplength α ; $x \leftarrow x + \alpha p^\star$

update f, $\bar{g}$, H

until termination criteria are satisfied

Aside from the choice of constraints and the priority scheme for including them in subproblems, an important feature of these algorithms is the mechanism for deciding whether or not to accept the current QP solution $\tilde{p}$ as a candidate for the search direction. Criteria for accepting $\tilde{p}$ must include a lower bound on $\|\tilde{p}\|_2$, to prevent search directions that are negligible in magnitude from being accepted :

$$\|\tilde{p}\|_2 > ptol(x) \geq 0, \tag{6.5.1}$$

and also an upper bound on $cos(\tilde{p}, \bar{g})$ to ensure that $\tilde{p}$ is bounded away from orthogonality to $\bar{g}$, and that it is a descent direction for the nonlinear least squares objective :

$$cos(\tilde{p}, \bar{g}) < -\theta_{\min} \leq 0. \tag{6.5.2}$$

In the numerical tests of Section 6.5.3, we have chosen the values

$$ptol(x) = \epsilon_M^{0.9}(1 + \|x\|_2)$$

and

$$\theta_{\min} = \epsilon_M^{2/3}.$$

For some of the tests, we also use the following criteria relative to the minimum, p_{QN}, of the QP objective function :

$$\|\tilde{p}\|_2 > \nu_p \|p_{QN}\|_2 , \tag{6.5.3}$$

$$cos(\tilde{p}, \bar{g}) < \nu_c cos(p_{QN}, \bar{g}), \tag{6.5.4}$$

and

$$\|\tilde{p}\|_2 \leq \max\{1 + \|x\|_2 , \|p_{QN}\|_2\}, \tag{6.5.5}$$

with

$$\nu_p = \nu_c = 0.01.$$

We have implemented some simple examples of these methods on a mixed set of test problems (see Section 6.5.3). Within any given iteration, a new QP subproblem differs from its predecessor by the addition of one new constraint, and possibly by the deletion of the constraint that was added to the previous QP. Numerical experiments were conducted to test various properties of the new methods (see Section 6.5.3).

One set of examples (Examples 2, 3, 6, 7 vs. Examples 4, 5, 8, 9) demonstrates the sensitivity of the methods to the order in which constraints are added to the QP subproblems, while another set (even numbered Examples 2–18 vs. odd numbered Examples 3–19) demonstrates sensitivity to the thresholds on $\|\tilde{p}\|_2$ and $cos(\tilde{p}, \bar{g})$ (conditions (6.5.1) – (6.5.5)). Examples 2–15 attempt to impose orthogonality constraints based on each individual residual (see (6.4.3) and (6.4.6)). There are instances in which these examples perform significantly better (in terms of function evaluations) than the BFGS method (Example 1), as well as some in which they are significantly worse. In particular, although the new methods perform well in general on zero-residual problems, they are not very good for problems with nonzero solutions, because they try to reduce residuals that may be at or below their minimum magnitude. Examples 16–19 attempt to impose nonascent constraints based on the QR factorization (see (6.4.15) and (6.4.16)). This results in an improvement, in some cases, over the constraints based on individual residuals in Examples 2–9, and a loss of efficiency in others. If the use of equality nonascent constraints ((6.4.3) and (6.4.15)) in Examples 2–5, 16, and 17 is compared the use of inequality nonascent constraints ((6.4.6) and (6.4.16)) in Examples 6–9, 18, and 19, we again find that neither approach is consistently better (or worse) than the other. In Examples 10–13, the choice of constraints is restricted to nonascent constraints corresponding only to relatively large residuals. As compared to Examples 2, 3, 6, and 7 which try nonascent constraints based on all of the residuals — starting with the largest — this modification results in significant improvement for one problem (23b.). Finally, Examples 14 and 15 try to impose orthogonality constraints corresponding to relatively small residuals in the subproblems. Although only a few test problems encounter residuals that are small enough be considered, these instances do show a signficant improvement over the BFGS method.

6.5.2 Algorithms that Obtain Constraint Bounds from a QP

The algorithms treated in this section modify individual QP constraints in a given set $\mathcal{T}$ in order to obtain a suitable subproblem, as opposed to selecting some subset of the constraints. If $\mathcal{T}$ is infeasible, or if the solution to

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$$\text{subject to} \quad p \in \mathcal{T}$$

is unacceptable as a search direction, then methods of this type relax some or all of the constraint bounds to form a new constraint set. The general form of an algorithm is given on the next page. We show how methods of this type are related to Gauss-Newton (Chapter 4) and Levenberg-Marquardt methods (Section 5.2), and present numerical results for some simple cases.

6.5.2.1 Minimal Constraint Bounds

In this subsection, we consider the problem of finding minimal perturbations to a given set of constraints $\mathcal{T}$, subject to the requirement that the resulting set of feasible points be nonempty. We shall limit ourselves to the case in which

$$\mathcal{T} = \{a_i^T p = -c_i \mid i = 1, 2, \ldots\} = \{0 \leq a_i^T p + c_i \leq 0 \mid i = 1, 2, \ldots\}$$

($\mathcal{T}$ is the set of linear equations $Ap = -c$), although it is straightforward to generalize from this example.

Minimal perturbations solve an optimization problem of the form :

$$\min_{b,p} \|b\| \tag{6.5.6}$$

subject to

$$-b^L \leq Ap + c \leq b^U$$
$$b = \begin{pmatrix} b^L \\ b^U \end{pmatrix} \geq 0$$
$$\begin{pmatrix} b \\ p \end{pmatrix} \in \mathcal{C},$$

where $\mathcal{C}$ represents additional constraints on b^L, b^U and p. For example, constraints in $\mathcal{C}$ may be simple bound constraints that restrict the components of b^L, b^U, and p to lie within fixed intervals. In all of the cases we consider, the constraints defining $\mathcal{C}$ are linear, so that if the objective is $\|b\|_2$, (6.5.7) is equivalent to a quadratic program. Alternatively, it would be possible to have $\|b\|_1$ or $\|b\|_\infty$ as the objective, so that (6.5.7) (with linear constraints in $\mathcal{C}$) is a linear program.

A simple example of (6.5.2) that involves only a single parameter β defines minimal bounds that are uniform over all of the constraints :

Algorithm that uses a QP to compute constraint bound

repeat

select an initial set of constraints C

use a QP to modify constraint bounds in C **and form** $\tilde{C}$

compute the solution p^* **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{C}$

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

$$\min_{\beta\,;p} \beta^{\mathrm{T}}\beta \tag{6.5.7}$$

subject to

$$\beta \geq 0$$

$$-\beta e \leq Ap + c \leq \beta e$$

$$e \equiv (1,\ldots,1)^{\mathrm{T}}.$$

The next theorem shows that (6.5.8) minimizes $\|Ap + c\|_\infty$.

Theorem 6.5-1 :

If $(\bar{\beta}\,;\bar{p})$ *solves* (6.5.8), *then* $\bar{p}$ *is an* l_∞ *solution to* $Ap \approx -c$.

Proof:

The $2m$ constraints of (6.5.8) are

$$-c \leq (e \quad A)\begin{pmatrix}\beta\\ p\end{pmatrix} \leq \infty \tag{C^L}$$

and

$$-\infty \leq (-e \quad A)\begin{pmatrix}\beta\\ p\end{pmatrix} \leq -c. \tag{C^U}$$

Suppose that $(\bar{\beta}\,;\bar{p})$ solves (6.5.8). Then at least one constraint must be active at $(\bar{\beta}\,;\bar{p})$. To see this, assume that all of the constraints are inactive. Define $\bar{r} = A\bar{p} + c$, and suppose $|\bar{r}_k| = \|\bar{r}\|_\infty$. Then $(|\bar{r}_k|\,;\bar{p})$ is a feasible point that reduces the objective. It follows that, at a solution, the objective has the value $\min_p \|Ap + c\|_\infty^2$. Hence if $(\bar{\beta}\,;\bar{p})$ solves (6.5.8), then $\bar{p}$ must be an l_∞ solution to $Ap \approx -c$. ∎

In another example, we define the bounds to be the smallest perturbation, in the l_2 norm, that allows all of the hyperplanes (6.5.6) to intersect. The bounds so defined solve the following QP :

$$\min_{b;p} b^{\mathrm{T}} b \tag{6.5.8}$$

subject to

$$b \equiv \begin{pmatrix} b^L \\ b^U \end{pmatrix} \geq 0$$

$$-b^L \leq Ap + c \leq b^U.$$

The next theorem shows that (6.5.9) minimizes $\|Ap + c\|_2$.

Theorem 6.5-2:

If $(\hat{b}^L ; \hat{b}^U ; \hat{p})$ *solves* (6.5.9), *then* $\hat{p}$ *is a least-squares solution to* $Ap \approx -c$.

Proof:

The $2m$ constraints of (6.5.9) are

$$-c \leq \begin{pmatrix} I & 0 & A \end{pmatrix} \begin{pmatrix} b^L \\ b^U \\ p \end{pmatrix} \leq \infty \tag{$\mathcal{C}^L$}$$

and

$$-\infty \leq \begin{pmatrix} 0 & -I & A \end{pmatrix} \begin{pmatrix} b^L \\ b^U \\ p \end{pmatrix} \leq -c. \tag{$\mathcal{C}^U$}$$

Suppose that $(\hat{b}^L ; \hat{b}^U ; \hat{p})$ solves (6.5.9). Let $\mathcal{C}_i^L$, $\mathcal{C}_i^U$ represent the ith constraint in $\mathcal{C}^L$, $\mathcal{C}^U$, respectively. For each i, at least one of $\mathcal{C}_i^L$ or $\mathcal{C}_i^U$ must be active at $(\hat{b}^L ; \hat{b}^U ; \hat{p})$. To see this, assume both are inactive for some value of i, and define $\hat{r} = A\hat{p} + c$. If $\hat{r}_i \geq 0$, then replacing $\hat{b}_i^U$ by $\hat{r}_i$ results in a feasible point that reduces the value of the objective. Similarly, if $\hat{r}_i \leq 0$, then $\hat{b}_i^L$ can be replaced by $-\hat{r}_i$. If both $\mathcal{C}_i^L$ and $\mathcal{C}_i^U$ are active, then $\hat{b}_i^L = \hat{b}_i^U = \hat{r}_i = 0$, while if only one is active, the other must vanish in order to minimize the objective. In view of these observations, the objective has the value $\sum_{i=1}^m \hat{r}_i^2 = \|A\hat{p} + c\|_2^2$ at a solution. Hence if $(\hat{b}^L ; \hat{b}^U ; \hat{p})$ solves (6.5.9), $\hat{p}$ must be a least-squares solution to $Ap \approx -c$. ∎

If $(b;\hat{p})$ solves (6.5.9), and $(\beta;p)$ solves (6.5.8), then $\|b\|_\infty \geq \beta$, because $(\|b\|_\infty;\hat{p})$ is feasible in (6.5.8).

When there is no reason to favor a perturbation in either direction, we might add the requirement that the upper and lower bounds be equal in magnitude, so that they solve the QP :

$$\min_{b;p} b^{\mathrm{T}}b \tag{6.5.9}$$

subject to

$$-b \leq Ap + c \leq b$$

$$b \geq 0.$$

The next theorem shows that, like (6.5.9), (6.5.10) also minimizes $\|Ap + c\|_2$.

Theorem 6.5-3 :

If $(\hat{b};\hat{p})$ solves (6.5.10), *then $\hat{p}$ is a least-squares solution to $Ap \approx -c$.*

Proof:

The $2m$ constraints of (6.5.10) are

$$-c \leq \begin{pmatrix} I & A \end{pmatrix} \begin{pmatrix} b \\ p \end{pmatrix} \leq \infty \tag{$\mathcal{C}^L$}$$

and

$$-\infty \leq \begin{pmatrix} -I & A \end{pmatrix} \begin{pmatrix} b \\ p \end{pmatrix} \leq -c. \tag{$\mathcal{C}^U$}$$

Suppose that $(\hat{b};\hat{p})$ solves (6.5.10). Let $\mathcal{C}_i^L$, $\mathcal{C}_i^U$ represent the ith constraint in $\mathcal{C}^L$, $\mathcal{C}^U$, respectively. For each i, at least one of $\mathcal{C}_i^L$ or $\mathcal{C}_i^U$ must be active at $(\hat{b};\hat{p})$, for if both are inactive for some value of i, then $\hat{b}_i$ can be replaced by $|\hat{r}_i|$, where $\hat{r} = A\hat{p} + c$, to reduce the objective while maintaining feasibility. Both $\mathcal{C}_i^L$ and $\mathcal{C}_i^U$ can be active only if $\hat{b}_i = \hat{r}_i = 0$.

Consequently, the objective is $\sum_{i=1}^{m} \hat{r}_i^2 = \|A\hat{p} + c\|_2^2$ at a minimum, so that $\hat{p}$ must be a least-squares solution to $Ap \approx -c$ if $(\hat{b}\,;\hat{p})$ solves (6.5.10). ∎

Solutions to (6.5.9) and (6.5.10) can easily be obtained from solutions to

$$\min_{p \in \Re^n} \|Ap + c\|_2^2, \tag{6.5.10}$$

or, equivalently,

$$\min_{b\,;p} b^{\mathrm{T}} b \tag{6.5.11}$$

$$\text{subject to} \quad Ap + c = b.$$

If $(\bar{b}\,;\bar{p})$ solves (6.5.12), then $(|\bar{b}|\,;\bar{p})$ solves (6.5.10), where $|\bar{b}|$ denotes the vector whose components are the absolute values of the components of $\bar{b}$. Moreover, if

$$\bar{b}_i^L = \begin{cases} |\bar{b}_i|, & \text{if } \bar{b}_i \le 0; \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\bar{b}_i^U = \begin{cases} \bar{b}_i, & \text{if } \bar{b}_i \ge 0; \\ 0, & \text{otherwise,} \end{cases}$$

then $(\bar{b}_i^L\,;\bar{b}_i^U\,;\bar{p})$ solves (6.5.9). Formulations (6.5.9) and (6.5.10) of (6.5.11) are important for our application because they seek explicit information about the feasible region of (6.1.1). The main result we shall use is the following corollary to Theorem 6.5-3.

Corollary 6.5-4:

If $(\hat{b}\,;\hat{p})$ *solves* (6.5.10), *then the region*

$$-\hat{b} \le Ap + c \le \hat{b} \tag{6.5.12}$$

contains only least-squares solutions to $Ap \approx -c$.

Proof:

Suppose $\tilde{p}$ satisfies (6.5.13). Then each element of $A\tilde{p} + c$ is restricted to be no larger in magnitude than the corresponding element of $\hat{b}$. Therefore

$$\|A\tilde{p} + c\|_2^2 \le \hat{b}^{\mathrm{T}}\hat{b} = \|A\hat{p} + c\|_2^2.$$

By Theorem (6.4-3), we cannot have $\|A\tilde{p}+c\|_2^2 < \|A\hat{p}+c\|_2^2$, since $\hat{p}$ minimizes $\|Ap+c\|_2$. Hence

$$\|A\tilde{p}+c\|_2^2 = \|A\hat{p}+c\|_2^2$$

must hold, so that $\tilde{p}$ also minimizes $\|Ap+c\|_2$. ∎

Corollary (6.5-4) implies that the bounds in (6.1.1) may need to be large in magnitude if the rows of A are linearly dependent. Another implication is that if the columns of A are linearly independent, then the feasible region defined by (6.5.13) contains only one vector, which may have large components if A is ill-conditioned (see Chapter 3). In the next section we show how to modify QP techniques for finding minimal bounds in order to take into account considerations beyond feasibily that are important in formulating (6.1.1).

6.5.2.2 Generalized Levenberg-Marquardt Methods

One scheme for obtaining suitable bounds is to compute them from a QP similar to (6.5.10), but with a penalty term $\omega p^{\mathrm{T}}p$, $\omega > 0$, added to the objective :

$$\min_{b\,;p} b^{\mathrm{T}}b + \omega p^{\mathrm{T}}p \tag{6.5.13}$$

subject to

$$-b \leq Ap + c \leq b.$$

$$b \geq 0$$

This technique forces the bounds to increase in magnitude when p would otherwise be large. By arguments very similar to those in the proof of Theorem 6.5-3, it can be shown that the direction $\tilde{p}$ that solves the augmented version (6.5.13) is a Levenberg-Marquardt search direction (see Section 5.2), that is

$$\tilde{p} = \mathrm{argmin}_{p\in\Re^n} c^{\mathrm{T}}Ap + \frac{1}{2}p^{\mathrm{T}}(A^{\mathrm{T}}A + \omega I)p.$$

The following numerical experiments were conducted on a mixed set of test problems (see Section 6.5.3) :

(i) Generalized Levenberg-Marquardt algorithm that computes bounds $\tilde{b}$ from (6.5.13) with $A = J$ and $c = f$, followed by computation of the search direction from

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p \qquad (6.5.14)$$
$$\text{subject to} \quad -\tilde{b} \leq Jp + f \leq \tilde{b}$$

(Example 22)

(ii) Same as (i) but with box constraints in (6.5.13)

(Example 24)

(iii) Generalized Levenberg-Marquardt algorithm that computes bounds $\tilde{b}$ from (6.5.13) with $A = J$ and $c = f$, followed by computation of the search direction from

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p \qquad (6.5.15)$$
$$\text{subject to} \quad -\tilde{b} - \sqrt{\omega} \leq Jp + f \leq \tilde{b} + \sqrt{\omega},$$

(Example 25)

(iv) Same as (i) and (iii) with the QR factorization

(Examples 21, 23, 26)

The method in (i) is rather efficient (in terms of function evaluations) relative to the BFGS method (Example 1) on most problems. However, this appears to be an attribute of the underlying Levenberg-Marquardt method (Example 20) rather than of (6.5.14). Only problems **35b.** and **40g.**, which are solved much more efficiently by BFGS than by Levenberg-Marquardt, benefit from the use of the second subproblem (6.5.14). By contrast, the method in (i) is considerably less efficient than Levenberg-Marquardt for problem **45d.** (on which the BFGS method fails).

From these observations, it might seem that (i) is a hybrid of the BFGS method and the Levenberg-Marquardt method. However, once $p = 0$ becomes feasible in (6.5.13), bounds can no longer be expanded by increasing ω. The method in (iii) is such a hybrid, because bounds in (6.5.15) go to infinity with ω. The use of this modification results in a gain in efficiency for those problems favored by the BFGS method (**35a.** and **40g.**), and a significant loss in efficiency for many of the other problems. Methods based on the QR factorization (iv) also produce mixed results, with better performance on some problems relative to the Levenberg- Marquardt method, and worse performance on others.

The algorithms discussed so far in this section allow arbitrary perturbations of the constraints, so that the computation of ω in the objective of (6.5.13) from a bound on the l_2 norm of p is relatively straightforward (a subroutine fron MINPACK was used). When there are restrictions on the perturbations (for example, we may be allowed to relax only an upper or a lower bound), it may not be possible to obtain ω directly. Such problems can be treated by using box constraints (bounds on $\|p\|_\infty$) rather than ellipsoidal contraints (bounds on $\|p\|_2$). Care must be taken to ensure that the bound on $\|p\|_\infty$ is no smaller than the minimum value that admits a feasible point. In (ii), we tested a method that uses box constraints and found performance to be similar to that observed for ellipsoidal constraints, but somewhat less efficient overall.

6.5.3 Details of the Numerical Tests

6.5.3.1 Software

The software package LSSOL [Gill et al. (1986a)] is used to solve the QP subproblem (6.1.1). This is combined with a linesearch procedure taken from the nonlinear programming code NPSOL [Gill et al. (1979); (1986b)] that requires both function values and first derivatives. The approximate Hessian H is set to I intially, and subsequently modified using the BFGS update. The update is omitted if $y_k^T s_k < -0.1\ \bar{g}_k^T p_k$, since otherwise the Hessian matrix might become singular or indefinite during the course of the algorithm (see Section 2.5). The update was never skipped in any of our tests, so that it could be used without modification to take into account the effect of constraints. For comparison, we have included results for the BFGS method for unconstrained optimization again using LSSOL to solve the QP subproblem, with the same update and linesearch, applied to each problem that was chosen to test the new methods. In making comparisons, the overall efficiency of SQP methods depends not only on the number of function evaluations, but also on number of QP iterations required to obtain a solution. When (6.1.1) includes inequality constraints, a number of iterations may be required in order to obtain a solution (see the references cited in Section 6.3). Moreover, our new methods typically solve more than one QP in an iteration, so that it is important to demonstrate that the subproblems can be solved efficiently.

The software package written by Wright and Glassman [1978] was used to compute the QR factorization of the Jacobian when required to define constraints. The parameters were chosen so that the estimated rank of the triangular factor R was maximal.

6.5.3.2 Parameters

With the exception of `Infinite Bound Size`, which was set at 10^{20}, parameters in LSSOL were kept at their default values.
The program was modified to accept a different feasibility tolerance for each constraint.
See Gill et al. [1986a] for details concerning the parameters in LSSOL.

In addition, the values $\eta = 0.5$ and $\alpha_{\max} = \min\{(2(1+\|x\|_2)+1)/\|p\|_2, 10^{20}\}$ are chosen for the linesearch.

6.5.3.3 Convergence Criteria

The convergence criteria are the same as those given in Section 4.7 for the Gauss-Newton methods.

In the tables, the following notation is used to describe conditions under which the algorithm terminates :

ABS. F	-	(4.7.1)
G	-	(4.7.2)
X	-	(4.7.3)
F LIM.	-	function evaluation limit reached
UNB. QP	-	unbounded QP subproblem
QP LIM.	-	iteration limit reached in QP subproblem

6.5.3.4 Notation

The following notation refers to the QR factorization of J (see Section 6.4.2) :

$$J = QRP$$

$\bar{R}$	:	the first $\min\{m,n\}$ rows of RP
$\bar{r}_i$	:	the ith row of $\bar{R}$
$\bar{f}$	:	the first $\min\{m,n\}$ rows of $Q^{\mathrm{T}}f$
$\bar{\phi}_i$	:	the ith component of $\bar{f}$

6.5.3.5 Test Problems

For testing the new methods, seven zero residual problems and seven problems with non-zero solutions, have been selected. They are as follows (where the comments refer to the test problems and software of Chapters 2, 4, and 5) :

Zero-Residual Problems

14. Wood $n = 4$; $m = 6$

This is an overdetermined set of linear equations. Most of the methods require a rather large number of function evaluations to solve this problem relative to its size.

21b. Extended Rosenbrock $n = m = 20$

This problem can be solved more efficiently, in terms of function evaluations, by specialized nonlinear least-squares methods than by unconstrained methods that use only first derivatives.

22b. Extended Powell Singular $n = m = 20$

This is a zero-residual problem in which the Jacobian is singular at the solution, so that none of the methods tested converges at a superlinear rate on this example.

29b. Discrete Integral $n = m = 20$

This problem is efficiently solved by all of the methods.

35b. Chebyquad $n = m = 9$

This is a zero-residual problem that is difficult for Gauss-Newton methods because of ill-conditioning in the Jacobian, but fairly efficiently solved by other methods.

36a. Matrix Square Root 1 $n = m = 4$

All of the algorithms tested in previous chapters failed on this problem except full-rank Gauss-Newton methods (see Section 4.5.2) and corrected Gauss-Newton methods (see Sections 5.3 and 5.6.2).

45d. Dennis, Gay, and Vu $n = m = 8$

This problem can be solved efficiently by Gauss-Newton methods and the corrected Gauss-Newton methods, although it is very difficult for unconstrained optimization methods, and moderately difficult for the other nonlinear least-squares methods.

Problems with Nonzero Solutions

9. Gauss $n = 3$; $m = 15$

This problem is efficiently solved by all of the methods.

19. Osborne 2 $n = 11$; $m = 65$

This problem is solved far more efficiently by the specialized nonlinear least-squares methods than by unconstrained optimization methods that use only first-derivative information.

20d. Watson $n = 20$; $m = 31$

This problem has several local minima where there are small but nonzero residuals, and is difficult for the unconstrained optimization methods. The problem is also characterized by a very ill-conditioned Jacobian (see Section 4.5.3), but is nevertheless easily solved by Gauss-Newton methods.

23b. Penalty I $n = 10$; $m = 11$

This problem is very difficult for the BFGS method with linesearch (NPSOL), but only moderately difficult for the other first-derivative methods.

24a. Penalty II $n = 4$; $m = 8$

This is a small problem that is very difficult for Gauss-Newton methods, the quasi-Newton version of the corrected Gauss-Newton method, and the first-derivative methods for unconstrained optimization, and moderately difficult for all other methods tested.

35a. Chebyquad $n = m = 8$

This is a problem with nonzero solution that is difficult for Gauss-Newton methods because of ill-conditioning in the Jacobian, but fairly easily solved by other methods. It has the unusual property that some of the residuals vanish at a minimum, while others are much larger in magnitude.

40g. McKeown 2 $n = 3$; $m = 4$

This is a small problem that is easily solved by unconstrained optimization methods, but is difficult for most nonlinear least-squares methods. In particular, although the Jacobian is well-conditioned, Gauss-Newton methods fail. This test problem was constructed so that the unit-step Gauss-Newton method is locally divergent (see Section 4.3).

Examples of the SQP Algorithms

Example 1

(BFGS Method)

repeat

compute solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

compute steplength α ; $x \leftarrow x + \alpha p_{QN}$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : An implementation of the BFGS method that uses the same software for the QI solver, quasi-Newton update, and linesearch as the SQP methods that follow.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	45	29	1	2.00	10^{-12}	10^{-11}	10^{-25}	G
21b.^{0}	20	20	209	104	1	4.47	10^{-13}	10^{-12}	10^{-27}	G
22b.^{0}	20	20	210	125	1	10^{-4}	10^{-9}	10^{-11}	10^{-17}	G
29b.^{0}	20	20	9	7	1	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	38	22	1	1.73	10^{-12}	10^{-12}	10^{-24}	G
36a.^{0}	4	4	(4000)	(2418)	1	16.8	10^{-6}	10^{-6}	10^{-11}	F LIM.
45d.^{0}	8	8	(1385)	(873)	1	52.0	10^{-1}	10^{1}	10^{-2}	UNB. QP

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	86	52	1	9.38	10^{-1}	10^{-11}	10^{-5}	G
20d.	20	31	99	47	1	1.06	10^{-7}	10^{-11}	10^{-13}	G
23b.	10	11	193	97	1	0.500	10^{-2}	10^{-12}	10^{-11}	G
24a.	4	8	670	392	1	0.759	10^{-3}	10^{-12}	10^{-11}	G
35a.	8	8	39	24	1	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	41	18	1	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 2

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \leq tol$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^T p = -\phi_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose an equality nonascent constraint for eacl

The constraints are tried in order of decreasing residual magnitude.

Example 2

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	388	166	11.6	2.00	10^{-12}	10^{-11}	10^{-25}	G
21b.^{0}	20	20	34	13	39.3	4.47	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
22b.^{0}	20	20	16	15	27.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	16.6	0.571	10^{-11}	10^{-11}	10^{-23}	G
35b.^{0}	9	9	31	15	6.40	1.73	10^{-11}	10^{-11}	10^{-21}	G
36a.^{0}	4	4	(4004)	(630)	7.97	12.3	10^{-5}	10^{-5}	10^{-10}	F LIM.
45d.^{0}	8	8	80	35	17.3	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	447	224	8.92	1.08	10^{-4}	10^{-11}	10^{-14}	G
19.†	11	65	(1144)	(237)	43.0	9.38	10^{-1}	10^{-3}	10^{-6}	TIME
20d.	20	31	116	64	20.3	1.30	10^{-8}	10^{-12}	10^{-17}	G
23b.	10	11	(1668)	(378)	22.8	0.524	10^{-2}	10^{-2}	10^{-3}	TIME
24a.	4	8	(4000)	(825)	8.55	0.841	10^{-1}	10^{0}	10^{-1}	F LIM.
35a.	8	8	(1342)	(651)	13.9	1.65	10^{-1}	10^{-4}	10^{-7}	TIME
40g.	3	4	(3003)	(529)	6.51	10^{-1}	10^{0}	10^{2}	10^{1}	F LIM.

† $\alpha_{\max} = \min\{(0.5(1 + \|x\|_2) + 1) / \|p\|_2, 10^{20}\}$

Example 3

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \leq tol$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^T p = -\phi_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.2)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP s as potential search directions.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	260	89	7.00	2.00	10^{-12}	10^{-11}	10^{-24}	G
21b.^{0}	20	20	34	13	39.3	4.47	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
22b.^{0}	20	20	16	15	27.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	16.6	0.571	10^{-11}	10^{-11}	10^{-23}	G
35b.^{0}	9	9	156	37	11.9	1.73	10^{-12}	10^{-12}	10^{-24}	G
36a.^{0}	4	4	102	42	6.86	45.8	10^{-8}	10^{-12}	10^{-16}	G
45d.^{0}	8	8	38	15	17.2	15.3	10^{-15}	10^{-13}	10^{-30}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	287	144	8.22	1.08	10^{-4}	10^{-8}	10^{-14}	X
19.†	11	65	(854)	(334)	30.2	11.4	10^{0}	10^{0}	10^{0}	TIME
20d.	20	31	96	54	22.0	1.91	10^{-8}	10^{-11}	10^{-16}	G
23b.	10	11	(1908)	(408)	19.0	0.703	10^{-1}	10^{-1}	10^{-1}	TIME
24a.	4	8	(4001)	(800)	7.01	12.3	10^{0}	10^{1}	10^{0}	F LIM.
35a.	8	8	126	19	14.0	1.63	10^{-1}	10^{-1}	10^{-2}	$p = 0$
40g.	3	4	(3001)	(506)	6.51	0.112	10^{1}	10^{3}	10^{1}	F LIM.

† $\alpha_{\max} = \min\{(0.5(1 + \|x\|_2) + 1) / \|p\|_2, 10^{20}\}$

Example 4

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \leq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $i \leq m$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^T p = -\phi_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose an equality nonascent constraint for each
The constraints are tried in order of increasing residual magnitude (the opposite or
Example 2).

Example 4

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	25	15	4.73	2.00	10^{-12}	10^{-11}	10^{-24}	G
21b.^{0}	20	20	34	13	38.5	4.47	0.00	0.00	0.00	ABS. F, G
22b.^{0}	20	20	16	15	26.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	16.6	0.571	10^{-11}	10^{-11}	10^{-23}	G
35b.^{0}	9	9	29	14	8.64	1.73	10^{-11}	10^{-11}	10^{-23}	G
36a.^{0}	4	4	(4004)	(601)	8.01	8.55	10^{-5}	10^{-4}	10^{-9}	F LIM.
45d.^{0}	8	8	179	54	17.3	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	314	160	8.92	1.08	10^{-4}	10^{-11}	10^{-14}	G
19.†	11	65	(824)	(201)	27.0	9.27	10^{-1}	10^{0}	10^{-2}	TIME
20d.	20	31	98	58	11.2	24.4	10^{-8}	10^{-12}	10^{-16}	G
23b.	10	11	(1808)	(422)	22.9	0.512	10^{-2}	10^{-2}	10^{-4}	TIME
24a.	4	8	2034	882	15.7	0.759	10^{-3}	10^{-14}	10^{-11}	G
35a.	8	8	(621)	(294)	12.3	1.65	10^{-1}	10^{-4}	10^{-7}	TIME
40g.	3	4	(3001)	(518)	6.87	0.114	10^{1}	10^{2}	10^{1}	F LIM.

† $\alpha_{\max} = \min\{(0.5(1+\|x\|_2)+1)/\|p\|_2, 10^{20}\}$

Example 5

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \leq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $i \leq m$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^T p = -\phi_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.2)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP as potential search directions.

Example 5

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	25	15	4.73	2.00	10^{-12}	10^{-11}	10^{-24}	G
21b.^{0}	20	20	34	13	39.3	4.47	0.00	0.00	0.00	ABS. F, G
22b.^{0}	20	20	16	15	26.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	16.6	0.571	10^{-11}	10^{-11}	10^{-23}	G
35b.^{0}	9	9	166	38	12.8	1.73	10^{-16}	10^{-15}	10^{-31}	ABS. F, G
36a.^{0}	4	4	3233	664	7.98	50.0	10^{-13}	10^{-11}	10^{-25}	G
45d.^{0}	8	8	161	44	13.7	15.3	10^{-13}	10^{-12}	10^{-26}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	(3000)	(1501)	5.80	1.07	10^{-3}	10^{-4}	10^{-2}	F LIM.
19.†	11	65	(1244)	(441)	23.2	9.50	10^{-1}	10^{-1}	10^{-2}	TIME
20d.	20	31	(199)	(101)	28.0	1.07	10^{-6}	10^{-5}	10^{-12}	TIME
23b.	10	11	(544)	(275)	20.9	2.92	10^{1}	10^{1}	10^{2}	TIME
24a.	4	8	(4002)	(793)	7.15	0.687	10^{-1}	10^{-1}	10^{-2}	F LIM.
35a.	8	8	125	21	12.4	1.63	10^{-1}	10^{-1}	10^{-3}	$p = 0$
40g.	3	4	(3001)	(522)	6.71	0.120	10^{1}	10^{3}	10^{1}	F LIM.

† $\alpha_{\max} = \min \{(0.2(1 + \|x\|_2) + 1) / \|p\|_2, 10^{20}\}$

Example 6

(Algorithm with Inequality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \leq tol$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min\{-\phi_i, 0\} \leq \nabla \phi_i^{\mathrm{T}} p \leq \max\{-\phi_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose an inequality nonascent constraint for eac
ual. The constraints are tried in order of decreasing residual magnitude.

Example 6

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	206	164	5.80	2.00	10^{-13}	10^{-12}	10^{-26}	G
21b.^{0}	20	20	76	50	22.7	4.47	10^{-11}	10^{-11}	10^{-22}	ABS. F, G
22b.^{0}	20	20	73	44	18.8	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	8	7	16.1	0.571	10^{-12}	10^{-12}	10^{-24}	G
35b.^{0}	9	9	25	17	6.88	1.73	10^{-11}	10^{-11}	10^{-21}	G
36a.^{0}	4	4	(4001)	(2662)	4.70	20.2	10^{-6}	10^{-6}	10^{-12}	F LIM.
45d.^{0}	8	8	299	205	24.2	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	13	10	2.30	1.08	10^{-4}	10^{-15}	10^{-14}	G
19.	11	65	(439)	(419)	24.9	9.38	10^{-1}	10^{-3}	10^{-8}	TIME
20d.	20	31	(143)	(121)	40.9	1.06	10^{-5}	10^{-6}	10^{-10}	TIME
23b.	10	11	191	114	10.3	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	143	98	3.13	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	46	32	11.2	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	57	41	2.90	10^{-9}	10^{0}	10^{-12}	10^{-7}	G

Example 8

(Algorithm with Inequality Nonascent Constraints)

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

assume that $|\phi_i(x)| \leq |\phi_j(x)|$ **if** $i \leq j$

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $i \leq m$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min\{-\phi_i, 0\} \leq \nabla \phi_i^T p \leq \max\{-\phi_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies** (6.5.1) - (6.5.5)

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose an inequality nonascent constraint for eac
ual. The constraints are tried in order of increasing residual magnitude.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	24	16	4.38	2.00	10^{-13}	10^{-12}	10^{-26}	G
21b.^{0}	20	20	65	43	23.1	4.47	10^{-12}	10^{-11}	10^{-24}	G
22b.^{0}	20	20	86	49	15.8	10^{-5}	10^{-10}	10^{-12}	10^{-19}	G
29b.^{0}	20	20	8	7	15.1	0.571	10^{-12}	10^{-12}	10^{-24}	G
35b.^{0}	9	9	21	14	9.43	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	(4000)	(2533)	4.96	20.2	10^{-6}	10^{-6}	10^{-12}	F LIM.
45d.^{0}	8	8	254	155	22.4	15.3	10^{-15}	10^{-11}	10^{-29}	ABS. F, G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	70	67	5.49	1.08	10^{-4}	10^{-14}	10^{-14}	F LIM.
19.	11	65	(400)	(367)	38.3	9.38	10^{-1}	10^{-2}	10^{-5}	TIME
20d.	20	31	(105)	(75)	76.9	1.05	10^{-4}	10^{-3}	10^{-8}	TIME
23b.	10	11	(523)	(415)	19.8	0.500	10^{-2}	10^{-5}	10^{-6}	TIME
24a.	4	8	455	283	5.70	0.759	10^{-3}	10^{-12}	10^{-11}	G
35a.	8	8	45	31	10.5	1.65	10^{-1}	10^{-12}	10^{-9}	G
40g.	3	4	(3001)	(2980)	5.97	10^{-7}	10^{0}	10^{-4}	10^{-7}	F LIM.

Example 3

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \leq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $i \leq m$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min\{-\phi_i, 0\} \leq \nabla \phi_i^T p \leq \max\{-\phi_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.2)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP as potential search directions.

Example 9

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	24	16	4.38	2.00	10^{-13}	10^{-12}	10^{-26}	G
21b.^{0}	20	20	29	21	22.6	4.47	0.00	0.00	0.00	ABS. F, G
22b.^{0}	20	20	87	52	12.4	10^{-5}	10^{-10}	10^{-11}	10^{-19}	G
29b.^{0}	20	20	8	7	15.1	0.571	10^{-12}	10^{-12}	10^{-24}	G
35b.^{0}	9	9	21	14	9.43	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	656	401	4.67	50.0	10^{-10}	10^{-12}	10^{-21}	G
45d.^{0}	8	8	164	106	23.5	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	79	55	5.80	1.08	10^{-4}	10^{-6}	10^{-10}	X
19.	11	65	(670)	(416)	36.4	9.55	10^{-1}	10^{0}	10^{-1}	TIME
20d.	20	31	117	62	42.0	1.06	10^{-8}	10^{-11}	10^{-15}	G
23b.	10	11	(873)	(288)	18.9	2.45	10^{1}	10^{1}	10^{1}	TIME
24a.	4	8	(4000)	(1709)	6.82	0.767	10^{-3}	10^{-6}	10^{-8}	F LIM.
35a.	8	8	(518)	(301)	17.3	1.65	10^{-1}	10^{-2}	10^{-5}	TIME
40g.	3	4	225	199	4.45	10^{-4}	10^{0}	10^{-1}	10^{-5}	X

Example 10

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

define $f_{mean} = \sqrt{\max_i\{|\phi_i(x)|\} * (\min_i\{|\phi_i(x)|\} + \epsilon_M)}$

compute the solution p_{QN} **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \geq 10^2 * f_{mean}$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^T p = -\phi_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose an equality nonascent constraint correspo

each residual whose magnitude is larger than a certain threshold value in the QP subp

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	108	71	1.17	2.00	10^{-13}	10^{-12}	10^{-26}	G
21b.^{0}	20	20	209	104	1.00	4.47	10^{-13}	10^{-12}	10^{-27}	G
22b.^{0}	20	20	(604)	(299)	9.21	10^{-4}	10^{-7}	10^{-9}	10^{-14}	TIME
29b.^{0}	20	20	9	7	1.00	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	33	16	4.50	1.73	10^{-11}	10^{-11}	10^{-21}	G
36a.^{0}	4	4	(4000)	(2321)	1.03	16.6	10^{-6}	10^{-6}	10^{-11}	F LIM.
45d.^{0}	8	8	3741	2142	1.46	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	88	52	1.00	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	80	44	2.52	1.06	10^{-6}	10^{-11}	10^{-12}	G
23b.	10	11	147	83	1.24	0.500	10^{-3}	10^{-11}	10^{-11}	G
24a.	4	8	670	392	1.00	0.759	10^{-3}	10^{-12}	10^{-11}	G
35a.	8	8	118	45	6.91	1.65	10^{-1}	10^{-12}	10^{-9}	G
40g.	3	4	41	18	1.00	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 11

(Algorithm with Equality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

define $f_{mean} = \sqrt{\max_i\{|\phi_i(x)|\} * (\min_i\{|\phi_i(x)|\} + \epsilon_M)}$

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $C^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \geq 10^2 * f_{mean}$ **and** $ncon \leq maxcon$ **do**

$\tilde{C} \leftarrow C^* \cup \{\nabla \phi_i^T p = -\phi_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{C}$

if $\tilde{p}$ **satisfies** (6.5.1) - (6.5.2)

then $p^* \leftarrow \tilde{p}$; $C^* \leftarrow \tilde{C}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP as potential search directions.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	108	71	1.17	2.00	10^{-13}	10^{-12}	10^{-26}	G
21b.^{0}	20	20	209	104	1.00	4.47	10^{-13}	10^{-12}	10^{-27}	G
22b.^{0}	20	20	(1720)	(600)	4.62	10^{-3}	10^{-5}	10^{-7}	10^{-11}	TIME
29b.^{0}	20	20	9	7	1.00	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	114	41	3.29	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	(4004)	(2300)	1.03	16.5	10^{-6}	10^{-6}	10^{-11}	F LIM.
45d.^{0}	8	8	(7935)	(4743)	1.10	16.5	10^{-2}	10^{0}	10^{-3}	TIME

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	88	52	1.00	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	84	46	2.63	1.06	10^{-6}	10^{-11}	10^{-12}	G
23b.	10	11	147	83	1.24	0.500	10^{-3}	10^{-11}	10^{-11}	G
24a.	4	8	1329	705	1.18	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	2114	1038	8.91	1.65	10^{-1}	10^{-6}	10^{-9}	X
40g.	3	4	41	18	1.00	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 12

(Algorithm with Inequality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

define $f_{mean} = \sqrt{\max_i\{|\phi_i(x)|\} * (\min_i\{|\phi_i(x)|\} + \epsilon_M)}$

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

compute the solution p_{QN} **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \geq 10^2 * f_{mean}$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min\{-\phi_i, 0\} \leq \nabla\phi_i^T p \leq \max\{-\phi_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose an inequality nonascent constraint corres
to each residual whose magnitude is larger than a certain threshold value in the QP subp

Example 12

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	111	68	1.18	2.00	10^{-13}	10^{-12}	10^{-25}	G
21b.^{0}	20	20	209	104	1.00	4.47	10^{-13}	10^{-12}	10^{-27}	G
22b.^{0}	20	20	182	109	1.13	10^{-4}	10^{-7}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	9	7	1.00	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	34	18	4.28	1.73	10^{-16}	10^{-16}	10^{-32}	ABS. F, G
36a.^{0}	4	4	(4003)	(2359)	1.00	16.8	10^{-6}	10^{-6}	10^{-11}	F LIM.
45d.^{0}	8	8	4296	2728	1.06	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	86	52	1.00	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	88	43	5.12	1.06	10^{-7}	10^{-12}	10^{-13}	G
23b.	10	11	63	41	1.05	0.500	10^{-3}	10^{-11}	10^{-11}	G
24a.	4	8	661	379	1.03	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	37	23	6.17	1.65	10^{-1}	10^{-12}	10^{-9}	G
40g.	3	4	41	18	1.00	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 13

(Algorithm with Inequality Nonascent Constraints)

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

define $f_{mean} = \sqrt{\max_i\{|\phi_i(x)|\} * (\min_i\{|\phi_i(x)|\} + \epsilon_M)}$

assume that $|\phi_i(x)| \geq |\phi_j(x)|$ **if** $i \leq j$

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \geq 10^2 * f_{mean}$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min\{-\phi_i, 0\} \leq \nabla\phi_i^T p \leq \max\{-\phi_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.2)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP s
as potential search directions.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	111	68	1.18	2.00	10^{-13}	10^{-12}	10^{-25}	G
21b.^{0}	20	20	209	104	1.00	4.47	10^{-13}	10^{-12}	10^{-27}	G
22b.^{0}	20	20	182	109	1.13	10^{-4}	10^{-7}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	9	7	1.00	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	34	18	4.28	1.73	10^{-16}	10^{-16}	10^{-32}	ABS. F, G
36a.^{0}	4	4	(4003)	(2359)	1.00	16.8	10^{-6}	10^{-6}	10^{-11}	F LIM.
45d.^{0}	8	8	(7863)	(4831)	1.11	13.6	10^{-2}	10^{0}	10^{-3}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	86	52	1.00	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	88	43	5.12	1.06	10^{-7}	10^{-12}	10^{-13}	G
23b.	10	11	63	41	1.05	0.500	10^{-3}	10^{-11}	10^{-11}	G
24a.	4	8	664	383	1.04	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	37	13	11.4	1.64	10^{-1}	10^{-2}	10^{-3}	X
40g.	3	4	41	18	1.00	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 14

(Algorithm with Orthogonality Constraints)

assume that $|\phi_i(x)| \leq |\phi_j(x)|$ **if** $i \leq j$

$tol \leftarrow \epsilon_M^{0.9}$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \leq tol$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^T p = 0\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5)**

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose orthogonality constraints in the QP subp
corresponding to residuals that are smaller in magnitude than a certain threshold *tol*.

Example 14

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	27	20	2.40	2.00	10^{-14}	10^{-13}	10^{-27}	G
21b.^{0}	20	20	209	104	1.00	4.47	10^{-13}	10^{-12}	10^{-27}	G
22b.^{0}	20	20	210	125	1.00	10^{-4}	10^{-9}	10^{-11}	10^{-17}	G
29b.^{0}	20	20	9	7	1.00	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	38	22	1.00	1.73	10^{-12}	10^{-12}	10^{-24}	G
36a.^{0}	4	4	(4001)	(2374)	1.01	16.9	10^{-6}	10^{-6}	10^{-11}	F LIM.
45d.^{0}	8	8	5145	3241	1.02	15.3	10^{-14}	10^{-12}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	86	52	1.00	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	80	38	2.37	1.06	10^{-7}	10^{-12}	10^{-13}	G
23b.	10	11	407	204	1.01	0.500	10^{-3}	10^{-11}	10^{-11}	G
24a.	4	8	670	392	1.00	0.759	10^{-3}	10^{-12}	10^{-11}	G
35a.	8	8	39	24	1.00	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	41	18	1.00	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 15

(Algorithm with Orthogonality Constraints)

assume that $|\phi_i(x)| \leq |\phi_j(x)|$ **if** $i \leq j$

$tol \leftarrow \epsilon_M^{0.9}$

if $n \leq m$ **then** $maxcon \leftarrow n$ **else** $maxcon \leftarrow n-1$ **endif**

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$; $ncon \leftarrow 0$

while $|\phi_i(x)| \leq tol$ **and** $ncon \leq maxcon$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\nabla \phi_i^{\mathrm{T}} p = 0\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies** (6.5.1) - (6.5.2)

then $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$; $ncon \leftarrow ncon + 1$ **endif**

$i \leftarrow i + 1$

endwhile

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous but with a different criteria for accepting QP solutions as potential search directions.

Example 15

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	27	20	2.40	2.00	10^{-14}	10^{-13}	10^{-27}	a
21b.^{0}	20	20	209	104	1.00	4.47	10^{-13}	10^{-12}	10^{-27}	a
22b.^{0}	20	20	210	125	1.00	10^{-4}	10^{-9}	10^{-11}	10^{-17}	a
29b.^{0}	20	20	9	7	1.00	0.571	10^{-12}	10^{-12}	10^{-23}	a
35b.^{0}	9	9	38	22	1.00	1.73	10^{-12}	10^{-12}	10^{-24}	a
36a.^{0}	4	4	1011	518	2.99	50.0	10^{-10}	10^{-15}	10^{-20}	a
45d.^{0}	8	8	4688	2979	1.04	15.3	10^{-13}	10^{-11}	10^{-26}	a

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	7	3	1.00	1.08	10^{-4}	10^{-12}	10^{-14}	a
19.	11	65	86	52	1.00	9.38	10^{-1}	10^{-11}	10^{-8}	a
20d.	20	31	80	38	2.37	1.06	10^{-7}	10^{-12}	10^{-13}	a
23b.	10	11	179	89	1.21	0.500	10^{-3}	10^{-11}	10^{-11}	a
24a.	4	8	670	392	1.00	0.759	10^{-3}	10^{-12}	10^{-11}	a
35a.	8	8	39	24	1.00	1.65	10^{-1}	10^{-11}	10^{-9}	a
40g.	3	4	41	18	1.00	10^{-9}	10^{0}	10^{-11}	10^{-7}	a

Example 16

(Algorithm with Constraints Based on the QR Factorization)

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$

for $i = 1, 2, \ldots, \min\{m, n\}$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\bar{r}_i^{\mathrm{T}} p = -\bar{\phi}_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5) then** $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$ **endif**

$i \leftarrow i + 1$

endfor

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose inequality nonascent constraints based o QR factorization.

Example 16

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	116	49	7.18	2.00	10^{-12}	10^{-11}	10^{-25}	G
21b.^{0}	20	20	34	13	39.5	4.47	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
22b.^{0}	20	20	16	15	30.6	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	16.6	0.571	10^{-11}	10^{-11}	10^{-23}	G
35b.^{0}	9	9	46	16	13.7	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	(4001)	(629)	6.22	12.1	10^{-5}	10^{-5}	10^{-10}	F LIM.
45d.^{0}	8	8	77	37	13.3	15.3	10^{-14}	10^{-11}	10^{-28}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	1.60	1.08	10^{-4}	10^{-14}	10^{-14}	G
19.	11	65	37	23	12.2	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	39	20	9.95	1.11	10^{-8}	10^{-11}	10^{-16}	G
23b.	10	11	125	51	16.0	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	3601	697	8.36	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	131	50	15.1	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	(3004)	(516)	6.84	10^{-1}	10^{0}	10^{2}	10^{1}	F LIM.

Example 17

(Algorithm with Constraints Based on the QR Factorization)

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$

for $i = 1, 2, \ldots, \min\{m, n\}$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\bar{r}_i^T p = -\bar{\phi}_i\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.2) then** $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$ **endif**

$i \leftarrow i + 1$

endfor

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP solu
as potential search directions.

Example 17

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	104	44	7.30	2.00	10^{-14}	10^{-13}	10^{-27}	G
21b.^{0}	20	20	34	13	39.5	4.47	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
22b.^{0}	20	20	16	15	30.6	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	16.6	0.571	10^{-11}	10^{-11}	10^{-23}	G
35b.^{0}	9	9	146	37	13.4	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	103	43	6.35	45.8	10^{-8}	10^{-14}	10^{-16}	G
45d.^{0}	8	8	58	25	12.0	15.3	10^{-12}	10^{-11}	10^{-24}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	1.60	1.08	10^{-4}	10^{-14}	10^{-14}	G
19.	11	65	37	23	12.2	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	34	18	8.39	1.11	10^{-8}	10^{-11}	10^{-16}	G
23b.	10	11	105	38	17.5	0.500	10^{-2}	10^{-12}	10^{-11}	G
24a.	4	8	721	178	7.74	0.759	10^{-3}	10^{-12}	10^{-11}	G
35a.	8	8	1310	215	15.7	1.65	10^{-1}	10^{-4}	10^{-9}	X
40g.	3	4	(3004)	(516)	6.84	10^{-1}	10^{0}	10^{2}	10^{1}	F LIM.

Example 18

(Algorithm with Constraints Based on the QR Factorization)

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$

for $i = 1, 2, \ldots, \min\{m, n\}$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min_i\{-\bar{\phi}_i, 0\} \leq \nabla \bar{r}_i^T p \leq \max_i\{-\bar{\phi}_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.5) then** $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$ **endif**

$i \leftarrow i + 1$

endfor

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : This algorithm attempts to impose inequality nonascent constraints based on QR factorization.

Example 18

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	46	35	6.26	2.00	10^{-15}	10^{-14}	10^{-29}	ABS. F, G
21b.^{0}	20	20	50	31	18.9	4.47	10^{-13}	10^{-12}	10^{-26}	G
22b.^{0}	20	20	44	38	23.7	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	8	7	16.1	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	21	15	13.1	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	(4000)	(2598)	4.37	20.1	10^{-6}	10^{-7}	10^{-12}	F LIM.
45d.^{0}	8	8	(903)	(588)	19.2	36.3	10^{-1}	10^{-2}	10^{-2}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	1.60	1.08	10^{-4}	10^{-14}	10^{-14}	G
19.	11	65	55	36	19.7	9.38	10^{-1}	10^{-12}	10^{-8}	G
20d.	20	31	95	54	25.0	1.06	10^{-7}	10^{-11}	10^{-13}	G
23b.	10	11	185	106	13.5	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	140	104	5.35	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	44	34	12.5	1.65	10^{-1}	10^{-13}	10^{-9}	G
40g.	3	4	94	80	4.05	10^{-9}	10^{0}	10^{-14}	10^{-7}	G

Example 19

(Algorithm with Constraints Based on the QR Factorization)

repeat

compute the solution p_{QN} **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

$p^* \leftarrow p_{QN}$; $\mathcal{C}^* \leftarrow \emptyset$

for $i = 1, 2, \ldots, \min\{m, n\}$ **do**

$\tilde{\mathcal{C}} \leftarrow \mathcal{C}^* \cup \{\min_i\{-\bar{\phi}_i, 0\} \leq \nabla \bar{r}_i^T p \leq \max_i\{-\bar{\phi}_i, 0\}\}$

compute the solution $\tilde{p}$ **to**

$$\min_{p \in \Re^n} \bar{g}^T p + \frac{1}{2} p^T H p$$

subject to $\tilde{\mathcal{C}}$

if $\tilde{p}$ **satisfies (6.5.1) - (6.5.2) then** $p^* \leftarrow \tilde{p}$; $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$ **endif**

$i \leftarrow i + 1$

endfor

compute steplength α ; $x \leftarrow x + \alpha p^*$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Same as the previous example but with a different criteria for accepting QP soluti as potential search directions.

Example 19

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	46	35	6.26	2.00	10^{-15}	10^{-14}	10^{-29}	ABS. F, G
21b.^{0}	20	20	50	31	18.9	4.47	10^{-13}	10^{-12}	10^{-26}	G
22b.^{0}	20	20	44	38	23.7	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	8	7	16.1	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	21	15	13.1	1.73	10^{-11}	10^{-11}	10^{-22}	G
36a.^{0}	4	4	(4000)	(2560)	4.50	20.1	10^{-6}	10^{-7}	10^{-12}	F LIM.
45d.^{0}	8	8	468	289	14.7	15.3	10^{-14}	10^{-11}	10^{-29}	ABS. F, G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	1.60	1.08	10^{-4}	10^{-14}	10^{-14}	G
19.	11	65	55	36	19.7	9.38	10^{-1}	10^{-12}	10^{-8}	G
20d.	20	31	88	45	26.0	1.06	10^{-7}	10^{-11}	10^{-13}	G
23b.	10	11	171	88	12.1	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	125	81	5.07	0.759	10^{-3}	10^{-13}	10^{-11}	G
35a.	8	8	44	34	12.5	1.65	10^{-1}	10^{-13}	10^{-9}	G
40g.	3	4	215	205	4.46	10^{-9}	10^{0}	10^{-12}	10^{-7}	G

Example 20

(Levenberg-Marquardt Algorithm)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} **to**

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

$$\|p\|_2 \leq \delta$$

i.e. compute ω **as a function of** δ **and solve**

$$\min_{b;p} b^{\mathrm{T}} b + \omega p^{\mathrm{T}} p$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

compute steplength α ; $x \leftarrow x + \alpha p_{LM}$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : An implemetation of the Levenberg-Marquardt algorithm that uses a convex QP solver to compute the search direction. The parameter ω is computed from δ using software from MINPACK. The update for δ differs somewhat from MINPACK.

Example 20

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	65	38	1	2.00	10^{-14}	10^{-13}	10^{-28}	G
21b.^{0}	20	20	17	13	1	4.47	0.00	0.00	0.00	ABS. F, G
22b.^{0}	20	20	16	15	1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	4	3	1	0.571	10^{-14}	10^{-14}	10^{-28}	G
35b.^{0}	9	9	13	7	1	1.73	10^{-12}	10^{-12}	10^{-24}	G
36a.^{0}	4	4	2929	2913	1	23.7	10^{-6}	10^{-13}	10^{-13}	G
45d.^{0}	8	8	101	88	1	15.3	10^{-11}	10^{-11}	10^{-22}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	3	2	1	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	24	16	1	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	6	5	1	5.88	10^{-9}	10^{-12}	10^{-18}	G
23b.	10	11	44	28	1	0.500	10^{-2}	10^{-12}	10^{-11}	G
24a.	4	8	48	31	1	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	82	47	1	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	635	565	1	10^{-9}	10^{0}	10^{-10}	10^{-7}	G

Example 21

(Levenberg-Marquardt Algorithm with QR Factorization)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} to

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq \bar{R}p + \bar{f} \leq b$$

$$b \geq 0$$

$$\|p\|_2 \leq \delta$$

i.e. compute ω as a function of δ and solve

$$\min_{b;p} b^{\mathrm{T}} b + \omega p^{\mathrm{T}} p$$

subject to

$$-b \leq \bar{R}p + \bar{f} \leq b$$

$$b \geq 0$$

compute steplength α ; $x \leftarrow x + \alpha p_{LM}$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : A version of the previous example that uses the QR factorization.

Example 21

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	65	38	1	2.00	10^{-14}	10^{-13}	10^{-28}	G
21b.^{0}	20	20	17	13	1	4.47	10^{-15}	10^{-14}	10^{-31}	ABS. F, G
22b.^{0}	20	20	16	15	1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	4	3	1	0.571	10^{-14}	10^{-14}	10^{-28}	G
35b.^{0}	9	9	13	7	1	1.73	10^{-12}	10^{-12}	10^{-24}	G
36a.^{0}	4	4	2929	2913	1	23.7	10^{-6}	10^{-13}	10^{-13}	G
45d.^{0}	8	8	101	88	1	15.3	10^{-11}	10^{-11}	10^{-22}	G

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	3	2	1	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	24	16	1	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	6	5	1	5.88	10^{-9}	10^{-12}	10^{-18}	G
23b.	10	11	44	28	1	0.500	10^{-2}	10^{-12}	10^{-11}	G
24a.	4	8	48	31	1	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	82	47	1	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	636	566	1	10^{-9}	10^{0}	10^{-10}	10^{-7}	G

Example 22

(Generalized Levenberg-Marquardt Algorithm)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} **to**

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

$$\|p\|_2 \leq \delta$$

i.e. compute ω **as a function of** δ **and solve**

$$\min_{b;p} b^{\mathrm{T}} b + \omega p^{\mathrm{T}} p$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

compute the solution p^* **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to $\quad -\tilde{b} \leq Jp + f \leq \tilde{b}$

compute steplength α ; $x \leftarrow x + \alpha p^*$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : Uses the Levenberg-Marquardt computation only as a means of obtaining bounds for the QP subproblem that determines the search direction.

Example 22

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	64	37	3.35	2.00	10^{-15}	10^{-14}	10^{-30}	ABS. F, G
21b.^{0}	20	20	17	13	7.15	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
22b.^{0}	20	20	16	15	11.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	6.20	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	13	7	2.43	1.73	10^{-12}	10^{-12}	10^{-24}	G
36a.^{0}	4	4	(2929)	(2914)	2.05	23.7	10^{-6}	10^{-8}	10^{-13}	QP LIM.
45d.^{0}	8	8	222	129	10.1	15.3	10^{-16}	10^{-13}	10^{-31}	ABS. F, G, X

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	3.00	1.08	10^{-4}	10^{-13}	10^{-14}	G
19.	11	65	36	23	9.04	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	41	24	15.4	1.10	10^{-8}	10^{-10}	10^{-16}	$\bar{g}^T p \geq 0$
23b.	10	11	44	28	2.04	0.500	10^{-2}	10^{-12}	10^{-11}	G
24a.	4	8	52	34	3.38	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	70	38	1.63	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	174	142	2.35	10^{-5}	10^{0}	10^{-3}	10^{-7}	QP LIM.

Example 23

(Generalized Levenberg-Marquardt Algorithm using QR Factorization)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} **to**

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq \bar{R}p + \bar{f} \leq b$$

$$b \geq 0$$

$$\|p\|_2 \leq \delta$$

i.e. compute ω **as a function of** δ **and solve**

$$\min_{b;p} b^{\mathrm{T}} b + \omega p^{\mathrm{T}} p$$

subject to

$$-b \leq \bar{R}p + \bar{f} \leq b$$

$$b \geq 0$$

compute the solution $p^{\star}$ **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$$\text{subject to} \quad -\tilde{b} \leq \bar{R}p + \bar{f} \leq \tilde{b}$$

compute steplength α ; $x \leftarrow x + \alpha p^{\star}$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : A version of the previous example that uses the QR factorization.

Example 23

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	68	37	3.08	2.00	10^{-12}	10^{-11}	10^{-25}	G
21b.^{0}	20	20	17	13	5.23	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
22b.^{0}	20	20	27	23	10.7	10^{-5}	10^{-10}	10^{-9}	10^{-19}	$g^T p \geq 0$
29b.^{0}	20	20	6	5	4.20	0.571	10^{-16}	10^{-16}	10^{-33}	ABS. F, G
35b.^{0}	9	9	28	17	4.65	1.73	10^{-11}	10^{-11}	10^{-23}	G
36a.^{0}	4	4	2929	2913	1.70	23.7	10^{-6}	10^{-13}	10^{-13}	G
45d.^{0}	8	8	146	98	6.00	15.3	10^{-14}	10^{-11}	10^{-29}	ABS. F, G, X

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	6	3	2.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	27	18	5.17	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	10	8	9.63	1.10	10^{-8}	10^{-12}	10^{-16}	G
23b.	10	11	46	29	5.21	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	50	30	2.40	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	96	51	6.10	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	73	51	2.27	10^{-9}	10^{0}	10^{-15}	10^{-7}	G

Example 24

(Generalized Levenberg-Marquardt Algorithm with Box Constraints)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} **to**

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

$$\|p\|_\infty \leq \delta$$

compute the solution p^* **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to $\quad -\tilde{b} \leq Jp + f \leq \tilde{b}$

compute steplength α ; $x \leftarrow x + \alpha p^*$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : An implementation of the generalized Levenberg-Marquardt algorithm tha

constraints.

Example 24

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	69	38	3.74	2.00	10^{-13}	10^{-12}	10^{-25}	G
21b.^{0}	20	20	17	12	16.0	4.47	10^{-16}	10^{-14}	10^{-31}	ABS. F, G
22b.^{0}	20	20	16	15	11.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	6.20	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	15	8	2.63	1.73	10^{-12}	10^{-11}	10^{-23}	G
36a.^{0}	4	4	(2661)	(2558)	4.80	22.6	10^{-6}	10^{-7}	10^{-12}	QP LIM.
45d.^{0}	8	8	244	137	13.8	15.3	10^{-15}	10^{-11}	10^{-29}	ABS. F, G, X

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	3.00	1.08	10^{-4}	10^{-13}	10^{-14}	G
19.	11	65	37	23	9.96	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	6	6	23.7	1.10	10^{-8}	10^{-8}	10^{-16}	$\bar{g}^T p \geq 0$
23b.	10	11	70	45	7.36	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	53	33	7.27	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	83	44	8.68	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	224	127	2.57	10^{-9}	10^{0}	10^{-11}	10^{-7}	G

Example 25

(Modified Generalized Levenberg-Marquardt Algorithm)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} **to**

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

$$\|p\|_2 \leq \delta$$

i.e. compute ω **as a function of** δ **and solve**

$$\min_{b;p} b^{\mathrm{T}} b + \omega p^{\mathrm{T}} p$$

subject to

$$-b \leq Jp + f \leq b$$

$$b \geq 0$$

compute the solution $p^{\star}$ **to**

$$\min_{p \in \mathbb{R}^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

$$\text{subject to} \quad -\tilde{b} - \sqrt{\omega} \leq Jp + f \leq \tilde{b} + \sqrt{\omega}$$

compute steplength α ; $x \leftarrow x + \alpha p^{\star}$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : A modification of the generalized Levenberg-Marquardt algorithm that allo

bounds to go to ∞ with ω.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	101	57	2.49	2.00	10^{-13}	10^{-12}	10^{-26}	G
21b.^{0}	20	20	194	98	2.55	4.47	10^{-13}	10^{-12}	10^{-26}	G
22b.^{0}	20	20	16	15	11.1	10^{-4}	10^{-8}	10^{-11}	10^{-15}	G
29b.^{0}	20	20	6	5	6.20	0.571	10^{-12}	10^{-12}	10^{-23}	G
35b.^{0}	9	9	37	21	4.05	1.73	10^{-11}	10^{-11}	10^{-23}	G
36a.^{0}	4	4	(3271)	(1918)	2.11	15.7	10^{-6}	10^{-6}	10^{-11}	TIME
45d.^{0}	8	8	(2028)	(1238)	2.03	24.7	10^{-1}	10^{0}	10^{-2}	TIME

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	8	5	3.00	1.08	10^{-4}	10^{-13}	10^{-14}	G
19.	11	65	36	23	9.04	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	13	10	33.6	1.12	10^{-8}	10^{-8}	10^{-16}	$g^T p \geq 0$
23b.	10	11	70	37	6.68	0.500	10^{-2}	10^{-13}	10^{-11}	G
24a.	4	8	429	258	2.33	0.759	10^{-3}	10^{-11}	10^{-11}	G
35a.	8	8	45	23	3.13	1.65	10^{-1}	10^{-11}	10^{-9}	G
40g.	3	4	44	24	2.71	10^{-9}	10^{0}	10^{-9}	10^{-7}	$g^T p \geq 0$

Example 26

(Modified Generalized Levenberg-Marquardt Algorithm using QR Factorization)

$\delta \leftarrow 2 * (1 + \|x\|_2)$

repeat

compute the solution $\tilde{b}$; p_{LM} **to**

$$\min_{b;p} b^{\mathrm{T}} b$$

subject to

$$-b \leq \bar{R}p + \bar{f} \leq b$$

$$b \geq 0$$

$$\|p\|_2 \leq \delta$$

i.e. compute ω **as a function of** δ **and solve**

$$\min_{b;p} b^{\mathrm{T}} b + \omega p^{\mathrm{T}} p$$

subject to

$$-b \leq \bar{R}p + \bar{f} \leq b$$

$$b \geq 0$$

compute the solution $p^{\star}$ **to**

$$\min_{p \in \Re^n} \bar{g}^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H p$$

subject to $\quad -\tilde{b} - \sqrt{\omega} \leq \bar{R}p + \bar{f} \leq \tilde{b} + \sqrt{\omega}$

compute steplength α ; $x \leftarrow x + \alpha p^{\star}$; $\delta \leftarrow \alpha * \delta$

update f, $\bar{g}$, H

until termination criteria are satisfied

Remarks : A version of the previous example that uses the QR factorization.

Zero-Residual Problems

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
14.^{0}	4	6	99	56	2.39	2.00	10^{-12}	10^{-11}	10^{-23}	G
21b.^{0}	20	20	241	133	2.57	4.47	10^{-12}	10^{-11}	10^{-24}	G
22b.^{0}	20	20	27	23	10.7	10^{-5}	10^{-10}	10^{-9}	10^{-19}	$g^T p \geq 0$
29b.^{0}	20	20	6	5	4.20	0.571	10^{-16}	10^{-16}	10^{-33}	ABS. F, G
35b.^{0}	9	9	43	23	2.43	1.73	10^{-13}	10^{-12}	10^{-25}	G
36a.^{0}	4	4	(3519)	(2119)	2.09	16.2	10^{-6}	10^{-6}	10^{-11}	TIME
45d.^{0}	8	8	(1923)	(1160)	2.03	25.2	10^{-1}	10^{0}	10^{-2}	TIME

Problems with Nonzero Solutions

	n	m	f, J evals.	iters.	ave. QP iters.	$\|x^*\|_2$	$\|f^*\|_2$	$\|\bar{g}^*\|_2$	est. err.	conv.
9.	3	15	6	3	2.00	1.08	10^{-4}	10^{-12}	10^{-14}	G
19.	11	65	27	18	5.17	9.38	10^{-1}	10^{-11}	10^{-8}	G
20d.	20	31	55	30	22.9	1.11	10^{-8}	10^{-10}	10^{-16}	$g^T p \geq 0$
23b.	10	11	74	39	7.41	0.500	10^{-2}	10^{-11}	10^{-11}	G
24a.	4	8	272	155	2.46	0.759	10^{-3}	10^{-12}	10^{-11}	G
35a.	8	8	48	24	3.00	1.65	10^{-1}	10^{-12}	10^{-9}	G
40g.	3	4	46	26	2.62	10^{-9}	10^{0}	10^{-10}	10^{-7}	X

6.6 Conclusions and Future Work

In this dissertation, we have proposed new algorithms for nonlinear least squares that solve quadratic programming subproblems, using techniques motivated by non-asymptotic as well as asymptotic considerations. Our approach differs substantially from previous methods, because information about the individual residuals and interrelationships between them can be taken into account in formulating subproblems. Moreover, convergence properties of the new methods are generally as good or better than for quasi-Newton methods for unconstrained optimization, because in some instances only a projection of the Hessian need be positive definite in order to achieve superlinear convergence. Preliminary results are promising, and there is every reason to believe that these algorithms will prove useful in practice.

There is much scope for further development of the algorithms introduced in this dissertation. One possibility would be to investigate special quasi-Newton updates for the new methods, perhaps using ideas from projected updates schemes for constrained optimization (see Nocedal and Overton [1985]). Although we were able to use the BFGS update as for unconstrained optimization without modification in our numerical tests, there may be a different update procedure that would give better overall performance. A second possibility for improvement is to combine the formulation and solution of the QP subproblems. For our numerical tests, we first posed QP subproblems and then solved them with existing QP software. An alterative would be to design QP-like solvers for nonlinear least squares that have the capability of internally selecting and modifying constraint sets. For example, constraints that would cause ill-conditioning when added to the QP working set (see, for example, Gill et al. [1986a]) could be dropped or altered, on account of the redundancy that exists in nonlinear least squares between the QP objective and constraints.

Another promising direction for future research is the extension of the new techniques for nonlinear least squares to SQP methods for constrained optimization. Much of the motivation for formulating QP subproblems discussed in Sections 6.4 and 6.5 carries over in a straightforward way to general nonlinear programming problems. A major difference is that the QP objective must explicitly approximate the Lagrangian function, since the associated Lagrange multipliers will generally be nonzero. In particular, the methods that use a QP

to compute bounds for the subproblems (Section 6.5.2) may be extended to trust-region methods for constrained optimization.

Finally, because there is flexibility within the algorithms for taking into account special features of particular problems, it may be possible to develop versions of the new methods that work well for specific problem categories.

6.7 Bibliography

Benveniste, R. "A Quadratic Programming Algorithm using Conjugate Search Directions", *Mathematical Programming* **16** (1979) 63–80.

Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press (1982).

Best, M. J., "Equivalence of some Quadratic Programming Algorithms", *Mathematical Programming* **30** (1984) 71–87.

Bartholomew-Biggs, M. C., "Line Search Procedures for Nonlinear Programming Algorithms with Quadratic Programming Subproblems", Technical Report No. 116, Numerical Optimisation Centre, The Hatfield Polytechnic (May 1981).

Bunch, J. R., and L. Kaufman, "A Computational Method for the Indefinite Quadratic Programming Problem", *Linear Algebra and its Applications* **34** (1980) 341–370.

Coleman, T. F., and D. C. Sorensen, "A Note on the Computation of an Orthogonal Basis for the Null Space of a Matrix", *Mathematical Programming* **29** (1984) 234–242.

Fiacco, A. V., and G. P. McCormick, *Nonlinear Programming*, Wiley (1968).

Fletcher, R., *Practical Methods of Optimization Vol. 2 : Constrained Optimzation*, Wiley (1981).

Fletcher, R., "Penalty Functions", in *Mathematical Programming—The State of the Art—Bonn 1982*, A. Bachem, M. Grötschel, and B. Korte (eds.), Springer-Verlag (1983) 87–144.

Fletcher, R., "Recent Developments in Linear and Quadratic Programming", in *The State of the Art in Numerical Analysis*, A. Iserles and M. J. D. Powell (eds.), Oxford Clarendon (1987).

Gill, P. E., and W. Murray, "Nonlinear Least Squares and Nonlinearly Constrained Optimization", in *Numerical Analysis — Proceedings Dundee 1975*, *Lecture Notes in Mathematics*, Vol. **506**, Springer-Verlag (1976) 134–147.

Gill, P. E., and W. Murray, "Algorithms for the Solution of the Nonlinear Least-Squares Problem", SIAM *Journal on Numerical Analysis*, Vol. **15** No. 5 (October 1978a) 977–992.

Gill, P. E., and W. Murray, "Numerically Stable Methods for Quadratic Programming", *Mathematical Programming* **14** (1978b) 349–372.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Two Steplength Algorithms for Numerical Optimization", Technical Report SOL 79-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University (1979).

Gill, P. E., W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press (1981).

Gill, P. E., N. I. M. Gould, W. Murray, M. A. Saunders, and M. H. Wright, "A Weighted Gram-Schmidt Method for Convex Quadratic Programming", *Mathematical Programming* **30** (1984a) 176–195.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "User's Guide for QPSOL (Version 3.2) : A Fortran Package for Quadratic Programming", Technical Report SOL 84-6, Systems Optimization Laboratory, Department of Operations Research, Stanford University (September 1984b).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Software and its Relationship to Methods", in *Numerical Optimization 1984*, P. T. Boggs, R. H. Byrd, and R. B. Schnabel (eds.), SIAM (1985), 139–159.

Gill, P. E., S. J. Hammarling, W. Murray, M. A. Saunders, and M. H. Wright, "User's Guide for LSSOL (Version 1.0): A Fortran Package for Constrained Linear Least-Squares and Convex Quadratic Programming", Technical Report SOL 86-1, Systems Optimization Laboratory, Department of Operations Research, Stanford University (January 1986a).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Users Guide for NPSOL (Version 4.0) : A Fortran Package for Nonlinear Programming", Technical Report SOL 86-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University (January 1986b).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Users Guide for NPSOL (Version 4.0) : A Fortran Package for Nonlinear Programming", Technical Report SOL 86-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University (January 1986b).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Model Building and Practical Aspects of Nonlinear Programming", in NATO ASI Series, Vol. **F15**, *Computational Mathematical Programming*, K. Schittkowski (ed.), Springer-Verlag (1985) 209–247.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "Some Theoretical Properties of an Augmented Lagrangian Function", Technical Report SOL 86-6R, SOL(September 1986c).

Goldfarb, D., and A. Idnani, "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs", *Mathematical Programming* **27** (1983) 1–33.

Goodman, J., "Newton's Method for Constrained Optimization", *Mathematical Programming* 33 (1985) 162–177.

Gould, N. I. M., "On Practical Conditions for the Existence and Uniqueness of Solutions to General Equality Quadratic Programming Problems", *Mathematical Programming* 32 (1985) 90–99.

Gurwitz, C. B., "Sequential Quadratic Programming Methods Based on Approximating a Projected Hessian Matrix", Technical Report # 219, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University (May 1986).

Hanson, R. J., and K. H. Haskell, "ALGORITHM 587 : Two Algorithms for the Linearly Constrained Least Squares Problem", ACM *Transactions on Mathematical Software*, Vol. 8 No. 3 (September 1982) 323–333.

Hanson, R. J., "Linear Least Squares with Bounds and Linear Constraints", SIAM *Journal on Scientific and Statistical Computing*, Vol. 7 No. 3 (July 1986) 826–834.

Han, S.-P., "Solving Quadratic Programs by an Exact Penalty Function", in *Nonlinear Programming 4*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), Academic Press (1981) 25–55.

Haskell, K. H., and R. J. Hanson, "An Algorithm for Linear Least Squares Problems with Equality and Nonnegativity Constraints", *Mathematical Programming* 21 (1981) 98–118.

Hoyle, S. C., "A Single Phase Method for Quadratic Programming", Technical Report SOL 86-9, Systems Optimizations Laboratory, Department of Operations Research, Stanford University (April 1986).

Moré, J. J., "The Levenberg-Marquardt Algorithm: Implementation and Theory", in *Numerical Analysis — Proceedings Dundee 1977, Lecture Notes in Mathematics*, Vol. 630, Springer-Verlag (1978) 105–116.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, "User Guide for MINPACK-1", Technical Report ANL-80-74, Argonne National Laboratory (1980).

Morrison, D. D., "Optimization by Least Squares", SIAM *Journal on Numerical Analysis*, Vol. 5 No. 1 (March 1968) 83–88.

Murray, W., and M. H. Wright, "Computation of the Search Direction in Constrained Optimization Algorithms", *Mathematical Programmming Study* 16 (1982) 62–83.

Nocedal, J., and M. L. Overton, "Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization", SIAM *Journal on Numerical Analysis*, Vol. 22 No. 5 (October 1985).

PORT Mathematical Subroutine Library, A. T. & T. Bell Laboratories, Inc., (May 1984).

Powell, M. J. D., "An Upper Triangular Matrix Method for Quadratic Programming", in *Nonlinear Programming 4*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), Academic Press (1981) 1–24.

Powell, M. J. D, "ZQPCVX : A Fortran Subroutine for Convex Quadratic Programming", Technical Report DAMTP 1983/NA17, Department of Applied Mathematics and Theoretical Physics, Cambridge University (1983b).

Powell, M. J. D., "On the Quadratic Programming Algorithm of Goldfarb and Idnani", *Mathematical Programming Study* **25** (1985) 46–61.

Robinson, S. M., "Perturbed Kuhn-Tucker Points and Rates of Convergence for a Class of Nonlinear Programming Algorithms", *Mathematical Programming* **7** (1974).

Sacher, R. S., "A Decomposition Algorithm for Quadratic Programming", *Mathematical Programming* **18** (1980) 16–30.

Schittkowski, K., and J. Stoer, "A Factorization Method for the Solution of Constrained Linear Least Squares Problems", *Numerische Mathematik* **31** (1979) 431–463.

Stoer, J., "On the Numerical Solution of Constrained Least-Squares Problems", SIAM *Journal on Numerical Analysis*, Vol. 8 No. 2 (June 1971) 382–411.

Stoer, J., "Principles of Sequential Quadratic Programming Methods for Solving Nonlinear Programs", in NATO ASI Series, Vol. **F15**, *Computational Mathematical Programming*, K. Schittkowski (ed.), Springer-Verlag (1985) 165–207.

Wright, M. H., and S. C. Glassman, "Fortran Subroutines to Solve the Linear Least Squares Problem and Compute the Complete Orthogonal Factorization", Technical Report SOL 78-8, Department of Operations Research, Stanford University (April 1978).

Appendix : Test Problems

Superscripts on problem numbers have the following interpretation :

^{0} : zero-residual problem
^{L} : linear least-squares problem

Problems from Moré, Garbow, and Hillstrom [1981]

	n	m	
1.^{0}	2	2	**Rosenbrock**
2.^{0}	2	2	**Freudenstein and Roth**
3.^{0}	2	2	**Powell Badly Scaled**
4.^{0}	2	3	**Brown Badly Scaled**
5.^{0}	2	3	**Beale**
6.	2	10	**Jennrich and Sampson**
7.^{0}	3	3	**Helical Valley**
8.	3	15	**Bard**
9.	3	15	**Gaussian**
10.	3	16	**Meyer**
11.^{0}	3	10	**Gulf Research and Development†**
12.^{0}	3	10	**Box 3-Dimensional**
13.^{0}	4	4	**Powell Singular**
14.^{0}	4	6	**Wood**
15.	4	11	**Kowalik and Osborne**
16.	4	20	**Brown and Dennis**
17.	5	33	**Osborne 1**
18.^{0}	6	13	**Biggs EXP6‡**

† For the **Gulf Research and Development Function (# 11)**, the formula

$$\phi_i(x) = \exp\left[-\frac{|y_i\ mi\ x_2|^{x_3}}{x_1}\right] - t_i$$

given in Moré, Garbow, and Hillstrom [1981] for the residual functions is in error. The correct formula is

$$\phi_i(x) = \exp\left[-\frac{|y_i - x_2|^{x_3}}{x_1}\right] - t_i$$

(see Moré, Garbow, and Hillstrom [1978]).

‡ For the **Biggs EXP6 Function (# 18)**, the minmum value for the sum of squares is given in Moré, Garbow, and Hillstrom [1981] as $5.65565\ldots \times 10^{-3}$. It can be easily verified that the residuals vanish at several points (for example $(1, 10, 1, 5, 4, 3)$).

Problems from Moré, Garbow, and Hillstrom [1981] (continued)

	n	m	
19.	11	65	**Osborne 2†**
20a.	6	31	**Watson**
20b.	9	31	**Watson**
20c.	12	31	**Watson**
20d.	20	31	**Watson**
21a.^{0}	10	10	**Extended Rosenbrock**
21b.^{0}	20	20	**Extended Rosenbrock**
22a.^{0}	12	12	**Extended Powell Singular**
22b.^{0}	20	20	**Extended Powell Singular**
23a.	4	5	**Penalty I**
23b.	10	11	**Penalty I**
24a.	4	8	**Penalty II**
24b.	10	20	**Penalty II**
25a.^{0}	10	12	**Variably Dimensioned**
25b.^{0}	20	22	**Variably Dimensioned**
26a.^{0}	10	10	**Trigonometric**
26b.^{0}	20	20	**Trigonometric**
27a.^{0}	10	10	**Brown Almost Linear**
27b.^{0}	20	20	**Brown Almost Linear**
28a.^{0}	10	10	**Discrete Boundary Value**
28b.^{0}	20	20	**Discrete Boundary Value**
29a.^{0}	10	10	**Discrete Integral**
29b.^{0}	20	20	**Discrete Integral**
30a.^{0}	10	10	**Broyden Tridiagonal**
30b.^{0}	20	20	**Broyden Tridiagonal**
31a.^{0}	10	10	**Broyden Banded**
31b.^{0}	20	20	**Broyden Banded**
32.^{L}	10	20	**Linear — Full Rank**
33.^{L}	10	20	**Linear — Rank 1**
34.^{L}	10	20	**Linear — Rank 1 with Zero Columns and Rows**
35a.	8	8	**Chebyquad**
35b.^{0}	9	9	**Chebyquad**
35c.	10	10	**Chebyquad**

† For **Osborne's Second Function (# 19)**, the value of $f(x^*)$ is given (to six figures) in Moré, Garbow, and Hillstrom [1981] as 4.01377×10^{-2}. The smallest value we were able to obtain was 4.01683×10^{-2}.

Appendix : Test Problems

Matrix Square Root Problems

	n	m	
36a.^{0}	4	4	**Matrix Square Root 1**
36b.^{0}	9	9	**Matrix Square Root 2**
36c.^{0}	9	9	**Matrix Square Root 3**
36d.^{0}	9	9	**Matrix Square Root 4**

These test problems come from a private communication of S. Hammarling to P. E. Gill in 1983.

	MATRIX	SQUARE ROOT
36a.^{0}	$\begin{pmatrix} 10^{-4} & 1 \\ 0 & 10^{-4} \end{pmatrix}$	$\begin{pmatrix} 10^{-2} & 50 \\ 0 & 10^{-2} \end{pmatrix}$
36b.^{0}	$\begin{pmatrix} 10^{-4} & 1 & 0 \\ 0 & 10^{-4} & 0 \\ 0 & 0 & 10^{-4} \end{pmatrix}$	$\begin{pmatrix} 10^{-2} & 50 & 0 \\ 0 & 10^{-2} & 0 \\ 0 & 0 & 10^{-2} \end{pmatrix}$
36c.^{0}	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
36d.^{0}	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$

• The identity matrix was used as the starting value in all instances. Note that the iteration should not be started with the zero matrix because it is a stationary point of the sum of squares.

Problems from Salane [1987]

	n	m	
37.	2	16	**Hanson 1**
38.	3	16	**Hanson 2**

Appendix : Test Problems

Problems from McKeown [1975a] (also McKeown [1975b])

	n	m		μ
39a.	2	3	**McKeown 1**	0.001
39b.	2	3	**McKeown 1**	0.01
39c.	2	3	**McKeown 1**	0.1
39d.	2	3	**McKeown 1**	1.0
39e.	2	3	**McKeown 1**	10.0
39f.	2	3	**McKeown 1**	100.0
39g.	2	3	**McKeown 1**	1000.0
40a.†	3	4	**McKeown 2**	0.001
40b.†	3	4	**McKeown 2**	0.01
40c.†	3	4	**McKeown 2**	0.1
40d.†	3	4	**McKeown 2**	1.0
40e.†	3	4	**McKeown 2**	10.0
40f.†	3	4	**McKeown 2**	100.0
40g.†	3	4	**McKeown 2**	1000.0
41a.	5	10	**McKeown 3**	0.001
41b.	5	10	**McKeown 3**	0.01
41c.	5	10	**McKeown 3**	0.1
41d.	5	10	**McKeown 3**	1.0
41e.	5	10	**McKeown 3**	10.0
41f.	5	10	**McKeown 3**	100.0
41g.	5	10	**McKeown 3**	1000.0

† In the data defining this problem given in McKeown [1975a] and [1975b], the matrix

$$B = \begin{pmatrix} 2.95137 & 4.87407 & -2.0506 \\ 4.87407 & 9.39321 & -3.93181 \\ -2.0506 & -3.93189 & 2.64745 \end{pmatrix}$$

is in error (it should be symmetric). The value

$$B = \begin{pmatrix} 2.95137 & 4.87407 & -2.0506 \\ 4.87407 & 9.39321 & -3.93189 \\ -2.0506 & -3.93189 & 2.64745 \end{pmatrix},$$

which is correct to six decimal digits, was used in our formulation of the problem.

Problems from DeVilliers and Glasser [1981] (also Salane [1987])

	n	m		starting value
42a.^{0}	4	24	**DeVilliers and Glasser 1**	(1.0, 8.0, 4.0, 4.412)
42b.^{0}	4	24	**DeVilliers and Glasser 1**	(1.0, 8.0, 8.0, 1.0)
42c.^{0}	4	24	**DeVilliers and Glasser 1**	(1.0, 8.0, 1.0, 4.412)
42d.^{0}	4	24	**DeVilliers and Glasser 1**	(1.0, 8.0, 4.0, 1.0)
43a.^{0}	5	16	**DeVilliers and Glasser 2**	(45.0, 2.0, 2.5, 1.5, 0.9)
43b.^{0}	5	16	**DeVilliers and Glasser 2**	(42.0, 0.8, 1.4, 1.8, 1.0)
43c.^{0}	5	16	**DeVilliers and Glasser 2**	(45.0, 2.0, 2.1, 2.0, 0.9)
43d.^{0}	5	16	**DeVilliers and Glasser 2**	(45.0, 2.5, 1.7, 1.0, 1.0)
43e.^{0}	5	16	**DeVilliers and Glasser 2**	(35.0, 2.5, 1.7, 1.0, 1.0)
43f.^{0}	5	16	**DeVilliers and Glasser 2**	(42.0, 0.8, 1.8, 3.15, 1.0)

Problems from Dennis, Gay, and Vu [1985]

	n	m		starting value
44a.^{0}†	6	6	**Exp. 791129**	(.299, −0.273, −.474, .474, −.0892, .0892)‡
44b.^{0}†	6	6	**Exp. 791226**	(−.3, .3, −1.2, 2.69, 1.59, −1.5)
44c.^{0}†	6	6	**Exp. 0121a**	(−.041, .03, −2.565, 2.565, −.754, .754)‡
44d.^{0}†	6	6	**Exp. 0121b**	(−.056, .026, −2.991, 2.991, −.568, .568)
44e.^{0}†	6	6	**Exp. 0121c**	(−.074, .013, −3.632, 3.632, −.289, .289)
45a.^{0}	8	8	**Exp. 791129**	(.299, .186, −0.273, .0254, −0.474, −.0892, .0892)‡
45b.^{0}	8	8	**Exp. 791226**	(−.3, −.39, .3, −.344, −1.2, 2.69, 1.59, −1.5)
45c.^{0}	8	8	**Exp. 0121a**	(−.041, −.775, .03, −.047, −2.565, 2.565, −.754, .754)‡
45d.^{0}	8	8	**Exp. 0121b**	(−.056, −.753, .026, −.047, −2.991, 2.991, −.568, .568)
45e.^{0}	8	8	**Exp. 0121c**	(−.074, −.733, .013, −.034, −3.632, 3.632, −.289, .289)

† Variables x_2 and x_4 (b and d in Dennis, Gay, and Vu [1985]) are eliminated from the linear constraints in order to get the 6-variable formulation of the problem (see Dennis, Gay, and Vu [1985]).

‡ Specification of some starting values in Dennis, Gay, and Vu [1985] is incomplete. The correct values were obtained from D. M. Gay in 1986.

Appendix : Test Problems

References :

Dennis, J. E., D. M. Gay, and P. A. Vu, "A New Nonlinear Equations Test Problem", Technical Report 83-16, Mathematical Sciences Department, Rice University (June 1983 — revised May 1985).

DeVilliers, N., and D. Glasser, "A Continuation Method for Nonlinear Regression", SIAM *Journal on Numerical Analysis*, Vol. 18 No. 6 (December 1981) 1139–1154.

McKeown, J. J., "Specialized versus General-Purpose Algorithms for Minimising Functions that are Sums of Squared Terms", *Mathematical Programming* 9 (1975a) 57–68.

McKeown, J. J., "On Algorithms for Sums of Squares Problems", Paper 14 in *Towards Global Optimization*, L. C. W. Dixon and G. Szegö (eds.), North Holland (1975b) 229–257.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization Software", Technical Report TM-324, Applied Mathematics Division, Argonne National Laboratory (July 1978).

Moré, J. J., B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization Software", ACM *Transactions on Mathematical Software*, Vol. 7 No. 1 (March 1981) 17–41.

Salane, D. E., "A Continuation Approach for Solving Large Residual Nonlinear Least Squares Problems", Sandia National Laboratories, Albuquerque, NM (1986) [to appear in SIAM *Journal on Scientific and Statistical Computing*].