

Reducible Nonlinear Programming Problems

by

Teresa Anne Parks

Technical Report 85-8, May 1985

RICE UNIVERSITY

REDUCIBLE NONLINEAR PROGRAMMING PROBLEMS


by

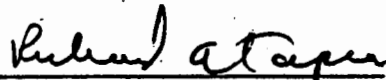
TERESA ANNE PARKS


A Thesis Submitted
In Partial Fulfillment Of The
Requirements For The Degree

DOCTOR OF PHILOSOPHY

Approved, Thesis Committee:


John E. Dennis Jr.,
Professor of Mathematical Sciences
Joint Chairman


Richard A. Tapia,
Professor of Mathematical Sciences
Joint Chairman


Richard D. Young,
Professor of Economics

Houston, Texas

May, 1985

Dedication

This thesis is dedicated to my parents, Edward Parks and Irene B. Parks.

Acknowledgements

I would like to thank first my committee: Dr. John Dennis, Dr. Richard Tapia, and Dr. Richard Young, for their assistance with this thesis and also for the valuable courses which they have presented. In addition, I would like to express my appreciation to Dr. Trond Steihaug, Dr. Francesco Zirilli, and Dr. Paul Pfeiffer. Special thanks must go to Velva Power, Vivian Choi, and Judy Forney for the work they have put into keeping the department running and for looking after the graduate students.

Mathematical and personal thanks go to Edward Dean, Robert Myers, Neely Atkinson, Maria Rosa Celis, Michael Pearlman, Robert Kelley, and Kathryn Turner. And I particularly want to thank John Quirein for asking me to find out about sensitivity theory for nonlinear programming problems.

I would also like to express my appreciation to Exxon Pipeline Company, which supported the beginning of my graduate work through their Employee Educational Assistance Program. I am grateful to Rice University for providing me with financial support, both from University funds and through access to various grants. This research was supported by ARO DAAG-29-83-K-0035 and NSF MCS81-16770.

CHAPTER 1

Introduction and Synopsis

1.1. Introduction

In this thesis we will be concerned with nonlinear programming problems having a particular form. In the unconstrained case, they can be written

$$\underset{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^k}{\text{minimize}} \quad f(x, \alpha), \quad (1.1)$$

indicating that there is some natural distinction between the groups of variables. The variables which will comprise the vector x are chosen by the requirement that, given a fixed k -vector $\hat{\alpha}$, we must be able to solve the (n -dimensional) problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x, \hat{\alpha}) \quad (1.2)$$

cheaply and accurately. For certain functions f , the solution of problem (1.2) provides an explicit formula expressing x in terms of α , and we may use it to eliminate x from the original problem entirely.

In a general minimization problem, every feasible vector is a candidate for a minimizer. The particular solution technique chosen to perform the

minimization will determine which of these points will be investigated. For our special case, we can use problem (1.2) to specify a relationship between the x -variables and the α -variables at the solution. It seems reasonable, then, to restrict our search for a minimizer to only those points (x, α) in \mathbb{R}^{n+k} at which x and α satisfy the given relationship. This is equivalent to adding a constraint to the problem. In the simplest cases, x can be written as an explicit function of α , and the requirement $x = g(\alpha)$ can be used to rewrite the problem as a function of α alone. In the more general case, the relation between x and α is implicit, and it may not even be possible to express it in the form of a nonlinear programming problem constraint. Yet, under suitable conditions, the original problem may still be reduced to a problem in α alone.

A problem of the form (1.1) for which it is possible to define a relationship expressing x in terms of α will be called **reducible**. To solve a problem of this type, we solve instead the **reduced problem**

$$\underset{\alpha \in \mathbb{R}^k}{\text{minimize}} \quad f(\alpha) = f(x(\alpha), \alpha) \quad (1.3)$$

This is a problem in fewer variables, but now every function evaluation requires the solution of a subproblem of the form (1.2).

The existence of a minimization subproblem in the function evaluation does not require any modifications in the routine used to solve problem (1.3). If this routine requires derivative information, however, we must be able to compute the gradient and possibly the Hessian of the reduced functional f . We will give

REDUCIBLE NONLINEAR PROGRAMMING PROBLEMS

TERESA ANNE PARKS

ABSTRACT

In this thesis we are concerned with general nonlinear programming problems in which the variables can be naturally separated into two groups. This separation has the property that if the variables in one of the groups are fixed, then it is easy to solve the problem in terms of the unspecified variables.

The existing work in nonlinear least squares has been extended to handle inequality constraints and further extended to certain classes of nonlinear programming problems which have special structure.

TABLE OF CONTENTS

Chapter 1	Introduction and Synopsis	1
Chapter 2	Mathematical Background	9
Chapter 3	The Semilinear Least Squares Problem	28
Chapter 4	Reducible Nonlinear Programming Problems	44
Chapter 5	Some Special Cases	65
Chapter 6	Constrained Semilinear Least Squares	74
Chapter 7	Extension to Problems Without Special Structure	84
Chapter 8	The Solution of Reducible Nonlinear Programming Problems	89
References	94

conditions under which this can be done and derive expressions for the gradient and Hessian of f . The expression for the gradient will be shown to be particularly simple. In fact, for unconstrained problems and for constrained problems of a certain type, we will prove that

$$\nabla f(\alpha) = \nabla_{\alpha} f(x(\alpha), \alpha).$$

Since the determination of $x(\alpha)$ is required by the function evaluation, the gradient of f is available for only the amount of work required to evaluate the partial derivatives of f with respect to α at the point $(x(\alpha), \alpha)$.

Although the expressions for the gradient and Hessian of the reduced functional allow us to use existing software immediately, they can also be used to suggest modified algorithms which take fuller advantage of the special structure of specific problems. This can be important if these problems are to be solved repeatedly or if they are hard to solve using standard techniques.

1.2. Example

Note: We include this example merely to illustrate our suggested approach.

We do not mean to imply that such dramatic success is to be expected.

Suppose that we need to solve the following problem and that the only minimization routine available is an implementation of the steepest descent method.

original problem:

$$\underset{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^m}{\text{minimize}} \quad f(x, \alpha) = \sum_{j=1}^m \left[100 (x_j - \alpha_j)^2 + (1 - \alpha_j)^2 \right]$$

This is Rosenbrock's function [51], extended to the case of $2m$ variables. It is well known to be an extremely difficult problem for steepest descent.

Instead of solving the problem directly, we will take advantage of its special structure. Since f is quadratic in x , it would be very easy to determine the optimal x if the optimal α were known. If we require that x and α always satisfy the relationship that $x(\alpha)$ is the value of x that minimizes f for the given value of α , then we can rewrite the problem as a function of α alone, hiding the computation of x in the objective function evaluation.

alternate formulation:

$$\underset{\alpha \in \mathbb{R}^m}{\text{minimize}} \quad f(\alpha) = \sum_{j=1}^m \left[100 [x(\alpha_j) - \alpha_j]^2 + (1 - \alpha_j)^2 \right]$$

where $x(\alpha)$ solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad t(x) = f(x; \alpha) = \sum_{j=1}^m \left[100 (x_j - \alpha_j)^2 + (1 - \alpha_j)^2 \right]$$

For a given value of α , define $[x(\alpha)]_j = \alpha_j^2$. Then $x(\alpha)$ satisfies the second-order sufficient conditions for a minimizer of the subproblem. Since the dependence of x on α is explicit, we may simply substitute α_j^2 for $[x(\alpha)]_j$ to avoid excess computation.

After this substitution, the alternate problem reduces to

$$\underset{\alpha \in \mathbb{R}^n}{\text{minimize}} \quad f(\alpha) = \sum_{j=1}^m (1 - \alpha_j)^2$$

which can be solved by steepest descent in a single iteration.

The two-variable Rosenbrock function is difficult for certain methods because it has a curved valley following the parabola $x = \alpha^2$. In our alternate formulation, we have constrained our search so that the only candidates for minimizers which we will consider are precisely those which lie on this curve.

In this problem, forming the alternate problem not only reduced the problem dimension but simplified its form. Unfortunately, this is an extremely rare occurrence. In general, expressing the problem as a function of α alone makes it more complicated and more nonlinear.

1.3. Synopsis

Chapter 2 contains the mathematical background material which will be used in subsequent chapters.

Chapter 3 is an introduction to the semilinear (or separable) least squares problem. This problem has been studied by a number of authors, [3], [8], [21], [22], [23], [24], [27], [28], [29], [30], [35], [36], [43], [46], [49], [50], [52], [54], [57], [58], [59]. We will use the simple Gauss-Newton method to provide a framework for the characterization of several of these solution techniques. We will show that certain of these methods can be considered either in terms of solution of the full problem or of the reduced problem. Two methods, that of Barham and Drane [3] and that of Kaufman [27], which appear dissimilar and which were derived from quite different viewpoints, will be shown to be based on identical local quadratic models. If these two methods were to be implemented using the same globalization strategy for handling poor initial guesses, then they would generate the same iterates.

The idea of a reduced problem was introduced for the unconstrained case in section 1.1. In chapter 4, we make a more rigorous definition and extend it to a general nonlinear programming problem. A theorem from sensitivity theory for nonlinear programming problems will be used to provide conditions under which there exists a unique differentiable function $x(\alpha)$ defined in a neighborhood of a given $\hat{\alpha}$. We then derive an expression for $\frac{dx}{d\alpha}$ and use it to express the gradient and Hessian of the reduced functional in terms of information from the

original problem and from the subproblem which defines $x(\alpha)$.

The development in chapter 4 is entirely theoretical and ignores our original stipulation that it must be easy to determine $x(\alpha)$. Chapter 5 contains some special cases of the general reducible nonlinear programming problem. These are chosen to provide examples of the application of the equations of chapter 4 and are cases in which the structure of the subproblem defining $x(\alpha)$ makes it easy to solve.

In chapter 6, we return to the semilinear least squares problem, defining a general constrained version and analyzing it using the techniques of chapter 4. We first demonstrate that the reduced functional in the constrained case has the same form as the reduced functional in the unconstrained case. The gradient and Hessian of f are easily represented as special cases of the formulas from chapter 4. In addition we will derive an expression for the Jacobian of the residual of the reduced problem. The Jacobians of the unconstrained and constrained semilinear least squares problems will be shown to have the same basic structure. This will suggest direct extension of the solution techniques for unconstrained problems to the constrained case.

Chapter 7 contains a modification of the problem formulation ideas presented in previous chapters. We will drop the assumption that there is a natural partition of the variables and allow the partitioning to be redefined at any iteration. We will, however, impose sufficient restrictions on the partitioning to maintain our requirement that the subproblem be easy to solve.

This modification will be shown to be the problem formulation used by the generalized reduced gradient method. Hence, the generalized reduced gradient method may be thought of as defining a sequence of locally reducible problems.

Finally, in chapter 8, we consider the application of the concepts of the previous chapters to the solution of nonlinear programming problems.

1.4. Notation

Throughout the paper $\| \cdot \|$ will refer to the usual 2-norm.

CHAPTER 2

Mathematical Background

2.1. Rate of Convergence

(See Ortega and Rheinboldt [44]).

Let $\{x_k\}$ be a sequence of vectors in \mathbb{R}^n .

$\{x_k\}$ converges to x_* if

$$\lim_{k \rightarrow \infty} \|x_k - x_*\| = 0.$$

Let $\{x_k\}$ be a sequence which converges to x_* .

2.1.1. Root Convergence Rates

A sequence $\{x_k\}$ converges with r -order p if p is the largest number such that

$$0 \leq \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|^p} < \infty.$$

The value of p that occurs in that limit is the asymptotic convergence rate.

If $p=1$, the convergence is r -linear. If $p=2$, the convergence is r -quadratic.

The sequence $\{x_k\}$ converges r -superlinearly if

$$\limsup_{k \rightarrow \infty} \|x_k - x_*\|^{\frac{1}{k}} = 0.$$

2.1.2. Quotient Convergence Rates

$\{x_k\}$ converges q -linearly if there is some constant $c \in [0,1)$ and an integer

$\hat{k} > 0$ such that

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

$\{x_k\}$ converges q -superlinearly if there exists a sequence $\{c_k\}$ converging to 0 such that

$$\|x_{k+1} - x_*\| \leq c_k \|x_k - x_*\| \quad \text{for all } k \geq \hat{k}.$$

$\{x_k\}$ converges q -quadratically if there exists a constant $c \geq 0$ and an integer \hat{k} such that

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

2.2. Generalized Inverses and Projection Matrices

2.2.1. The Pseudoinverse and Other Generalized Inverse Matrices

Let A be an $n \times m$ matrix. A has an inverse if and only if $n=m$ and A is of full rank. In the general case, however, there do exist matrices which exhibit some of the properties of an inverse. Unfortunately, the nomenclature in this

area is not standardized. We will use the following notation taken from Boullion and Odell [6] and from Golub and Pereyra [23].

We will consider matrices X satisfying one or more of the following properties:

$$AXA = A \quad (1)$$

$$XAX = X \quad (2)$$

$$(XA)^T = XA \quad (3)$$

$$(AX)^T = AX \quad (4)$$

A **generalized inverse** of A is a matrix $X=A^g$ satisfying property (1).

A **reflexive generalized inverse** of A is a matrix $X=A^r$ satisfying properties (1) and (2).

A **symmetric g-inverse** of A is a matrix $X=A^s$ satisfying properties (1) and (4).

A **left weak generalized inverse** of A is a matrix $X=A^w$ satisfying properties (1), (2), and (3).

A **right weak generalized inverse** of A is a matrix $X=A^v$ satisfying properties (1), (2), and (4).

The **pseudoinverse** of A is the unique matrix $X=A^+$ satisfying properties (1), (2), (3), and (4). The pseudoinverse was introduced by Moore [41] and rediscovered by Penrose [48]. Early work in the area was done by Tseng [55] and by Bergmann, Penfield, Schiller, and Zatkis [5].

Although pseudoinverses are commonly used in the theoretical derivation of algorithms, it is often cheaper computationally to implement algorithms numerically using generalized inverses satisfying only as many of the conditions (1) - (4) as are essential to the problem being considered.

2.2.2. Projection Matrices

Let S be a subspace of \mathbb{R}^n . The matrix $P \in \mathbb{R}^{n \times n}$ is the **orthogonal projector** onto S if

$$(i) \quad P \text{ is symmetric } (P^T = P)$$

$$(ii) \quad P \text{ is idempotent } (P^2 = P)$$

$$(iii) \quad \text{The column space of } P \text{ is } S.$$

We will also refer to P as a **projection matrix**.

We will be concerned with some special projection matrices associated with a given $m \times n$ matrix A :

$$P_A \equiv AA^+ \quad \text{is the orthogonal projector onto the column space of } A.$$

$$Q_A \equiv A^+A \quad \text{is the orthogonal projector onto the row space of } A.$$

The projectors onto the orthogonal complements of the column and row spaces of A will be denoted P_A^\perp and Q_A^\perp , respectively.

2.2.3. Useful Properties of Pseudoinverses and Projection Matrices

Let A be any $m \times n$ real matrix. Collected below are some of the properties of A^+ , P_A , and Q_A which are used in the derivation of certain results appearing in later chapters.

- (i) $P_A A = A$ and $P_A^\perp A = 0$
- (ii) $A Q_A = A$ and $A Q_A^\perp = 0$
- (iii) $A^+ P_A = A^+$ and $A^+ P_A^\perp = 0$
- (iv) $Q_A A^+ = A^+$ and $Q_A^\perp A^+ = 0$
- (v) $(A^+)^+ = A$
- (vi) $(A^T)^+ = (A^+)^T$
- (vii) $(A A^T)^+ = (A^+)^T A^+$
- (viii) $(A^T A)^+ = A^+ (A^+)^T$
- (ix) $A^+ = (A^T A)^+ A^T$ and $A^+ = A^T (A A^T)^+$.

In fact,

$$A^+ = (A^T A)^w A^T \text{ for any choice of } (A^T A)^w$$

$$A^+ = A^T (A A^T)^v \text{ for any choice of } (A A^T)^v$$

- (x) $Q_A = A^w A$ for any choice of A^w
- (xi) $P_A = A A^-$ for any choice of A^-

- (xii) Let B be an $n \times k$ real matrix. It is not true in general that

$$(AB)^+ = B^+ A^+$$

2.3. The Fréchet Derivative of a Matrix

2.3.1. Definition

Let $A(\alpha)$ be an $m \times n$ matrix, each of whose elements is a function of $\alpha \in \mathbb{R}^k$. Assume that each of these elements $a_{ij}(\alpha)$ is differentiable at some point $\hat{\alpha}$. The Fréchet derivative of $A(\alpha)$ at $\hat{\alpha}$ is given by

$$D_1(\hat{\alpha}) \equiv \left\{ \frac{\partial a_{ij}(\hat{\alpha})}{\partial \alpha_s} \right\} \text{ for } i = 1, \dots, m \text{ and } s = 1, \dots, k$$

$D_1(\hat{\alpha})$ is a tridimensional tensor. It consists of k "slabs", each of which is an $m \times n$ matrix containing the partial derivatives of the a_{ij} with respect to one of the components of α .

$A(\alpha)$ is of local constant rank at a point $\hat{\alpha}$ if there exists a neighborhood $N(\hat{\alpha})$ such that $A(\alpha)$ has constant rank for all $\alpha \in N(\hat{\alpha})$. We will assume that $A(\alpha)$ is of local constant rank at every point α at which we need to compute its derivative.

2.3.2. Properties of the Fréchet Derivative of a Matrix

- (i) If A is constant, then $D_1 = 0$
- (ii) $D(A^T) = (D_1)^T$

(iii) If $C(\alpha) = A(\alpha)B(\alpha)$, then

$$DC(\alpha) = D1(\alpha)B(\alpha) + A(\alpha)DB(\alpha)$$

(iv) If $A(\alpha)$ is nonsingular and differentiable, then

$$D[A^{-1}(\alpha)] = -A^{-1}(\alpha)D1(\alpha)A^{-1}(\alpha)$$

2.3.3. The Fréchet Derivative of Projectors and Pseudoinverses

Early work in this area was done by Kruskal [31], Hearon and Evans [26], Wedin [57], Pavel-Parvu and Korganoff [47], Fletcher and Lill [19], Pérez and Scolnik [49], Guttman, Pereyra, and Scolnik [24], and Decell [9]. The following results are taken from Golub and Pereyra [22].

Lemma 2.1: Let $P_A = AA^+$ and $Q_A = A^+A$. Then

$$\begin{aligned} DP_A &= P_A^\perp (D1)A^+ + [P_A^\perp (D1)A^+]^T \\ DQ_A &= A^+ + (D1)Q_A^\perp + [A^+ (D1)Q_A^\perp]^T. \end{aligned}$$

Lemma 2.2:

$$D1^+ = -A^+(D1)A^+ + A^+(A^+)^T (D1^T)P_A^\perp + Q_A^\perp (D1^T)(A^+)^T A^+$$

2.4. Linear Least Squares

(see Lawson and Hanson [34])

Let A be an $m \times n$ matrix and assume that $m > n$.

Let b be an m -vector.

The linear least squares problem has the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|Ax - b\|^2 \quad (\text{LLS})$$

The solutions of problem (LLS) are exactly those vectors x which satisfy the normal equations

$$A^T Ax - A^T b = 0.$$

Hence any vector of the form $x = A^-b$, where A^- is a symmetric g-inverse of A , solves problem (LLS).

The choice

$$\hat{x} = A^+b$$

selects the unique solution of minimal norm. That is, \hat{x} is the unique solution of

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \|x\| \\ &\text{subject to} \quad A^T Ax - A^T b = 0 \end{aligned}$$

2.5. Nonlinear Least Squares

(General references: Dennis [10], [11])

2.5.1. Definition of the Problem

The general nonlinear least squares problem has the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) = \frac{1}{2} \|F(x)\|^2 \quad (\text{NLS})$$

where $F(x) \in \mathbb{R}^m$ and $m > n$.

F is called the residual function of problem (NLS). Let J denote its Jacobian. Then the gradient and Hessian of f at any point x are given by

$$\nabla f(x) = J(x)^T F(x)$$

$$\nabla^2 f(x) = J(x)^T J(x) + S(x)$$

$$\text{where } S(x) = \sum_{i=1}^m F_i(x) \nabla^2 F_i(x).$$

2.5.2. The Gauss-Newton Method

The Gauss-Newton method for the solution of problem (NLS) is an iterative process which, at iteration k and current point x_k , computes a step s_k satisfying

$$J(x_k)^T J(x_k) s_k = -J(x_k)^T F(x_k) \quad (2.1)$$

Hence, Gauss-Newton may be thought of as a Newton-type method with the Hessian approximation $J^T J$.

The basic Gauss-Newton method requires that $J^T J$ be nonsingular at each x_k . If we view the procedure as making a linear least squares model

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \| J s + F \|^2$$

then it is natural to generalize the step definition to

$$s_k = -J(x_k)^+ F(x_k)$$

(see Ben-Israel [4]).

2.5.3. Local Convergence of the Gauss-Newton Method

2.5.3.1. Gauss-Newton Using the True Jacobian

The following theorem gives conditions for local convergence of the Gauss-Newton method. (See Dennis and Schnabel [13]).

THEOREM 2.3: For $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$, define

$$f(x) = \frac{1}{2} F(x)^T F(x)$$

and assume that f is twice continuously differentiable in an open convex set $D \subset \mathbb{R}^n$. Suppose that the Jacobian of F satisfies

- (i) $J(x) \in Lip_\gamma(D)$
- (ii) There exists a point $x_* \in D$ such that $J(x_*)^T F(x_*) = 0$
- (iii) $\|J(x)\|_2 \leq \beta$ for all $x \in D$
- (iv) The smallest eigenvalue λ of $J(x_*)^T J(x_*)$ is nonnegative.
- (v) There exists a nonnegative scalar σ such that

$$\| [J(x) - J(x_*)]^T F(x_*) \|_2 \leq \sigma \|x - x_*\|_2 \quad \text{for all } x \in D.$$

If $\sigma < \lambda$, then, given any $c \in (1, \frac{\lambda}{\sigma})$, there is a corresponding $\epsilon > 0$ such that every possible starting value x_0 in an ϵ -neighborhood $N(x_*, \epsilon)$ of x_* , the sequence generated by the Gauss-Newton method

$$x_{k+1} = x_k - [J(x_k)^T J(x_k)]^{-1} J(x_k)^T F(x_k)$$

is well-defined, converges to x_* , and obeys

$$\|x_{k+1} - x_*\| \leq \frac{c\sigma}{\lambda} \|x_k - x_*\| + \frac{c\beta\gamma}{2\lambda} \|x_k - x_*\|^2$$

$$\|x_{k+1} - x_*\| \leq \frac{c\sigma + \lambda}{2\lambda} \|x_k - x_*\| < \|x_k - x_*\|$$

Pf: See Dennis and Schnabel [13].

Thus, if $S(x_*) = 0$, then the Gauss-Newton method is locally q -quadratically convergent.

If $S(x_*)$ is small relative to $J(x_*)^T J(x_*)$, then the Gauss-Newton method is locally q -linearly convergent.

If $S(x_*)$ is too large, we are not guaranteed even local convergence.

2.5.3.2. Gauss-Newton Using an Approximate Jacobian

We would like to consider the consequences when the Gauss-Newton method is applied using a particular type of approximation to the Jacobian.

Theorem 2.4: For $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$, define $f(x) = \frac{1}{2} F(x)^T F(x)$ and assume that

f is twice continuously differentiable in an open convex set $D \subset \mathbb{R}^n$ satisfying

$$(a) \quad J(x) = U(x) + V(x)$$

$$(b) \quad V(x)^T U(x) = 0$$

$$(c) \quad V(x)^T F(x) = 0.$$

Suppose that

$$(i) \quad J(x) \in Lip_\gamma(D)$$

$$(ii) \quad \text{There exists a point } x_* \in D \text{ such that } J(x_*)^T F(x_*) = 0$$

$$(iii) \quad \|U(x)\| \leq \beta \text{ for all } x \in D$$

$$(iv) \quad \text{The smallest eigenvalue } \lambda \text{ of } U(x_*)^T U(x_*) \text{ is nonnegative}$$

$$(v) \quad \text{There exists a nonnegative scalar } \sigma \text{ such that}$$

$$\|[U(x) - U(x_*)]^T F(x_*)\| \leq \sigma \|x - x_*\| \quad \text{for all } x \in D.$$

If $\sigma < \lambda$, then, given any $c \in (1, \frac{\lambda}{\sigma})$ there is a corresponding $\epsilon > 0$ such that for every possible starting value x_0 in an ϵ -neighborhood $N(x_*, \epsilon)$ of x_* , the sequence generated by the modified Gauss-Newton method

$$x_{k+1} = x_k - [U(x_k)^T U(x_k)]^{-1} U(x_k)^T F(x_k)$$

is well-defined, converges to x_* , and obeys

$$\|x_{k+1} - x_*\| \leq \frac{c\lambda}{\lambda} \|x_k - x_*\| + \frac{c\beta\gamma}{2\lambda} \|x_k - x_*\|$$

$$\|x_{k+1} - x_*\| \leq \frac{c\sigma + \lambda}{2\lambda} \|x_k - x_*\| < \|x_k - x_*\|.$$

Pf: Notation: Let $J_k = J(x_k)$, etc.

Assume that $0 < \sigma \leq \lambda$, and let $c \in (1, \frac{\lambda}{\sigma})$ be given.

Then $U(x_*)^T U(x_*)$ is positive definite and hence nonsingular.

So $U(x)^T U(x)$ is nonsingular for all x in some neighborhood of x_* . In particular, there is some $\eta > 0$ such that $U_0^T U_0$ is nonsingular and

$$\| (U_0^T U_0)^{-1} \| \leq \frac{c}{\lambda} \quad \text{for any } x_0 \in N(x_*, \eta)$$

Let

$$\epsilon = \min\left(\eta, \frac{\lambda - c\sigma}{c\beta\gamma}\right)$$

$$x_1 = x_0 - (U_0^T U_0)^{-1} U_0^T F_0$$

which is well-defined.

$$\begin{aligned} x_1 - x_* &= (x_0 - x_*) - (U_0^T U_0)^{-1} U_0^T F_0 \\ &= -(U_0^T U_0)^{-1} [U_0^T F_0 + (U_0^T U_0)(x_* - x_0)] \\ &= -(U_0^T U_0)^{-1} [U_0^T F_0 + U_0^T F_* - U_0^T F_* \\ &\quad + U_0^T V_0(x_* - x_0) + U_0^T U_0(x_* - x_0)] \\ &= -(U_0^T U_0)^{-1} [U_0^T F_* - U_0^T (F_* - F_0 - J_0(x_* - x_0))] \end{aligned}$$

$$\begin{aligned} \|x_1 - x_*\| &\leq \| (U_0^T U_0)^{-1} \| \left[\|U_0^T F_* - U_*^T F_*\| + \|U_0\| \|F_* - F_0 - J_0(x_* - x_0)\| \right] \\ &\leq \frac{c}{\lambda} [\sigma \|x_0 - x_*\| + \beta \frac{\gamma}{2} \|x_0 - x_*\|^2] \end{aligned}$$

Since

$$x_0 \in N(x_*, \epsilon) \text{ implies } \|x_0 - x_*\| \leq \frac{\lambda - c\sigma}{c\beta\gamma},$$

we have

$$\begin{aligned} \|x_1 - x_*\| &\leq \frac{c\sigma}{\lambda} \|x_0 - x_*\| + \left(\frac{c\beta\gamma}{2\lambda} \right) \left(\frac{\lambda - c\sigma}{c\beta\gamma} \right) \|x_0 - x_*\| \\ &= \left(\frac{c\sigma + \lambda}{2\lambda} \right) \epsilon 0. \end{aligned}$$

And

$$c < \frac{\lambda}{\sigma} \text{ implies } \frac{c\sigma}{2\lambda} < \frac{\lambda^2\sigma}{2\sigma\lambda} = \frac{\lambda}{2},$$

giving

$$\|x_1 - x_*\| < \epsilon 0.$$

And the remainder of the proof follows by induction

2.5.4. Modifications of the Gauss-Newton Method

Although the Gauss-Newton method can work very well, there are difficulties when the starting point is far from the solution, the problem has a large value of the residual function at the solution, or $J^T J$ is singular (or poorly conditioned). One type of modification which has been used is exemplified by the method of Hartley [25], which uses a line search and computes a damped Gauss-Newton step based on some appropriate criteria.

An alternate approach is to use as the approximate Hessian $J^T J + B$. The Levenberg-Marquardt method ([37], [40], [42]) uses $B = \lambda I$, where λ is a suitable parameter. This method, while successful in many cases, has local convergence properties similar to those of Gauss-Newton. Other choices for B

can be used to improve the convergence properties. For example, B can be chosen to be an appropriate type of approximation to S . In NL2SOL (see Dennis, Gay, and Welsch [12]), B is a secant approximation.

2.6. First-Order Sensitivity Analysis of a Second-Order Local Solution

Sensitivity theory in nonlinear programming is concerned with an analysis of the behavior of a given nonlinear programming problem under perturbation of the parameters appearing in the problem. The development below follows that of Fiacco [17]. (See also Bank, Guddat, Klatte, Kummer, and Tammer [2] and Fiacco [15], [16]).

Consider the problem of determining a local solution $x(\epsilon)$ of the following problem.

$$\begin{aligned}
 [P(\epsilon)] \quad & \underset{x}{\text{minimize}} \quad f(x, \epsilon) \\
 & \text{subject to} \quad g_i(x, \epsilon) \geq 0 \quad i=1, \dots, m \\
 & \quad \quad \quad h_j(x, \epsilon) = 0 \quad j=1, \dots, p
 \end{aligned}$$

where $x \in \mathbb{R}^n$ and ϵ is a parameter vector in \mathbb{R}^t .

Let

$$l(x, u, w, \epsilon) \equiv f(x, \epsilon) - \sum_{i=1}^m u_i g_i(x, \epsilon) + \sum_{j=1}^p w_j h_j(x, \epsilon)$$

denote the Lagrangian of problem $P(\epsilon)$.

We are interested in analyzing the behavior of a local solution $x(\bar{\epsilon})$ of Problem $P(\epsilon)$ when $\bar{\epsilon}$ is subject to perturbation. For simplicity, and without loss of generality, let $\bar{\epsilon}=0$. Also assume that each of the functions is twice continuously differentiable.

2.6.1. Second-Order Sufficient Conditions for a Minimizer of Problem $P(\epsilon)$ (See Fiacco and McCormick [18], McCormick [30], and Fiacco [17])

A point x^* satisfies the first-order conditions for problem $P(\epsilon)$ if there exist vectors $u^* \in \mathbb{R}^m$ and $w^* \in \mathbb{R}^p$ such that

- (i) $\nabla_x l(x^*, u^*, w^*, \epsilon) = 0$
- (ii) $g_i(x^*, \epsilon) \geq 0 \quad i=1, \dots, m$
- (iii) $h_j(x^*, \epsilon) = 0 \quad j=1, \dots, p$
- (iv) $u_i^* g_i(x^*) = 0 \quad i=1, \dots, m$
- (v) $u_i^* \geq 0 \quad i=1, \dots, m$

Condition (iv) is the complementarity condition. If we require in addition that one of u_i^* and $g_i(x^*)$ must be strictly positive, then strict complementarity will be said to hold.

The first-order conditions are necessary conditions provided the constraints satisfy some sort of constraint qualification which describes certain characteristics that the constraint set must have. For the purposes of sensitivity analysis, we will make a stronger linear independence assumption which will imply that appropriate constraint qualifications hold.

A point \hat{x} which satisfies the constraints of problem $P(\epsilon)$ is a **regular point** if the gradients of the equality constraints and the binding inequality constraints form a linearly independent set when evaluated at \hat{x} .

The following lemma gives second-order sufficient conditions for x^* to be a strict local minimizer of problem $P(0)$.

Let

$$B(x^*, 0) \equiv \{i \mid g_i(x^*, 0) = 0\}$$

$$D(x^*, 0) \equiv \{i \in B(x^*, 0) \mid u_i^* > 0\}$$

Lemma 2.5: Suppose that the functions defining problem $P(0)$ are twice continuously differentiable in a neighborhood of x^* . Then x^* is a strict local minimizer of problem $P(0)$ if there exist Lagrange multiplier vectors $u^* \in \mathbb{R}^m$ and $w^* \in \mathbb{R}^p$ such that the first-order conditions hold at x^* and, further, if for every nonzero $z \in \mathbb{R}^n$ such that

$$z^T \nabla g_i(x^*, 0) = 0 \quad i \in D(x^*, 0)$$

$$z^T \nabla g_i(x^*, 0) \geq 0 \quad i \in B(x^*, 0) - D(x^*, 0)$$

$$z^T \nabla h_j(x^*, 0) = 0 \quad j=1, \dots, p$$

we have that

$$z^T \left[\nabla^2 l(x^*, u^*, w^*, 0) \right] z > 0.$$

These conditions are valid even when a vector z of the appropriate type does not exist, and they may also be applied in the unconstrained case.

2.6.2. Sensitivity Analysis

Conditions for the existence of a function $x(\epsilon)$ may be obtained by extending the implicit function theorem to the case of mathematical programming. We are interested in the case in which this function not only exists but is locally unique and differentiable. Sufficient conditions for this are given in the following theorem.

Theorem 2.6: (Basic Sensitivity Theorem; Fiacco [14])

If

- (i) The functions defining problem $P(\epsilon)$ are twice continuously differentiable in x , and if their gradients with respect to x and the constraints are once continuously differentiable in ϵ in a neighborhood of $(x^*, 0)$.
- (ii) The second-order sufficiency conditions for a local minimizer of $P(0)$ hold at x^* , with associated Lagrange multipliers u^* and w^* .

(iii) The gradients

$$\nabla g_i(x^*, 0), \quad \text{for all } i \text{ such that } g_i(x^*, 0) = 0,$$

and

$$\nabla h_j(x^*, 0), \quad \text{for all } j,$$

are linearly independent.

(iv) Strict complementarity holds. (ie, multipliers corresponding to binding inequality constraints are strictly positive),

then

(a) x^* is a local isolated minimizer of $P(0)$ and the associated Lagrange multipliers u^* and w^* are unique.

(b) For ϵ in a neighborhood of 0, there exists a unique once continuously differentiable vector function

$$y(\epsilon) = [x(\epsilon), u(\epsilon), w(\epsilon)]^T$$

satisfying the second-order sufficient conditions for a local minimizer of $P(\epsilon)$ and such that

$$y(0) = (x^*, u^*, w^*) \equiv y^*;$$

hence $x(\epsilon)$ is a locally unique local minimizer of $P(\epsilon)$ with associated unique Lagrange multipliers $u(\epsilon)$ and $w(\epsilon)$

(c) For ϵ near 0, the set of binding inequalities is unchanged, strict complementarity holds, and the binding constraint gradients are linearly independent at $x(\epsilon)$.

CHAPTER 3

The Semilinear Least Squares Problem

3.1. The Unconstrained Problem

3.1.1. Problem Definition

The (unconstrained) semilinear least squares problem has the form

$$\underset{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^l}{\text{minimize}} \quad f(x, \alpha) = \frac{1}{2} \|F(x, \alpha)\|^2 \quad (\text{SLS})$$

$$\text{where } F(x, \alpha) = A(\alpha)x - b; \quad A(\alpha) \in \mathbb{R}^{m \times n}$$

It is a nonlinear least squares problem in which some of the variables (x_1, x_2, \dots, x_n) appear linearly in the residual F . A common problem of this type is curve fitting in which the model function is expressed as the sum of exponentials (or rational functions or expressions describing probability distributions), each term having a coefficient.

The semilinear least squares problem is more commonly known as a separable nonlinear least squares problem or as a nonlinear least squares problem whose variables separate. However, this terminology is somewhat misleading. First, there is a more common usage of the word separable to indicate, for example, problems in which the objective function can

be expressed as $f(x, \alpha) = f_1(x) + f_2(\alpha)$ or as $f(x, \alpha) = f_1(x) f_2(\alpha)$. Second, although the word "separate" connotes a lack of connection between the variable groups, the most important feature of these problems is that there is a strong and distinct relationship between x and α .

3.1.2. Ideas for Solution Using the Gauss-Newton Technique

The Gauss-Newton method for the solution of nonlinear least squares problems will provide a framework for the comparison of several procedures for the solution of problem SLS. Recall that the Gauss-Newton step is computed from the equation

$$(J^T J) s = -J^T F \quad (3.1)$$

so that

$$s = -J^+ F.$$

The Jacobian of the residual function F of problem SLS is

$$J = [J_x \quad J_\alpha]$$

$$\text{with } J_x = A(\alpha)$$

$$J_\alpha = Df(\alpha) x - b$$

where D indicates the Fréchet derivative.

The step s has components corresponding to each of the $n+k$ variables in the problem. We will derive separate expressions for the step s_x in the x -variables and the step s_α in the α -variables.

Expanding equation (1) above, we obtain the pair of equations

$$A^T A s_x + A^T J_\alpha s_\alpha = -A^T F \quad (3.2)$$

$$J_\alpha^T A s_x + J_\alpha^T J_\alpha s_\alpha = -J_\alpha^T F. \quad (3.3)$$

Multiply equation (3.2) on the left by $J_\alpha^T (A^+)^T$ and subtract the result from equation (3.3). This yields

$$J_\alpha^T (I - AA^+) J_\alpha s_\alpha = -J_\alpha^T (I - AA^+) F.$$

Defining

$$P_A^\perp \equiv I - AA^+,$$

we obtain

$$J_\alpha^T P_A^\perp J_\alpha s_\alpha = -J_\alpha^T P_A^\perp F.$$

Since the projection matrix P_A^\perp is symmetric and idempotent, this equation can also be written

$$(P_A^\perp J_\alpha)^T (P_A^\perp J_\alpha) s_\alpha = -(P_A^\perp J_\alpha)^T P_A^\perp F.$$

So take

$$s_\alpha = -(P_A^\perp J_\alpha)^T P_A^\perp F.$$

From equation (3.2), s_x may be computed from s_α as

$$s_x = -A^+ (F - J_\alpha s_\alpha).$$

The Gauss-Newton process, expressed in terms of the steps we have derived, can be written

ALGORITHM 0: (Gauss-Newton on the full problem)

- (1) Choose x^1 and α^1
- (2) For $j=1, 2, \dots$ until convergence do
 - (2.1) Test for convergence
 - (2.2) $s_\alpha = -(P_A^\perp J_\alpha)^+ P_A^\perp F$
 $s_x = -A^+ F - A^+ J_\alpha s_\alpha$
 (where these quantities are evaluated at x^j, α^j)

$$(2.3) \quad \begin{aligned} \alpha^{j+1} &= \alpha^j + s_\alpha^j \\ x^{j+1} &= x^j + s_x^j. \end{aligned}$$

Problem SLS, however, has more structure than a general nonlinear least squares problem. If the optimal value of α were known, then problem SLS would reduce to a linear least squares problem in x . We would like to take advantage of this simplification.

A straightforward application of this idea is an alternation scheme in which one group of variables is held constant while work is done on the other set. This type of method has been applied in the solution of statistical problems. An example is NIPALS, developed by Wold and Lyttkens [50].

ALGORITHM 1: (Alternation)

- (1) Choose α^j
- (2) For $j=1, 2, \dots$ until convergence do
 - (2.1) Solve the linear least squares problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|A(\alpha^j)x - b\|^2 \quad (3.4)$$

to obtain x^j

- (2.2) Test for convergence

$$(2.3) \quad s_\alpha = -J_\alpha^+ F$$

$$(2.4) \quad \alpha^{j+1} = \alpha^j + s_\alpha.$$

The full Gauss-Newton method (Algorithm 0) treats the x and α variables alike. The alternation method of Algorithm 1 acknowledges the simplicity of the problem when expressed as a function of x alone by solving an entire minimization problem in x following each new step in α . The computation of the step in α is done by treating x as a constant. Yet the problem has no special structure in this case.

We have already derived an expression for s_α in the full Gauss-Newton case. It may be computed independently of s_x and does not require any extra assumptions on the form of the function. Since the problem has special structure as a function of x , we can easily find a point \hat{x} which is better than $x^j + s_x$ in the sense that $f(\hat{x}, \alpha^j + s_\alpha) \leq f(x^j + s_x, \alpha^j + s_\alpha)$. In fact, if we choose \hat{x} to be the solution of the linear least squares problem (3.4), then we obtain the lowest function value it is possible to achieve using $\alpha^j + s_\alpha$. So we might try computing a step in α and then choosing as our next x an optimal partner of the new α . This is the method proposed by Barham and Drane [3]. (See also Walling [56]).

ALGORITHM 2: (Gauss-Newton in α ; corrected x)

- (1) Choose α^j
- (2) For $j=1, 2, \dots$ until convergence do
 - (2.1) Solve the linear least squares problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|A(\alpha^j)x - b\|^2$$

to obtain x^j

- (2.2) Test for convergence
- (2.3) $s_\alpha = -(P_A^\perp J_\alpha)^+ P_A^\perp F$
- (2.4) $\alpha^{j+1} = \alpha^j + s_\alpha.$

Consider the linear least squares problem which appears in Algorithms 1 and 2. In general, there may be many solutions. If we choose the solution of minimal norm, we can give an explicit representation for x^j in terms of α^j :

$$x^j = A(\alpha^j)^+ b.$$

Under suitable conditions, we can define a differentiable function

$$x(\alpha) = A(\alpha)^+ b$$

and use it to eliminate x from the objective function.

Define the variable projection functional $f: \mathbb{R}^k \rightarrow \mathbb{R}$ by

$$\begin{aligned} f(\alpha) &\equiv f(x(\alpha), \alpha) \\ &= \frac{1}{2} \| A(\alpha)A(\alpha)^+ b - b \|^2. \end{aligned}$$

The corresponding residual function will be denoted by

$$F(\alpha) = -P_A^\perp(\alpha) b.$$

We may then define a reduced problem

$$\underset{\alpha \in \mathbb{R}^k}{\text{minimize}} \quad f = \frac{1}{2} \| -P_A^\perp(\alpha) b \|^2.$$

After the solution $\hat{\alpha}$ is computed, \hat{x} is set to $A(\hat{\alpha})^+ b$.

The variable projection functional f involves fewer variables than the full functional J but at the cost of greater complexity. In particular, determination of the Jacobian of F requires differentiation of the matrix projection function P_A . This difficulty slowed the development of variable projection methods. Lawton and Sylvestre [38], for example, stated that it is "impractical to obtain the analytical derivative" of F and used finite differences instead. Scolnik [54], Guttman, Pereyra, and Scolnik [24], and Pérez and Scolnik [40] developed expressions for the derivative of P_A , beginning with simple cases and working toward the general problem. Golub and Pereyra [21], [22], [23], showed how to

differentiate pseudoinverses and projection matrices in the general non-full-rank case and derived expressions for the Jacobian of F and the gradient of f .

Whenever the form of a problem is modified, as with a change of variable, there is the possibility of adding or deleting solution points. The following theorem of Golub and Pereyra [22] (and in [21] for the more general case in which $g(x)$ replaces x) asserts that the change from minimization of the full functional to minimization of the variable projection functional does not add any critical points and does not delete the solution of the original problem.

Theorem 3.1: Let $f(x, \alpha)$ and f be defined as above. Assume that $A(\alpha)$ has constant rank $r \leq \min(m, n)$ for all α in the open set $\Omega \subseteq \mathbb{R}^k$.

- (a) If $\hat{\alpha}$ is a critical point (or a global minimizer in Ω) of $f(\alpha)$ and if $\hat{x} = A(\hat{\alpha})^+ b$, then $(\hat{x}, \hat{\alpha})$ is a critical point of $f(x, \alpha)$ (or a global minimizer for $\alpha \in \Omega$) and $f(\hat{x}, \hat{\alpha}) = f(\hat{\alpha})$.
- (b) If $(\hat{x}, \hat{\alpha})$ is a global minimizer of $f(x, \alpha)$ for $\alpha \in \Omega$, then $\hat{\alpha}$ is a global minimizer of $f(\alpha)$ in Ω and $f(\hat{\alpha}) = f(\hat{x}, \hat{\alpha})$. Furthermore, if there is a unique \hat{x} among the minimizing pairs of $f(x, \alpha)$, then $\hat{x} = A(\hat{\alpha})^+ b$.

This theorem provides justification for solving problem SLS by minimizing the variable projection functional. The Jacobian of F can be computed by applying the rules for differentiation of projection matrices.

$$\begin{aligned}
J(\alpha) &= P_A^\perp (D_1) A^+ b + (A^+)^T (D_1^T) P_A^\perp b \\
&= P_A^\perp J_\alpha - P_A (A^+)^T (D_1^T) F.
\end{aligned} \tag{3.5}$$

The Gauss-Newton method applied to the variable projection functional f is

ALGORITHM 3: (Gauss-Newton on the reduced problem)

- (1) Choose α^1
- (2) For $j=1, 2, \dots$ until convergence do
 - (2.1) Test for convergence. On convergence, go to (3) with $\hat{\alpha} = \alpha^j$
 - (2.2) $s_\alpha = -[P_A^\perp J_\alpha - P_A (A^+)^T (D_1^T) F]^+ P_A^\perp F$
 - (2.3) $\alpha^{j+1} = \alpha^j + s_\alpha$
- (3) $\hat{x} = A(\hat{\alpha})^+ b$

Unfortunately, computation of $J(\alpha)$ is expensive. Golub and Pereyra perform a trapezoidal orthogonal factorization of the matrix $A(\alpha)$ and use a symmetric g-inverse A^- instead of the pseudoinverse A^+ (recall that P_A will be the same in either case.) Krogh [30] assumes a full column rank A and performs a QR decomposition. He also notes that the method extends immediately to the case in which b is also a function of α .

Both of these methods must form the tensor D_1 since it appears in the expression for $J(\alpha)$ as both D_1 and D_1^T . Kaufman [27] drops the second term on the right-hand side of equation (3.5). Using this approximation to the full Jacobian speeds the computation and introduces more flexibility into the handling of the tensor D_1 . For example, we can evaluate the $m \times k$ matrix J_α directly rather than performing the multiplications.

3.1.3. Solution of the Reduced Problem

Algorithms 1, 2, and 3 have been presented as methods for applying the Gauss-Newton procedure to the original problem, SLS. The same algorithms may also be thought of as various applications of Gauss-Newton to minimization of the variable projection functional.

We are interested primarily in the three forms of the step calculation.

$$\begin{aligned}
(s_\alpha)_1 &= -J_\alpha^+ F \\
(s_\alpha)_2 &= -(P_A^\perp J_\alpha)^+ P_A^\perp F \\
(s_\alpha)_3 &= -[P_A^\perp J_\alpha - P_A (A^+)^T (D_1^T) F]^+ P_A^\perp F.
\end{aligned}$$

Recall that the quantities on the right-hand side are, in each case, evaluated at a point (x, α) satisfying

$$A(\alpha)^T A(\alpha) x - A(\alpha)^T b = 0.$$

The general form for a vector x satisfying this equation is

$$x = A(\alpha)^+ b + y,$$

where y is in the null space of $A(\alpha)$. Then

$$\begin{aligned} F(x, \alpha) &= A(\alpha) x - b \\ &= A(\alpha) A(\alpha)^+ b + A(\alpha) y - b \\ &= -P_A^\perp(\alpha) b \\ &= F(\alpha). \end{aligned}$$

Clearly, $P_A^\perp F = F$ and $P_A^\perp F = F$. So we may rewrite the step calculations as

$$\begin{aligned} (s_\alpha)_1 &= -J_\alpha^+ F \\ (s_\alpha)_2 &= -(P_A^\perp J_\alpha)^+ F \\ (s_\alpha)_3 &= -[P_A^\perp J_\alpha - P_A (A^+)^T (D_H^T) F]^+ F. \end{aligned}$$

The step $(s_\alpha)_3$ is the actual Gauss-Newton step for the variable projection functional. The other steps are approximations to it which arise from approximating J by some of its terms. Note that $(s_\alpha)_2$, which was derived from the Barham and Drane approach, is exactly the step used by Kaufman.

Ruhe and Wedin [52] have computed the asymptotic r -convergence rates of all four of these algorithms for solving problem SLS. They determined that each algorithm is r -linear in general and that Algorithms 2 and 3 display

superlinear convergence whenever Algorithm 0 (Gauss-Newton on the full problem) does. Algorithm 1, however, always has linear convergence.

Algorithm 3 is just Gauss-Newton on the reduced problem. So, by theorem 2.3, it is locally q -linearly convergent for problems satisfying the conditions of the hypothesis of that theorem. Algorithm 2 involves a Jacobian approximation of the form required by the hypothesis of theorem 2.4. Hence, Algorithm 2 displays local q -linear convergence when applied to suitable problems. Algorithms 0, 2, and 3 are all locally q -quadratically convergent for problems in which the residual at the solution is zero.

3.1.4. Solution of the Semilinear Least Squares Problem in Practice

The Gauss-Newton method has provided a convenient framework for the discussion of the semilinear least squares problem. While it is valuable from a theoretical standpoint, other methods are used in practice. By theorem 3.1, we can find critical points for the full problem SLS by working with the reduced problem instead; and Golub and Pereyra have provided expressions for the gradient of the variable projection functional and for the Jacobian of the associated residual function. The minimization may then be done using any appropriate method.

In [22], Golub and Pereyra compare the solution of the full functional and variable projection formulations of the semilinear least squares problem. The minimization was accomplished by a variety of methods requiring different kinds

of derivative information (function only, gradient of f or F , and Jacobian of F or F). In general, the variable projection formulation was found to require fewer function evaluations than the full functional version. However, the greater complexity introduced by the variable transformation means that each iteration in the reduced problem formulation requires more work, and there were cases in which solution of the original problem took less time. The Jacobian approximation of Kaufman [27] and the revised computations recommended by Krogh [30] and Kaufman [27] have reduced the work per iteration in later tests.

3.2. The Constrained Problem

3.2.1. Linear Inequality Constraints on α

The variable projection method may be extended to constrained problems as well. Kaufman has shown (see [23]), that a problem of the form

$$\begin{aligned} \underset{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^k}{\text{minimize}} \quad & f(x, \alpha) = \frac{1}{2} \| A(\alpha) x - b \|^2 \\ \text{subject to} \quad & H^T \alpha \geq d \end{aligned}$$

is equivalent to

$$\underset{\alpha \in \mathbb{R}^k}{\text{minimize}} \quad f(\alpha) = \frac{1}{2} \| -P_A^\perp(\alpha) b \|^2$$

$$\text{subject to } H^T \alpha \geq d$$

followed by

$$\hat{x} = A(\alpha)^+ b.$$

3.2.2. Semilinear Equality Constraints

Constraints of the form

$$H(\alpha) x = g(\alpha)$$

will be called semilinear equality constraints. Kaufman and Pereyra [28] have shown that elimination of constraints of this type from a minimization problem with a semilinear least squares objective function yields a (more complicated) unconstrained semilinear least squares problem.

If the constraints are consistent (ie, if $g(\alpha)$ lies in the column space of $H(\alpha)$), then the general form for any vector x which satisfies them is

$$x = H^+(\alpha) g(\alpha) + Y(\alpha) z$$

where $Y(\alpha)$ is a basis for the null space of $H(\alpha)$ and z is some vector in $\mathbb{R}^{n - \text{rank}(A)}$.

Substituting this expression into the objective function, we obtain

$$s(z, \alpha) = \frac{1}{2} \| (A(\alpha) Y(\alpha)) z - (b - A(\alpha) H(\alpha)^+ g(\alpha)) \|^2$$

which is a semilinear least squares problem in the variables z and α .

The new problem can be solved by a general variable projection method or by a modification which takes into account the special structure of the transformed problem.

$$\text{Let } G = AY$$

$$\rho = b - AH^+g$$

$$\tilde{F} = -P_G^\perp \rho.$$

Then the Jacobian of the new residual function \tilde{F} is

$$P_G^\perp(Dg) G^+ \rho - P_G (G^+)^T (DG^T) \tilde{F}$$

As in the unconstrained case, we form a Jacobian approximation by dropping the second term. Then we may use the change of variables formulas to write the first term using the original variables.

$$P_{AY}^\perp [-A H^+(D1) \gamma + A H^+(Dg) + (D1) \gamma]$$

where $\gamma = Y(AY)^+(b - AH^+g) + H^+g$

Kaufman and Pereyra used a symmetric g-inverse $(AY)^-$ in place of $(AY)^+$ in their implementation of this method. Corradi [8] reduced the computation required still further by replacing H^+ by an arbitrary generalized inverse $H^{\#}$.

CHAPTER 4

Reducible Nonlinear Programming Problems

4.1. Problem Statement

In this chapter, we will be concerned with the following three problems.

$$\begin{aligned} \text{[I]} \quad & \underset{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^l}{\text{minimize}} \quad f(x, \alpha) \\ & \text{subject to} \\ & g_i(x, \alpha) \geq 0 \quad i = 1, \dots, p_1 \\ & h_i(x, \alpha) = 0 \quad i = 1, \dots, p_2 \\ & c_i(x) \geq 0 \quad i = 1, \dots, p_3 \\ & d_i(x) = 0 \quad i = 1, \dots, p_4 \\ & r_i(\alpha) \geq 0 \quad i = 1, \dots, p_5 \\ & s_i(\alpha) = 0 \quad i = 1, \dots, p_6 \end{aligned}$$

$$\begin{aligned} \text{[II]} \quad & \underset{\alpha \in \mathbb{R}^l}{\text{minimize}} \quad f(\alpha) \equiv f(x(\alpha), \alpha) \\ & \text{subject to} \end{aligned}$$

$$r_i(\alpha) \geq 0 \quad i = 1, \dots, p_5$$

$$s_i(\alpha) = 0 \quad i = 1, \dots, p_6$$

where $x(\alpha)$ solves

$$(A) \quad \begin{aligned} & \text{minimize } t(x) \equiv f(x; \alpha) \\ & \text{subject to} \\ & g_i(x, \alpha) \geq 0 \quad i = 1, \dots, p_1 \\ & h_i(x, \alpha) = 0 \quad i = 1, \dots, p_2 \\ & c_i(x) \geq 0 \quad i = 1, \dots, p_3 \\ & d_i(x) = 0 \quad i = 1, \dots, p_4 \end{aligned}$$

Problem [I] represents the original nonlinear programming problem which we need to solve; problem [II] is the corresponding reduced problem. We will give conditions under which we can find a point (x^*, α^*) satisfying the first-order conditions for problem [I] by solving problem [II] instead.

4.2. Analysis of the Subproblem

First consider the subproblem (A). The elements of the vector α are variables in the original problem, but they act as parameters in the subproblem. So we may analyze problem (A) using results from sensitivity theory. We will use this analysis to derive an expression for $\frac{dx(\alpha)}{d\alpha}$.

Assume that problem (A) satisfies the hypothesis of the basic sensitivity theorem at the point $(\hat{x}, \hat{\alpha})$. In particular, the first-order necessary conditions

hold at $(\hat{x}, \hat{\alpha})$.

Let the Lagrangian for problem (A) be

$$\begin{aligned} l(x, u, v, \lambda, \mu; \alpha) = & f(x, \alpha) - \sum_{i=1}^{p_1} u_i g_i(x, \alpha) - \sum_{i=1}^{p_2} v_i c_i(x) \\ & + \sum_{i=1}^{p_3} \lambda_i h_i(x, \alpha) + \sum_{i=1}^{p_4} \mu_i d_i(x) \end{aligned}$$

By the conclusions of the sensitivity theorem, the system of equations

$$\begin{aligned} \nabla_x l(x, \alpha) &= 0 \\ u_i g_i(x, \alpha) &= 0 & i = 1, \dots, p_1 \\ v_i c_i(x) &= 0 & i = 1, \dots, p_3 \\ h_i(x, \alpha) &= 0 & i = 1, \dots, p_2 \\ d_i(x) &= 0 & i = 1, \dots, p_4 \end{aligned}$$

holds for all α in some neighborhood of $\hat{\alpha}$. Furthermore, this system may be differentiated with respect to α , and its Jacobian matrix is nonsingular.

Differentiating the first equation, we obtain

$$\begin{aligned} 0 = & [(\nabla_x^2 f)(dx/d\alpha) + \nabla_{x\alpha}^2 f] - \sum_{i=1}^{p_1} [u_i(\nabla_x^2 g_i)(dx/d\alpha) \\ & + u_i(\nabla_{x\alpha}^2 g_i) + (\nabla_x g_i)(du_i/d\alpha)] \\ & - \sum_{i=1}^{p_2} [v_i(\nabla_x^2 c_i)(dx/d\alpha) + (\nabla_x c_i)(dv_i/d\alpha)] \end{aligned}$$

$$\begin{aligned}
& + \sum_{i=1}^{p_2} [\lambda_i (\nabla_x^2 h_i) (dx/d\alpha) + \lambda_i (\nabla_{x\alpha}^2 h_i) \\
& + (\nabla_x h_i) (d\lambda_i/d\alpha)] + \sum_{i=1}^{p_4} [\mu_i (\nabla_x^2 d_i) (dx/d\alpha) \\
& + (\nabla_x d_i) (d\mu_i/d\alpha)]
\end{aligned}$$

$$\begin{aligned}
0 &= (\nabla_x^2 l) (dx/d\alpha) + (\nabla_{x\alpha}^2 l) - \sum_{i=1}^{p_1} (\nabla_x g_i) (du_i/d\alpha) \\
&- \sum_{i=1}^{p_2} (\nabla_x c_i) (dv_i/d\alpha) + \sum_{i=1}^{p_3} (\nabla_x h_i) (d\lambda_i/d\alpha) \\
&+ \sum_{i=1}^{p_4} (\nabla_x d_i) (d\mu_i/d\alpha)
\end{aligned}$$

Differentiating the general g-constraint yields

$$u_i (\nabla_x g_i)^T (dx/d\alpha) + u_i (\nabla_{\alpha} g_i) + g_i (du_i/d\alpha) = 0.$$

The other constraint equations give similar results.

In matrix form, the full system of equations resulting from differentiation with respect to α is

$$M \left(\frac{dy}{d\alpha} \right) = N \quad (4.1)$$

$$\text{where } \frac{dy}{d\alpha} = \begin{bmatrix} dx/d\alpha \\ du/d\alpha \\ dv/d\alpha \\ d\lambda/d\alpha \\ d\mu/d\alpha \end{bmatrix}$$

$$\text{with } u = (u_1, \dots, u_{p_1})^T$$

$$v = (v_1, \dots, v_{p_2})^T$$

$$\lambda = (\lambda_1, \dots, \lambda_{p_3})^T$$

$$\mu = (\mu_1, \dots, \mu_{p_4})^T$$

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & M_{22} & 0 & 0 & 0 \\ M_{31} & 0 & M_{33} & 0 & 0 \\ M_{14}^T & 0 & 0 & 0 & 0 \\ M_{15}^T & 0 & 0 & 0 & 0 \end{bmatrix}$$

with

$$M_{11} = \nabla_x^2 l$$

$$M_{12} = [-\nabla_x g_1, \dots, -\nabla_x g_{p_1}]$$

$$M_{13} = [-\nabla_x c_1, \dots, -\nabla_x c_{p_2}]$$

$$M_{14} = [\nabla_x h_1, \dots, \nabla_x h_{p_3}]$$

$$M_{21} = -\text{diag}(v_1, \dots, v_{p_2}) M_{12}^T$$

$$M_{22} = \text{diag}(g_1, \dots, g_{p_1})$$

$$M_{31} = -\text{diag}(v_1, \dots, v_{p_2}) M_{13}^T$$

$$M_{33} = \text{diag}(c_1, \dots, c_{p_2})$$

and

$$N = \begin{bmatrix} -\nabla_{\alpha x}^2 l & -u_1 \nabla_{\alpha} g_1, \dots, u_{p_1} g_{p_1} & 0 & -\nabla_{\alpha} h_1, \dots, \nabla_{\alpha} h_{p_2} & 0 \end{bmatrix}$$

$$M \text{ is } (n + p_1 + p_2 + p_3 + p_4) \times (n + p_1 + p_2 + p_3 + p_4)$$

$$\frac{dy}{d\alpha} \text{ is } (n + p_1 + p_2 + p_3 + p_4) \times k$$

$$N \text{ is } (n + p_1 + p_2 + p_3 + p_4) \times k$$

Suppose, without loss of generality, that the inequality constraints that are binding at $\hat{\alpha}$ are:

$$g_1, \dots, g_{q_1} \text{ and } c_1, \dots, c_{p_1}.$$

When convenient, we will refer to groups of constraints or multipliers collectively. For example,

$$g \equiv \begin{bmatrix} \bar{g} \\ \hat{g} \end{bmatrix} = \begin{bmatrix} g_1 \\ \vdots \\ g_{q_1} \\ g_{q_1+1} \\ \vdots \\ g_{p_1} \end{bmatrix}$$

First consider the equations of (4.1) involving the g -constraints. Since the value of each binding constraint is zero and the multipliers associated with nonbinding constraints are zero by complementarity, these equations have the form

$$-\Psi_{\bar{g}} (\nabla_x \bar{g})^T (dx/d\alpha) = \Psi_{\bar{g}} (\nabla_{\alpha} \bar{g})^T$$

$$\tilde{G} (d\bar{u}/d\alpha) = 0$$

$$\text{where } \Psi_{\bar{g}} \equiv \text{diag}(-u_1, \dots, -u_{q_1})$$

$$\tilde{G} \equiv \text{diag}(g_{q_1+1}, \dots, g_{p_1})$$

\tilde{G} is nonsingular by definition, and $\Psi_{\bar{g}}$ is nonsingular by the assumption of strict complementarity. Hence

$$-(\nabla_x \bar{g})^T (dx/d\alpha) = (\nabla_{\alpha} \bar{g})^T$$

$$d\bar{u}/d\alpha = 0$$

By a similar argument,

$$-(\nabla_x \bar{c})^T (dx/d\alpha) = 0$$

$$d\bar{v}/d\alpha = 0.$$

The equations corresponding to the equality constraints are simply

$$\begin{aligned} (\nabla_x h)^T (dx/d\alpha) &= -(\nabla_\alpha h)^T \\ (\nabla_x d)^T (dx/d\alpha) &= 0. \end{aligned}$$

We have shown that the derivatives of the multipliers corresponding to nonbinding constraints are all zero. This was to be expected since the sensitivity theorem shows that constraints which are nonbinding at $(\hat{x}, \hat{\alpha})$ will remain nonbinding throughout the neighborhood; and their associated multipliers must, therefore, remain constant at zero.

Defining

$$\begin{aligned} \bar{\theta} &= (u_1, \dots, u_{q_1}, v_1, \dots, v_{q_2}, \lambda_1, \dots, \lambda_{p_2}, \mu_1, \dots, \mu_{p_4})^T \\ \bar{R}^T &= [-\nabla_x \bar{\theta} \mid -\nabla_x \bar{c} \mid \nabla_x h \mid \nabla_x d] \\ \bar{\Gamma}^T &= [-\nabla_\alpha \bar{\theta} \mid -\nabla_\alpha \bar{c} \mid \nabla_\alpha h \mid \nabla_\alpha d] \\ &= [-\nabla_\alpha \bar{\theta} \mid 0 \mid \nabla_\alpha h \mid 0], \end{aligned}$$

The reduced system of equations becomes

$$\begin{bmatrix} \nabla_x^2 l & \bar{R}^T \\ \bar{R} & 0 \end{bmatrix} \begin{bmatrix} dx/d\alpha \\ d\bar{\theta}/d\alpha \end{bmatrix} = - \begin{bmatrix} \nabla_{x\alpha}^2 l \\ \bar{\Gamma} \end{bmatrix}.$$

\bar{R} is $(q_1 + q_3 + p_2 + p_4) \times n$ and consists of (signed) gradients of the active

constraints. By the regularity assumption and the results of the sensitivity theorem, it has full row rank.

By the results of the sensitivity theorem, the matrix

$$\begin{bmatrix} \nabla_x^2 l & \bar{R}^T \\ \bar{R} & 0 \end{bmatrix}$$

is nonsingular. Denote its inverse by

$$W \equiv \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^T & W_{22} \end{bmatrix}$$

(Note that W_{11} and W_{22} are symmetric).

Then

$$\begin{aligned} dx/d\alpha &= -[W_{11}(\nabla_{x\alpha}^2 l) + W_{12}\bar{\Gamma}] \\ d\bar{\theta}/d\alpha &= -[W_{12}^T(\nabla_{x\alpha}^2 l) + W_{22}\bar{\Gamma}]. \end{aligned}$$

Lemma 4.1: (see McCormick [30])

Let B be an $n \times (q_1 + q_3 + p_2 + p_4)$ basis for the null space of \bar{R} . Then

$$W_{11} = B[B^T(\nabla_x^2 l)B]^{-1}B^T$$

$$W_{12} = [I - W_{11}(\nabla_x^2 l)] \bar{R}^+$$

$$W_{22} = -W_{12}^T (\nabla_x^2 l) W_{12}$$

where \bar{R}^+ denotes the pseudoinverse of \bar{R} .

Note:

(i) Every column of B is orthogonal to the gradients of the active constraints.

Hence the second-order sufficiency conditions, which hold throughout the neighborhood, require that $B^T (\nabla_x^2 l) B$ be positive definite (and thus nonsingular).

(ii) Any generalized inverse \bar{R}^g may be used in place of \bar{R}^+ .

Before concluding this section, we note for later reference that the definition of the matrix W requires that

$$W_{11} \bar{R}^T = 0$$

$$W_{12}^T \bar{R}^T = I$$

4.3. Analysis of the Reduced Problem

Now we are ready to consider problem [II]. We shall assume that the hypotheses of the basic sensitivity theorem hold for problem (A) at every α of

interest. Then we have the existence and differentiability of an $x(\alpha)$ corresponding to each α .

4.4. The Gradient of the Reduced Functional

$$\text{Lemma 4.2: } \nabla_x f = -\bar{R}^T \bar{\theta}.$$

Pf: From the first-order conditions for problem (A),

$$\begin{aligned} 0 &= \nabla_x l \\ &= \nabla_x f - \sum_{i=1}^{p_1} u_i (\nabla_x g_i) - \sum_{i=1}^{p_2} v_i (\nabla_x c_i) \\ &\quad + \sum_{i=1}^{p_2} \lambda_i (\nabla_x h_i) + \sum_{i=1}^{p_4} \mu_i (\nabla_x d_i) \\ &= \nabla_x f - \sum_{i=1}^{q_1} u_i (\nabla_x g_i) - \sum_{i=1}^{q_2} v_i (\nabla_x c_i) \\ &\quad + \sum_{i=1}^{p_2} \lambda_i (\nabla_x h_i) + \sum_{i=1}^{p_4} \mu_i (\nabla_x d_i) \\ &= \nabla_x f + \bar{R}^T \bar{\theta} \end{aligned}$$

Lemma 4.3: Let

$$\begin{aligned} \Gamma^T &= [\nabla_\alpha g \mid -\nabla_\alpha c \mid \nabla_\alpha h \mid \nabla_\alpha d] \\ &= [-\nabla_\alpha g \mid 0 \mid \nabla_\alpha h \mid 0] \end{aligned}$$

and

$$\theta = (u_1, \dots, u_{p_1}, v_1, \dots, v_{p_2}, \lambda_1, \dots, \lambda_{p_3}, \mu_1, \dots, \mu_{p_4})^T.$$

$$\text{Then } \Gamma^T \theta = \bar{\Gamma}^T \bar{\theta}$$

Pf: Each of the extra terms in $\Gamma^T \theta$ contains a multiplier corresponding to a nonbinding constraint and thus must be zero.

$$\text{Lemma 4.4: } \nabla_{\alpha} f(\alpha) = \nabla_{\alpha} f(x(\alpha), \alpha) + \Gamma^T \theta$$

Pf:

$$\begin{aligned} \nabla_{\alpha} f &= \nabla_{\alpha} f + (dx/d\alpha)^T (\nabla_x f) \\ &= \nabla_{\alpha} f + [(\nabla_{\alpha x}^2 f) W_{11} + \bar{\Gamma}^T W_{12}^T] \bar{R}^T \bar{\theta} \\ &= \nabla_{\alpha} f + \bar{\Gamma}^T \bar{\theta} \\ &= \nabla_{\alpha} f + \Gamma^T \theta \end{aligned}$$

Corollary 4.5: If no constraint involves both x and α , then

$$\nabla_{\alpha} f(\alpha) = \nabla_{\alpha} f(x(\alpha), \alpha)$$

4.5. The Hessian of the Reduced Functional

Lemma 4.6:

$$\begin{aligned} \nabla_{\alpha}^2 f &= \nabla_{\alpha}^2 f - [\nabla_{\alpha x}^2 f \mid \bar{\Gamma}^T] \begin{bmatrix} \nabla_x^2 f & \bar{R}^T \\ \bar{R} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{x\alpha}^2 f \\ \bar{\Gamma} \end{bmatrix} \\ &= \begin{bmatrix} (dx/d\alpha)^T & \mid & I \end{bmatrix} \begin{bmatrix} \nabla_x^2 f & \nabla_{\alpha x}^2 f \\ \nabla_{x\alpha}^2 f & \nabla_{\alpha}^2 f \end{bmatrix} \begin{bmatrix} dx/d\alpha \\ I \end{bmatrix} \end{aligned}$$

where the quantities on the right side of the equations are evaluated at the point $(x(\alpha), \alpha)$.

Pf: Differentiate the expression $\nabla_{\alpha} f = \nabla_{\alpha} f + \Gamma^T \theta$, which must hold at every value of α .

$$\begin{aligned} \nabla_{\alpha} f &= \nabla_{\alpha} f - \sum_{i=1}^{p_1} u_i (\nabla_{\alpha} g_i) - \sum_{i=1}^{p_2} v_i (\nabla_{\alpha} c_i) \\ &\quad + \sum_{i=1}^{p_3} \lambda_i (\nabla_{\alpha} h_i) + \sum_{i=1}^{p_4} \mu_i (\nabla_{\alpha} d_i) \end{aligned}$$

$$\begin{aligned} \nabla_{\alpha}^2 f &= \nabla_{\alpha}^2 f + (dx/d\alpha)^T (\nabla_{x\alpha}^2 f) \\ &\quad - \sum_{i=1}^{p_1} [(\nabla_{\alpha} g_i)(du_i/d\alpha) + u_i (\nabla_{\alpha}^2 g_i) \\ &\quad \quad + u_i (dx/d\alpha)^T (\nabla_{x\alpha}^2 g_i)] \\ &\quad - \sum_{i=1}^{p_2} [(\nabla_{\alpha} c_i)(dv_i/d\alpha) + v_i (\nabla_{\alpha}^2 c_i) \\ &\quad \quad + v_i (dx/d\alpha)^T (\nabla_{x\alpha}^2 c_i)] \end{aligned}$$

$$\begin{aligned}
& + \sum_{i=1}^{p_2} [(\nabla_{\alpha} h_i)(d\lambda_i/d\alpha) + \lambda_i(\nabla_{\alpha}^2 h_i) \\
& \quad + \lambda_i(dx/d\alpha)^T(\nabla_{x\alpha}^2 h_i)] \\
& + \sum_{i=1}^{p_1} [(\nabla_{\alpha} d_i)(d\mu_i/d\alpha) + \mu_i(\nabla_{\alpha}^2 d_i) \\
& \quad + \mu_i(dx/d\alpha)^T(\nabla_{x\alpha}^2 d_i)] \\
& = \nabla_{\alpha}^2 l + (dx/d\alpha)^T(\nabla_{x\alpha}^2 l) + (d\theta/d\alpha)^T \bar{\Gamma} \\
& = \nabla_{\alpha}^2 l + (dx/d\alpha)^T(\nabla_{x\alpha}^2 l) + (d\bar{\theta}/d\alpha)^T \bar{\Gamma} \\
& = \nabla_{\alpha}^2 l - \left[\nabla_{\alpha x}^2 l \mid \bar{\Gamma}^T \right] \begin{bmatrix} \nabla_x^2 l & \bar{R}^T \\ \bar{R} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{x\alpha}^2 l \\ \bar{\Gamma} \end{bmatrix}.
\end{aligned}$$

To show the second equality, note that

$$\begin{aligned}
(dx/d\alpha)^T(\nabla_x^2 l)(dx/d\alpha) & = (\nabla_{\alpha x}^2 l) W_{11} (\nabla_x^2 l) W_{11} (\nabla_{x\alpha}^2 l) \\
& \quad + (\nabla_{\alpha x}^2 l) W_{11} (\nabla_x^2 l) W_{12} \bar{\Gamma} \\
& \quad + \bar{\Gamma}^T W_{12}^T (\nabla_x^2 l) W_{11} (\nabla_{x\alpha}^2 l) \\
& \quad + \bar{\Gamma}^T W_{12}^T (\nabla_x^2 l) W_{12} \\
& = (\nabla_{x\alpha}^2 l) W_{11} (\nabla_{x\alpha}^2 l) - \bar{\Gamma}^T W_{22} \bar{\Gamma}
\end{aligned}$$

and

$$\begin{aligned}
(d\bar{\theta}/d\alpha)^T \bar{\Gamma} & = (\nabla_{\alpha x}^2 l) W_{12} \bar{\Gamma} - \bar{\Gamma}^T W_{22} \bar{\Gamma} \\
& = (\nabla_{\alpha x}^2 l) (dx/d\alpha) + (\nabla_{\alpha x}^2 l) W_{11} (\nabla_{x\alpha}^2 l) \\
& \quad - \bar{\Gamma}^T W_{22} \bar{\Gamma} \\
& = (dx/d\alpha)^T (\nabla_x^2 l) (dx/d\alpha) + (\nabla_{\alpha x}^2 l) (dx/d\alpha) .
\end{aligned}$$

4.8. The Correspondence Between the Original and Reduced Problems

Theorem 4.7:

- (1) Let the function f in problem [I] be twice continuously differentiable in x , and assume that its gradient with respect to x is continuously differentiable in α .
- (2) Let each of the constraints present in the problem be continuously differentiable in its arguments.
- (3) Assume that, for every $\hat{\alpha}$, the subproblem (A) has a solution $x(\hat{\alpha})$ such that
 - (a) the second-order sufficiency conditions for a local minimizer of problem (A) hold at $x(\hat{\alpha})$ (with appropriate Lagrange multipliers),
 - (b) the gradients (with respect to x) of those constraints of problem (A) which are binding at $x(\hat{\alpha})$ are linearly independent,
 - (c) strict complementarity holds for problem (A) at $x(\hat{\alpha})$.

Then problems [I] and [II] are related in the following way:

- (i) Let (x^*, α^*) be a global minimizer of problem [I]. Then x^* satisfies the first-order conditions for problem (A), α^* is a global minimizer of problem [II], and $f(\alpha^*) = f(x^*, \alpha^*)$. Furthermore, if there is a unique x^* among the pairs (x^*, α^*) yielding the (same) minimal value of f , then $x^* = x(\alpha^*)$.
- (ii) Let α^* satisfy the first-order conditions for problem [II]. Then $(x(\alpha^*), \alpha^*)$ satisfies the first-order conditions for problem [I].

Pf: We have previously defined the Lagrangian for problem (A) as

$$\begin{aligned} l(x, u, v, \lambda, \mu; \alpha) = & f(x, \alpha) - \sum_{i=1}^{p_1} u_i g_i(x, \alpha) \\ & - \sum_{i=1}^{p_2} v_i c_i(x) + \sum_{i=1}^{p_3} \lambda_i h_i(x, \alpha) \\ & + \sum_{i=1}^{p_4} \mu_i d_i(x) \end{aligned}$$

Let the Lagrangians for problems [I] and [II] be, respectively,

$$\begin{aligned} L_I(x, \alpha, \hat{u}, \hat{v}, \hat{w}, \hat{\lambda}, \hat{\mu}, \hat{\nu}) = & f(x, \alpha) - \sum_{i=1}^{p_1} \hat{u}_i g_i(x, \alpha) - \sum_{i=1}^{p_2} \hat{v}_i c_i(x) \\ & - \sum_{i=1}^{p_3} \hat{w}_i r_i(\alpha) + \sum_{i=1}^{p_4} \hat{\lambda}_i h_i(x, \alpha) \\ & + \sum_{i=1}^{p_5} \hat{\mu}_i d_i(x) + \sum_{i=1}^{p_6} \hat{\nu}_i s_i(\alpha) \end{aligned}$$

$$L_{II}(\alpha, w, \nu) = f(\alpha) - \sum_{i=1}^{p_3} w_i r_i(\alpha) + \sum_{i=1}^{p_6} \nu_i s_i(\alpha)$$

- (i) Let (x^*, α^*) be a global minimizer of problem [I].

Then there are multipliers

$$\hat{u}^*, \hat{v}^*, \hat{w}^*, \hat{\lambda}^*, \hat{\mu}^*, \hat{\nu}^*$$

such that

$$\nabla_x L_I(x^*, \alpha^*, \hat{u}^*, \hat{v}^*, \hat{w}^*, \hat{\lambda}^*, \hat{\mu}^*, \hat{\nu}^*) = 0$$

$$\nabla_\alpha L_I(x^*, \alpha^*, \hat{u}^*, \hat{v}^*, \hat{w}^*, \hat{\lambda}^*, \hat{\mu}^*, \hat{\nu}^*) = 0$$

$$g_i(x^*, \alpha^*) \geq 0 \quad i = 1, \dots, p_1$$

$$h_i(x^*, \alpha^*) = 0 \quad i = 1, \dots, p_2$$

$$c_i(x^*) \geq 0 \quad i = 1, \dots, p_3$$

$$d_i(x^*) = 0 \quad i = 1, \dots, p_4$$

$$r_i(\alpha^*) \geq 0 \quad i = 1, \dots, p_5$$

$$s_i(\alpha^*) = 0 \quad i = 1, \dots, p_6$$

$$\hat{u}_i^* g_i(x^*, \alpha^*) = 0 \quad i = 1, \dots, p_1$$

$$\hat{v}_i^* c_i(x^*) = 0 \quad i = 1, \dots, p_3$$

$$\hat{w}_i^* r_i(\alpha^*) = 0 \quad i = 1, \dots, p_5$$

$$\hat{\lambda}_i^* \geq 0 \quad i = 1, \dots, p_2$$

$$\hat{\mu}_i^* \geq 0 \quad i = 1, \dots, p_4$$

$$\hat{\nu}_i^* \geq 0 \quad i = 1, \dots, p_6$$

Clearly, x^* and α^* satisfy the constraints of problem [II] and subproblem

(A).

Identify corresponding multipliers. That is, let

$$\begin{aligned} u_i^* &= \hat{u}_i^* & i &= 1, \dots, p_1 \\ v_i^* &= \hat{v}_i^* & i &= 1, \dots, p_3 \\ w_i^* &= \hat{w}_i^* & i &= 1, \dots, p_5 \\ \lambda_i^* &= \hat{\lambda}_i^* & i &= 1, \dots, p_2 \\ \mu_i^* &= \hat{\mu}_i^* & i &= 1, \dots, p_4 \\ \nu_i^* &= \hat{\nu}_i^* & i &= 1, \dots, p_6 \end{aligned}$$

Then problem [II] and subproblem (A) have nonnegative multipliers corresponding to inequality constraints, and complementarity holds.

We have that

$$\begin{aligned} 0 &= \nabla_x L_I(x^*, \alpha^*, \hat{u}^*, \hat{v}^*, \hat{w}^*, \hat{\lambda}^*, \hat{\mu}^*, \nu^*) \\ &= \nabla_x f(x^*, \alpha^*) - \sum_{i=1}^{p_1} \hat{u}_i^* \nabla_x g_i(x^*, \alpha^*) \\ &\quad - \sum_{i=1}^{p_3} \hat{v}_i^* \nabla_x c_i(x^*) + \sum_{i=1}^{p_2} \hat{\lambda}_i^* \nabla_x h_i(x^*, \alpha^*) \\ &\quad + \sum_{i=1}^{p_4} \hat{\mu}_i^* \nabla_x d_i(x^*) \\ &= \nabla_x l(x^*, u^*, v^*, \lambda^*, \mu^*; \alpha^*) \end{aligned}$$

by the identification of multipliers.

Hence, x^* satisfies the first-order conditions for (A).

By the definition of subproblem (A), we must have that

$$f(x(\alpha^*), \alpha^*) = f(x^*, \alpha^*).$$

So $(x(\alpha^*), \alpha^*)$ is also a global minimizer for problem [I].

Clearly, if (x^*, α^*) is the unique global minimizer, then $x^* = x(\alpha^*)$.

$$\begin{aligned} \nabla_\alpha L_{II}(\alpha^*, w^*, \nu^*) &= \nabla_\alpha f(\alpha^*) - \sum_{i=1}^{p_5} w_i^* \nabla_\alpha r_i(\alpha^*) \\ &\quad + \sum_{i=1}^{p_6} \nu_i^* \nabla_\alpha s_i(\alpha^*) \\ &= \nabla_\alpha f(\alpha^*, x(\alpha^*)) - \sum_{i=1}^{p_1} u_i^* \nabla_\alpha g_i(\alpha^*, x(\alpha^*)) \\ &\quad + \sum_{i=1}^{p_3} \lambda_i^* \nabla_\alpha h_i(\alpha^*, x(\alpha^*)) - \sum_{i=1}^{p_2} w_i^* \nabla_\alpha r_i(\alpha^*) \\ &\quad + \sum_{i=1}^{p_4} \nu_i^* \nabla_\alpha s_i(\alpha^*) \\ &= \nabla_\alpha L_I(x(\alpha^*), \alpha^*, u^*, v^*, w^*, \lambda^*, \mu^*, \nu^*) \\ &= 0 \end{aligned}$$

(with the correct identification of multipliers).

Hence, α^* satisfies the first-order conditions for problem [II].

Suppose that there is some feasible β such that

$$f(\beta) < f(\alpha^*)$$

Then, by the definition of f ,

$$f(x(\beta), \beta) > f(x(\alpha^*), \alpha^*)$$

contradicting the fact that $(x(\alpha^*), \alpha^*)$ is a global minimizer of problem [I].

Therefore, α^* is a global minimizer of problem [II].

(ii) Let α^* satisfy the first-order conditions for problem [II].

Define $x^* = x(\alpha^*)$.

Then there exist multipliers $u^*, v^*, w^*, \lambda^*, \mu^*, \nu^*$ such that

$$\nabla_{\alpha} L_{II}(\alpha^*, w^*, \nu^*) = 0$$

$$r_i(\alpha^*) \geq 0 \quad i = 1, \dots, p_5$$

$$s_i(\alpha^*) = 0 \quad i = 1, \dots, p_6$$

$$\lambda_i^* r_i(\alpha^*) = 0 \quad i = 1, \dots, p_5$$

$$w_i^* \geq 0 \quad i = 1, \dots, p_5$$

$$\nabla_x l(x^*, u^*, v^*, \lambda^*, \mu^*; \alpha^*) = 0$$

$$g_i(x^*, \alpha^*) \geq 0 \quad i = 1, \dots, p_1$$

$$h_i(x^*, \alpha^*) = 0 \quad i = 1, \dots, p_2$$

$$c_i(x^*) \geq 0 \quad i = 1, \dots, p_3$$

$$d_i(x^*) = 0 \quad i = 1, \dots, p_4$$

$$u_i^* g_i(x^*, \alpha^*) = 0 \quad i = 1, \dots, p_1$$

$$v_i^* c_i(x^*) = 0 \quad i = 1, \dots, p_3$$

$$u_i^* \geq 0 \quad i = 1, \dots, p_1$$

$$v_i^* \geq 0 \quad i = 1, \dots, p_3$$

Again, identification of the corresponding multipliers gives the constraint and multiplier conditions for problem [II].

$$\begin{aligned} \nabla_x L_I(x^*, \alpha^*, u^*, v^*, w^*, \lambda^*, \mu^*, \nu^*) \\ &= \nabla_x l(x^*, u^*, v^*, \lambda^*, \mu^*; \alpha^*) \\ &= 0 \end{aligned}$$

And finally,

$$\begin{aligned} 0 &= \nabla_{\alpha} L_{II}(\alpha^*, w^*, \nu^*) \\ &= \nabla_{\alpha} L_I(x(\alpha^*), \alpha^*, u^*, v^*, w^*, \lambda^*, \mu^*, \nu^*) \end{aligned}$$

from part (i).

CHAPTER 5

Some Special Cases

In chapter 4, we showed that certain general nonlinear programming problems of type [I] have associated with them reduced problems of type [II]. We propose using problem [II] to determine a point which satisfies the first-order conditions for problem [I], and we have derived expressions for the gradient and Hessian of the reduced functional, f . At this point, it is instructive to consider special cases of problem [I] in which the complexity of the calculations is reduced.

A study of simple cases is not just an exercise in the application of the general results. Since subproblem (A) must be solved each time the reduced functional, f , is evaluated, it is reasonable to assume that we will be interested in solving the reduced functional version of the problem only in cases in which [II] or (A) is particularly easy to solve.

5.1. Effect of Constraint Presence or Absence

We will use the following notation:

Constraints such as c and d , which involve x only, will be called **x -constraints**.

Constraints such as g and h , which involve both x and α , will be called **mixed constraints**.

Constraints such as r and s , which involve α only, will be called **α -constraints**.

We have already shown that in the absence of mixed constraints the gradient of the reduced functional has the simple form

$$\nabla_{\alpha} f(\alpha) = \nabla_{\alpha} f(x(\alpha), \alpha) .$$

The presence or absence of the various constraint types influences the complexity of the routines which must be used to solve problem [II] and its subproblem (A). We are particularly interested in cases in which one of these problems is unconstrained. The following table characterizes the simplifications which occur when certain of the constraint types are absent.

Type of Constraints Present	Problem II	Subproblem A	$\nabla_{\alpha} f$
x, α , mixed	constrained	constrained	$\nabla_{\alpha} f + \Gamma^T \theta$
α , mixed	constrained	constrained	$\nabla_{\alpha} f + \Gamma^T \theta$
x , mixed	unconstrained	constrained	$\nabla_{\alpha} f + \Gamma^T \theta$
x, α	constrained	constrained	$\nabla_{\alpha} f$
mixed	unconstrained	constrained	$\nabla_{\alpha} f + \Gamma^T \theta$
α	constrained	unconstrained	$\nabla_{\alpha} f$
x	unconstrained	constrained	$\nabla_{\alpha} f$
none	unconstrained	unconstrained	$\nabla_{\alpha} f$

5.2. The Case of n Active Constraints

Suppose that, for a given value of α , there are precisely n constraints of the subproblem active at $x(\alpha)$ and that the gradients of these constraints are linearly independent. Then the matrix \bar{R} of active constraint gradients is nonsingular and

$$\begin{bmatrix} \nabla_x^2 l & \bar{R}^T \\ \bar{R} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & \bar{R}^{-1} \\ \bar{R}^{-T} & -\bar{R}^{-T} (\nabla_x^2 l) \bar{R}^{-1} \end{bmatrix}$$

so that

$$dx/d\alpha = -\bar{R}^{-1} \bar{\Gamma}$$

The gradient of f is still

$$\nabla_{\alpha} f = \nabla_{\alpha} f + \Gamma^T \theta,$$

but the Hessian reduces to

$$\begin{aligned} \nabla_{\alpha}^2 f &= \nabla_{\alpha}^2 l - (\nabla_{\alpha x}^2 l) \bar{R}^{-1} \bar{\Gamma} \\ &\quad - \bar{\Gamma}^T \bar{R}^{-T} (\nabla_{xx}^2 l) \\ &\quad + \bar{\Gamma}^T \bar{R}^{-T} (\nabla_x^2 l) \bar{R}^{-1} \bar{\Gamma} \end{aligned}$$

5.2.1. n binding constraints; no mixed constraints

If, in addition, none of the constraints in the subproblem involves α , then

$$\nabla_{\alpha} f = \nabla_{\alpha} f$$

and

$$\nabla_{\alpha}^2 f = \nabla_{\alpha}^2 f.$$

5.2.2. n binding constraints; n equality constraints

Let \hat{x} be a feasible point at which none of the inequality constraints is binding. Assume that there are exactly n equality constraints. Then we have n linearly independent constraint gradients.

Lemma 5.1: Let $\hat{\alpha}$ be given. Suppose \hat{x} satisfies:

$$h(\hat{x}, \hat{\alpha}) = 0$$

$$d(\hat{x}) = 0$$

$$g(\hat{x}, \hat{\alpha}) > 0$$

$$c(\hat{x}) > 0$$

$$\bar{R}^T = [\nabla_x h \mid \nabla_x d] \text{ is nonsingular.}$$

Then \hat{x} satisfies the first-order conditions for subproblem (A).

Pf: The Lagrangian for the subproblem is

$$l(x, u, v, \lambda, \mu; \alpha) = f(x, \alpha) - u^T g - v^T c \\ + \lambda^T h + \mu^T d.$$

Let the vector of Lagrange multipliers be

$$\theta = \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{\lambda} \\ \hat{\mu} \end{bmatrix}.$$

Let

$$\hat{u} = 0, \quad \hat{v} = 0, \quad \text{and} \\ \bar{\theta} = \begin{bmatrix} \hat{\lambda} \\ \hat{\mu} \end{bmatrix} = -\bar{R}^T (\nabla_x f).$$

The constraints are satisfied, complementarity holds, and the multipliers associated with inequality constraints are nonnegative.

$$\begin{aligned} \nabla_x l &= \nabla_x f - (\nabla_x g) u - (\nabla_x c) v \\ &\quad + (\nabla_x h) \lambda + (\nabla_x d) \mu \\ &= \nabla_x f + \bar{R}^T \bar{\theta} \\ &= \nabla_x f - \bar{R}^T \bar{R}^T (\nabla_x f) \end{aligned}$$

$$= 0$$

5.3. Special Structure in the Subproblem

We would like to choose the partitioning of the variables into x and α in such a way that subproblem (A) has a unique and differentiable solution for every α and that the subproblem has a structure which may be used to speed its solution.

5.3.1. Solution of (A) is Unnecessary

Suppose $x(\alpha)$ is an explicit function of α . In this case, x can be eliminated from the problem entirely (as in the unconstrained semilinear least squares problem).

5.3.2. (A) is a simple unconstrained problem

5.3.2.1. The Objective Function of (A) is Quadratic

$$f(x, \alpha) = a(\alpha) + b(\alpha)^T x + \frac{1}{2} x^T C(\alpha) x$$

Assume that $C(\alpha)$ is positive definite for all α .

Then the objective function for the subproblem is

$$t(x) = a + b^T x + \frac{1}{2} x^T C x$$

(where a, b, C denote the corresponding functions evaluated at the current value of α).

and hence $x(\alpha)$ solves

$$C x(\alpha) = -b.$$

So in this case, determination of $x(\alpha)$ requires only the solution of a linear system.

5.3.2.2. Unconstrained Semilinear Least Squares

In this case, subproblem (A) is a linear least squares problem. It can be solved at each step, or the minimal norm solution can be chosen and the variable x eliminated.

The Hessian approximations used in the various algorithms discussed in chapter 3 can be related to the Hessian of the original function through the equation

$$\nabla_{\alpha}^2 f = \begin{bmatrix} (dx/d\alpha)^T & 1 \end{bmatrix} \begin{bmatrix} \nabla_x^2 f & \nabla_{x\alpha}^2 f \\ \nabla_{\alpha x}^2 f & \nabla_{\alpha}^2 f \end{bmatrix} \begin{bmatrix} dx/d\alpha \\ 1 \end{bmatrix}$$

(See Ruhe and Wedin [52]).

5.3.3. (A) is a Linear Programming Problem

In this case, $\nabla_x^2 l = 0$. If there are fewer than n linearly independent constraint gradients, then the second-order sufficiency conditions fail to hold and the inverse matrix W does not exist. In fact, $x(\alpha)$ may not even be continuous. So assume that $x(\alpha)$ is a nondegenerate solution, with n linearly independent binding constraint gradients.

Then

$$\begin{bmatrix} \nabla_x^2 l & \bar{R}^T \\ \bar{R} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & \bar{R}^{-1} \\ \bar{R}^{-T} & 0 \end{bmatrix}$$

so that

$$dx/d\alpha = -\bar{R}^{-1} \bar{\Gamma}$$

Also,

$$\nabla_{\alpha} f = \nabla_{\alpha} f + \bar{\Gamma}^T \bar{\theta}$$

and

$$\nabla_{\alpha}^2 f = \nabla_{\alpha}^2 l - (\nabla_{\alpha}^2 l) \bar{R}^{-1} \bar{\Gamma} - \bar{\Gamma}^T \bar{R}^{-T} (\nabla_{x\alpha}^2 l)$$

If the subproblem constraints do not involve α , then $\bar{\Gamma} = 0$ and hence

$$\nabla_{\alpha} f = \nabla_{\alpha} f$$

$$\nabla_{\alpha}^2 f = \nabla_{\alpha}^2 f$$

5.3.4. (A) is a quadratic programming problem

In this case, the objective function of subproblem (A) is quadratic in x and its constraints are linear in x . Since efficient computer codes for solving quadratic programming problems are available, subproblems of this form are also easy to solve.

We defer further discussion of this type of problem until the next chapter, in which we will consider an important special case.

CHAPTER 6

Constrained Semilinear Least Squares Problems

6.1. Statement of the Problem

We will now consider the case in which the original objective function f has the form

$$f(x, \alpha) = \frac{1}{2} F(x, \alpha)^T F(x, \alpha)$$

$$\text{with } F(x, \alpha) = A(\alpha)x - b(\alpha).$$

The subproblem which determines $x(\alpha)$ will be a constrained linear least squares problem. So we will restrict attention to the case in which the constraints are linear in x .

The full functional constrained semilinear least squares problem is

$$\begin{aligned} \text{CSLS}[I] \quad & \underset{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^k}{\text{minimize}} \quad f(x, \alpha) = \frac{1}{2} \|A(\alpha)x - b\|^2 \\ & \text{subject to } G(\alpha)x - \beta(\alpha) \geq 0 \end{aligned}$$

Lemma 6.1: $F(\alpha) = -P_{AB}^L(\alpha) [b(\alpha) - A(\alpha)\bar{R}^+(\alpha)\bar{\xi}(\alpha)]$.

Pf: Let \hat{x} be the solution of the subproblem.

Identify the binding and nonbinding constraints at \hat{x} :

$$\bar{R}\hat{x} - \bar{\xi} = 0 \quad (i)$$

$$-(\bar{R}\hat{x} - \bar{\xi}) > 0 \quad (ii)$$

Let $p = p_1 + p_2 + p_3 + p_4$ be the total number of constraints, and let

$q = q_1 + p_2 + q_3 + p_4$ be the number of binding constraints.

The general form of a vector \hat{x} satisfying (i) is

$$\hat{x} = \bar{R}^+\bar{\xi} + z'$$

where z' is an element of the null space of \bar{R} .

Expressing z' in terms of the basis B , we obtain

$$\hat{x} = \bar{R}^+\bar{\xi} + B\hat{z}$$

where \hat{z} is some vector in \mathbb{R}^q .

To determine \hat{z} , apply the change of variables

$$x = \bar{R}^+\bar{\xi} + Bz$$

to the subproblem.

The problem expressed in terms of z is

$$\underset{z \in \mathbb{R}^q}{\text{minimize}} \quad \frac{1}{2} \| (AB)z - b - A\bar{R}^+\bar{\xi} \|^2$$

$$\text{subject to } -\bar{R}Bz + (\bar{\xi} - \bar{R}\bar{R}^+\bar{\xi}) > 0$$

Then there is a vector of Lagrange multipliers $\zeta \in \mathbb{R}^{p-q}$ such that \hat{z} and ζ

satisfy the first-order conditions

$$(AB)^T(AB)\hat{z} - (AB)^T(b - A\bar{R}^+\bar{\xi}) + (\bar{R}B)^T\zeta = 0$$

$$-\bar{R}B\hat{z} + (\bar{\xi} - \bar{R}\bar{R}^+\bar{\xi}) \geq 0$$

$$\zeta_i [-\bar{R}_i B\hat{z} + (\bar{\xi}_i - \bar{R}_i \bar{R}^+ \bar{\xi})] = 0$$

(for each constraint)

$$\zeta \geq 0$$

Since the constraints represented in \bar{R} are nonbinding by definition, it must be true that $\zeta = 0$.

Then

$$(AB)^T(AB)\hat{z} = (AB)^T(b - A\bar{R}^+\bar{\xi})$$

Hence

$$\hat{z} = [(AB)^T(AB)]^{-1}(AB)^T(b - A\bar{R}^+\bar{\xi})$$

$$= (AB)^+(b - A\bar{R}^+\bar{\xi})$$

Finally,

$$F(\alpha) = A(\alpha)x(\alpha) - b(\alpha)$$

$$= A(\alpha)\hat{x} - b(\alpha)$$

$$= A(\alpha)B(\alpha)\hat{z} - [b(\alpha) - A(\alpha)\bar{R}^+(\alpha)\bar{\xi}(\alpha)]$$

$$= P_{AB}(\alpha) [b(\alpha) - A(\alpha)\bar{R}^+(\alpha)\bar{\xi}(\alpha)] - [b(\alpha) - A(\alpha)\bar{R}^+(\alpha)\bar{\xi}(\alpha)]$$

$$= -P_{AB}^L(\alpha) [b(\alpha) - A(\alpha)\bar{R}^+(\alpha)\bar{\xi}(\alpha)]$$

6.5. The Jacobian of the Reduced Functional

Now we are ready to give an expression for the Jacobian of F

Lemma 6.2:

$$J(\alpha) = P_{AB}^L(\alpha) [J_\alpha(\alpha) - A(\alpha)\bar{R}^+(\alpha)\bar{\Gamma}(\alpha)] \\ - P_{AB} [(A(\alpha)B(\alpha))^+]^T B(\alpha)K(\alpha)$$

where $K(\alpha) \equiv K(x(\alpha), \alpha)$

and $K(x, \alpha) \equiv [D_1^T(\alpha)] F(x, \alpha) - [D_G^T(\alpha)] u + [D_I^T(\alpha)] \lambda$.

Pf: $F(\alpha) = A(\alpha)x(\alpha) - b(\alpha)$

So

$$J(\alpha) = D_A(\alpha)x(\alpha) + A(\alpha)(dx(\alpha)/d\alpha) - D_b(\alpha)$$

Suppressing the argument, α , we have

$$J = J_\alpha + A(dx/d\alpha)$$

Recall that

$$\begin{aligned} \frac{dx}{d\alpha} &= -W_{11}(\nabla_{x\alpha}^2 f) - W_{12}\bar{\Gamma} \\ &= W_{11}[A^T A \bar{R}^+ \bar{\Gamma} - A^T J_\alpha - K] - \bar{R}^+ \bar{\Gamma} \\ &= B[(AB)^T(AB)]^{-1} B^T [A^T(A\bar{R}^+ \bar{\Gamma} - J_\alpha) - K] - \bar{R}^+ \bar{\Gamma} \\ &= B(AB)^+(A\bar{R}^+ \bar{\Gamma} - J_\alpha) - B[(AB)^T(AB)]^{-1} B^T K - \bar{R}^+ \bar{\Gamma} \end{aligned}$$

So

$$A \frac{dx}{d\alpha} = P_{AB}(A\bar{R}^+ \bar{\Gamma} - J_\alpha) - [(AB)^+]^T B^T K - A\bar{R}^+ \bar{\Gamma}$$

Then

$$\begin{aligned} J &= J_\alpha - A\bar{R}^+ \bar{\Gamma} - P_{AB}(J_\alpha - A\bar{R}^+ \bar{\Gamma}) - [(AB)^+]^T B^T K \\ &= P_{AB}^L(J_\alpha - A\bar{R}^+ \bar{\Gamma}) - P_{AB}[(AB)^+]^T B^T K \end{aligned}$$

6.6. A General Form for the Problem Structure

Unfortunately, the complicated expression obscures the structure in the Jacobian. So we will rewrite the basic form below. This general structure holds in both the unconstrained and constrained cases.

$$F = -P^\perp v$$

$$J = P^\perp \Psi + P \Phi$$

where the actual form of the vector v , the matrices Ψ and Φ , and the projection matrix P depends on the problem.

Consider the two terms in the expression for J . Since

$$(P_{AB}\Phi)^T(P_{AB}^\perp\Psi) = 0$$

and

$$(P_{AB}\Phi)^T F = 0,$$

we note that the structure which was used to advantage by Kaufman [27] and by Barham and Drane [3] in the unconstrained case carries over to the general

$$H(\alpha)x - \delta(\alpha) = 0$$

$$Cx - \rho = 0$$

$$Dx - \sigma \geq 0$$

$$r(\alpha) \geq 0$$

$$s(\alpha) = 0$$

The corresponding reduced problem is

$$CSLS[II] \quad \underset{\alpha \in \mathbb{R}^t}{\text{minimize}} \ f(\alpha) = \frac{1}{2} F^T F$$

$$\text{subject to } r(\alpha) \geq 0$$

$$s(\alpha) = 0$$

$$\text{where } F(\alpha) \equiv A(\alpha)x(\alpha) - b(\alpha)$$

$x(\alpha)$ solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ t(x) \equiv \frac{1}{2} \|A(\alpha)x - b(\alpha)\|^2$$

$$\text{subject to } G(\alpha)x - \beta(\alpha) \geq 0$$

$$H(\alpha)x - \delta(\alpha) = 0$$

$$Cx - \rho \geq 0$$

$$Dx - \sigma = 0$$

6.2. Notation

We will use the notation developed in chapter 3. Recall that the matrix R contains the (signed) gradients with respect to x of the constraints. In the present case,

$$R = \begin{bmatrix} -G(\alpha) \\ -C \\ H(\alpha) \\ D \end{bmatrix}$$

For convenience, define

$$\xi = \begin{bmatrix} -\beta(\alpha) \\ -\delta(\alpha) \\ \rho \\ \sigma \end{bmatrix}$$

As before, \bar{R} and $\bar{\xi}$ will denote quantities related to binding constraints, while \tilde{R} and $\tilde{\xi}$ will refer to nonbinding constraints. B will be a basis for the null space of \bar{R} .

The Jacobian of the original matrix function F will be denoted

$$J \equiv [J_x \ J_\alpha]$$

where

$$J_x = A$$

$$J_\alpha = (D1)x - (D\delta)$$

while the Jacobian of F is J .

The Lagrangian of the subproblem is

$$\begin{aligned}
 l(x, u, v, \lambda, \mu; \alpha) &= \frac{1}{2} \|Ax - b\|^2 - u^T(Gx - \beta) - v^T(Cx - \rho) \\
 &\quad + \lambda^T(Hx - \delta) + \mu^T(Dx - \sigma) \\
 &= \frac{1}{2} \|Ax - b\|^2 + \theta^T(Rx - \xi)
 \end{aligned}$$

where θ is the vector of Lagrange multipliers.

The Hessian of l has the following components

$$\begin{aligned}
 \nabla_x^2 l &= A^T A \\
 \nabla_{x\alpha}^2 l &= A^T J_\alpha + K \\
 \nabla_{\alpha x}^2 l &= J_\alpha^T A + K^T \\
 \nabla_\alpha^2 l &= J_\alpha^T J_\alpha + S_\alpha
 \end{aligned}$$

where $K \equiv (DA^T)(Ax - b) - (DG^T)u + (DH^T)\lambda$

$$\begin{aligned}
 S_\alpha &\equiv - \sum_{i=1}^{p_1} u_i (\nabla_{\alpha}^2 g_i) + \sum_{i=1}^{p_2} \lambda_i (\nabla_{\alpha}^2 h_i) \\
 &= -u^T [(D^2 G)x - (D^2 \beta)] + \lambda^T [(D^2 H)x - (D^2 \delta)]
 \end{aligned}$$

Note that only the term S_α involves second derivatives of the matrices defining the problem.

6.3. The Gradient and Hessian of the Reduced Functional

The gradient and Hessian of the reduced functional f have already been defined.

$$\nabla_\alpha f = J_\alpha^T (Ax - b) + \theta^T \Gamma$$

$$\text{where } \Gamma = \begin{bmatrix} -(\nabla_\alpha g)^T \\ 0 \\ (\nabla_\alpha h)^T \\ 0 \end{bmatrix} = \begin{bmatrix} -[(DG)x - D\beta] \\ 0 \\ (DH)x - D\delta \\ 0 \end{bmatrix}$$

Let

$$Y \equiv [(AB)^+]^T B^T K$$

$$Z \equiv A \bar{R}^+ \bar{\Gamma}$$

Then

$$\begin{aligned}
 \nabla_\alpha^2 f &= (J_\alpha^T J_\alpha + S_\alpha) - (-Z^T P_{AB}^\perp Z) \\
 &\quad - (J_\alpha^T P_{AB} J_\alpha + J_\alpha^T Y + Y^T J_\alpha + Y^T Y) \\
 &\quad - (J_\alpha^T Z - J_\alpha^T P_{AB} Z - Y^T Z + K^T \bar{R}^+ \bar{\Gamma}) \\
 &\quad - (Z^T J_\alpha - Z^T P_{AB} J_\alpha - Z^T Y + \bar{\Gamma}^T (\bar{R}^+)^T K) \\
 &= J_\alpha^T P_{AB}^\perp J_\alpha - J_\alpha^T Y - Y^T J_\alpha - Y^T Y - J_\alpha^T P_{AB}^\perp Z - Z^T P_{AB}^\perp J_\alpha \\
 &\quad + Y^T Z + Z^T Y - K^T \bar{R}^+ \bar{\Gamma} - \bar{\Gamma}^T (\bar{R}^+)^T K + Z^T P_{AB}^\perp Z + S_\alpha
 \end{aligned}$$

6.4. The Residual of the Reduced Problem

Since the reduced problem is a nonlinear least squares problem, we would like to develop an expression for the Jacobian of F as well. We begin by investigating the form of F itself.

case as well. Since the second term does not contribute to $J^T F$, the full matrix J is not required in order to compute the gradient of f .

CHAPTER 7

Extension to Problems Without Special Structure

7.1. The Original Problem

In this chapter, we will consider the general problem

$$\begin{aligned} & \underset{y \in \mathbb{R}^n}{\text{minimize}} && f(y) \\ & \text{subject to} && h_i(y) = 0 \quad i = 1, \dots, p \\ & && l_y \leq y \leq u_y \\ & && \text{(where } l_y \text{ and } u_y \text{ are constant)} \end{aligned}$$

In which there is no obvious partitioning for the variables.

In the method discussed previously, we maintained the same partition of the variables throughout the solution process. However, the conditions implying existence, continuity, and differentiability of $x(\alpha)$ given α are all local. So if we can assume that these conditions hold at every point with any partition, then we are free to redefine the partitioning whenever it seems advantageous.

7.2. The Reduced Problem

Rewrite the given problem in the alternate form described in chapter 4.

For convenience, let

$$y = \begin{bmatrix} x \\ \alpha \end{bmatrix}, \quad l_y = \begin{bmatrix} l_x \\ l_\alpha \end{bmatrix}, \quad u_y = \begin{bmatrix} u_x \\ u_\alpha \end{bmatrix}$$

$$\text{minimize}_\alpha f(\alpha) \equiv f(x(\alpha), \alpha)$$

$$\text{subject to } l_\alpha \leq \alpha \leq u_\alpha$$

where $x(\alpha)$ solves

$$\text{minimize}_x t(x) = f(x; \alpha)$$

$$\text{subject to } h_i(x; \alpha) = 0 \quad i = 1, \dots, p$$

$$l_x \leq x \leq u_x$$

7.3. Restrictions on the Partitioning of the Variables

In this case, there is no natural structure in f which makes the subproblem easy to solve. So we will use the choice of a partition to simplify its structure.

We will require that

- (i) x has p elements
- (ii) there are exactly p binding constraints at the solution of the subproblem.

Note that if we assume regularity at every point for the original problem there can be at most m binding constraints at a given point and hence no more than $m-p$ variables can be at a bound. Thus the partitioning requirements above are reasonable.

7.4. Solution of the Subproblem

We will maintain all of the assumptions from the original method. In particular, the gradients of the binding constraints from the subproblem will be linearly independent. In the present case, there are p linearly independent constraint gradients.

The constraint derivative matrices for this problem are

$$R^T = \begin{bmatrix} -I_p & -I_p & \nabla_x h \end{bmatrix}, \quad \bar{R}^T = \nabla_x h$$

$$\Gamma^T = \begin{bmatrix} 0 & 0 & \nabla_\alpha h \end{bmatrix}, \quad \bar{\Gamma}^T = \nabla_\alpha h$$

The Lagrange multipliers are

$$\theta^T = \begin{bmatrix} v^T & w^T & \lambda^T \end{bmatrix}, \quad \bar{\theta} = \lambda$$

We showed in section (4.2.2) that

$$\bar{\theta} = -\bar{R}^{-T} (\nabla_x f)$$

and that any \hat{x} which satisfies conditions (i) and (ii) also satisfies the first-order conditions for the subproblem.

This means that, given $\hat{\alpha}$, we can define $x(\hat{\alpha})$ by (i) and (ii). The nonsingularity of \bar{R} guarantees differentiability of $x(\alpha)$ in a neighborhood of $\hat{\alpha}$ without requiring that $x(\hat{\alpha})$ satisfies the second-order sufficiency conditions. The value of $x(\hat{\alpha})$ can be determined by solving the equality constraints; no minimization of $t(x)$ is required.

7.5. The Reduced Gradient

The gradient of the reduced functional f at $\hat{\alpha}$ is given by

$$\begin{aligned}\nabla_{\alpha} f(\hat{\alpha}) &= \nabla_{\alpha} f + \bar{\Gamma}^T \bar{\theta} \\ &= \nabla_{\alpha} f - (\nabla_{\alpha} h)^T (\nabla_x h)^{-T} (\nabla_x f)\end{aligned}$$

(where the right-hand side is evaluated at $(x(\hat{\alpha}), \hat{\alpha})$).

This is called the reduced gradient and the solution technique we have just described is the generalized reduced gradient method.

The reduced gradient method, Wolfe [60], arose as an attempt to generalize the simplex method of linear programming to handle a nonlinear objective function. Note that when the constraints are linear the determination of $x(\alpha)$ involves only the solution of a linear system. This method was generalized to the case of nonlinear constraints by Abadie and Carpentier [1]. Since the solution methods for the minimization of f and for the determination of $x(\alpha)$ are not specified, these can be chosen to suit the type of problem being considered. For example, the implementation by Lasdon, Waren, Jain, and Ratner [33] involves a variable metric method and is suitable for moderately sized problems,

while Gabriele [20] uses a conjugate gradient technique for the solution of large sparse problems.

CHAPTER 8

The Solution of Reducible Nonlinear Programming Problems

8.1. Using the Ideas We Have Developed

For problems with appropriate structure, the technique of rewriting the original problem in reduced form can be useful to both numerical analysts and users of computer programs. The first group will be interested in tailoring numerical methods to take advantage of the structure of the problem, while the second group will be interested in tailoring the problem to fit the available solution techniques. Consider the constrained semilinear least squares problem of chapter 6 and assume that there are no constraints that involve α but not x . In its full form, this is a nonlinearly constrained nonlinear least squares problem. Since computer programs of this type are not widely available, the problem would probably have to be solved using a general method, and no advantage would be taken of the least squares structure. The reduced problem, on the other hand, requires an unconstrained nonlinear least squares program and a linearly constrained linear least squares problem.

Now consider solution of the unconstrained semilinear least squares problem of chapter 3. Since the formula for the gradient in this case is very simple, we might decide to use a general method which requires only first derivative and

function information. We would need to call an unconstrained linear least solver during each function evaluation in order to determine $x(\alpha)$, but our work in coding the function and gradient evaluation would be minimal.

Of course, this simple approach fails to take advantage of the least squares structure of the problem and also of the fact that we can actually eliminate x from the problem. Golub and Pereyra [21] have provided a routine which evaluates the Jacobian of the reduced functional. The user needs only to supply instructions for evaluating A and Df . Any suitable nonlinear least squares program may then be used to complete the solution. If desired, an approximate Jacobian could be generated instead (as suggested by Kaufman).

Finally, we might want to modify the nonlinear least squares solution technique itself. One of the main justifications for computing $J^T J$ in solving general nonlinear least squares problems is that we need to evaluate J in order to compute the gradient. In the present case, however, all that is needed for the formation of the gradient is the partial Jacobian J_α . Hence it might be worthwhile to compute $J_\alpha^T J_\alpha$ and develop a special secant approximation to the remaining terms of the Hessian of the variable projection functional.

8.2. Identification of Suitable Problems

We have given sufficient conditions for the uniqueness and differentiability of a local function $x(\alpha)$. These conditions are not necessary. We were able to solve the unconstrained semilinear least squares problem by assuming that A

had local constant rank rather than that A had full column rank. However, it is not wise to apply the reduction technique blindly. Consider, for example, a problem involving only mixed constraints. The points α at which the function will be evaluated will be determined by an unconstrained minimization technique and so are unrestricted. The subproblem, however, may not even have a feasible region for some values of α .

8.3. Choice of a Solution Technique for the Subproblem

If there are no mixed constraints, the subproblem exists only to provide values of x given α . In the presence of mixed constraints, however, the evaluation of the gradient of the reduced functional requires the vector of Lagrange multipliers associated with the solution of the subproblem at the current value of α . However, many of the solution methods which could be used to solve the subproblem use multiplier estimates and can return to the calling routine the values of the multipliers at the solution.

If all or part of the Hessian is to be computed, it may be necessary to have a basis for the null space of the active constraint gradients. In simple cases, this is easy to provide. If the only constraints are bound on the variables, then B will consist of unit vectors. In the case of "probability" constraints (i.e., the variables are nonnegative and sum to one), McCord [38] suggests a basis containing a row of the form $[-1, \dots, -1]$ along with appropriate unit vectors. When the constraints are more complicated, a solution technique that also operates in the null space seems appropriate. Although such routines usually

do not form the basis explicitly, it should be possible to make use of the same basis information in the determination of $x(\alpha)$ and the evaluation (or approximation) of the Hessian of f .

8.4. Suggestions for Further Investigation

The technique of problem reduction described above has been shown to be advantageous in certain special cases. More testing should be done to identify the types of problems for which this method is suitable and to determine appropriate solution techniques for the various parts of the reduced problem.

In the theoretical development, we assumed that the subproblem was solved exactly. Further analysis is needed to account for the actual accuracy to which this auxiliary problem will be solved. The effect of subproblem solution inaccuracy may help to specify problems which are suitable for this method and also solution techniques which should be used in solving for x in terms of α .

The close relationship between the solution of reducible problems by forming the alternate problem and the solution of general problems by the generalized reduced gradient method means that we should be able to take advantage of work that has been done in the area of reduced-gradient-type methods. This includes recovery from cases in which the subproblem is not well defined for certain points. Since the generalized reduced gradient method has been shown to be effective and robust in comparative studies, see Colville [7] and Schittkowski [53], the application of similar methods to problems with

special structure should also yield favorable results.

REFERENCES

- (1) Abadie, J. and J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the case of Nonlinear Constraints", in *Optimization*, (R. Fletcher, ed.), London, Academic Press, (1969), pp. 37-47.
- (2) Bank, B., J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Nonlinear Parametric Optimization*, Basel, Birkhauser Verlag, (1983).
- (3) Barham, R. H. and W. Drane, "An Algorithm for Least Squares Estimation of Nonlinear Parameters When Some of the Parameters are Linear", *Technometrics*, **14**, (1972), pp. 757-766.
- (4) Ben-Israel, Adi, "A Modified Newton-Raphson Method for the Solution of Systems of Equations", *Israel Journal of Mathematics*, **3**, (1965), pp. 94-99.
- (5) Bergmann, Peter G., Robert Penfield, Ralph Schiller, and Henry Zatkis, "The Hamiltonian of the General Theory of Relativity with Electromagnetic Field", *The Physical Review*, **80** (Second Series), (1950), pp. 81-88.
- (6) Boullion, Thomas L. and Patrick L. Odell, *Generalized Inverse Matrices*, New York, Wiley-Interscience, (1971).
- (7) Colville, A. R., "A Comparative Study on Nonlinear Programming Codes", IBM New York Scientific Center Report 320-2949, (1968).
- (8) Corradi, Corrado, "A Note on the Solution of Separable Nonlinear Least Squares Problems with Separable Nonlinear Equality Constraints", *SIAM Journal on Numerical Analysis*, **18**, (1981), pp. 1134-1138.
- (9) Decell, H. P., "On the Derivative of the Generalized Inverse of a Matrix", *Linear and Multilinear Algebra*, **1**, (1974), pp. 357-359.

- (10) Dennis, J. E., Jr., "Nonlinear Least Squares and Equations", in *The State of the Art in Numerical Analysis*, (D. Jacobs, ed.), London, Academic Press, (1977), pp. 269-312.
- (11) Dennis, J. E., Jr., "Algorithms for Nonlinear Fitting", in *Nonlinear Optimization 1981*, (M. J. D. Powell, ed.), London, Academic Press, (1982), pp. 67-78.
- (12) Dennis, J. E., Jr., D. M. Gay, and R. E. Welsch, "An Adaptive Nonlinear Least-Squares Algorithm", *ACM Transactions on Mathematical Software*, 7, (1981), pp. 348-368.
- (13) Dennis, J. E., Jr. and Robert B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood Cliffs, New Jersey, Prentice-Hall, Inc., (1983).
- (14) Fiacco, Anthony V., "Sensitivity Analysis for Nonlinear Programming Using Penalty Methods", *Mathematical Programming*, 10, (1976), pp. 287-311.
- (15) Fiacco, Anthony V., ed., *Mathematical Programming with Data Perturbations I*, New York, Marcel Dekker, (1982).
- (16) Fiacco, Anthony V., ed., *Mathematical Programming with Data Perturbations II*, New York, Marcel Dekker, (1983).
- (17) Fiacco, Anthony V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, New York, Academic Press, (1983).
- (18) Fiacco, Anthony V. and Garth P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, New York, John Wiley and Sons, (1968).
- (19) Fletcher, R. and Shirley A. Lill, "A Class of Methods for Nonlinear Programming; II: Computational Experience", in *Nonlinear Programming*, (J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds.), London, Academic Press, (1970), pp. 67-92.
- (20) Gabriele, G. A., *Large Scale Nonlinear Programming Using the Generalized Reduced Gradient Method*, Ph.D. Dissertation, Department of Mechanical Engineering, Purdue University, (1980).

- (21) Golub, G. H. and V. Pereyra, "The Differentiation of Pseudoinverses and Nonlinear Least Squares Problems Whose Variables Separate", Technical Report STAN-CS-72-261, Computer Science Department, Stanford University, (February 1972).
- (22) Golub, G. H. and V. Pereyra, "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate", *SIAM Journal on Numerical Analysis*, 10, (1973), pp. 413-432.
- (23) Golub, G. H. and V. Pereyra, "Differentiation of Pseudoinverses, Separable Nonlinear Least Square Problems and Other Tales", in *Generalized Inverses and Applications*, (M. Nashed, ed.), New York, Academic Press, (1976), pp. 303-324.
- (24) Guttman, I., V. Pereyra, and H. D. Scolnik, "Least Squares Estimation for a Class of Nonlinear Models", *Technometrics*, 15, (1973), pp. 209-218.
- (25) Hartley, H. O., "The Modified Gauss-Newton Method for the Fitting of Non-Linear Regression Functions by Least Squares", *Technometrics*, 3, (1961), pp. 269-280.
- (26) Hearon, John Z. and J. W. Evans, "Differentiable Generalized Inverses", *Journal of Research of the National Bureau of Standards*, 72B, (1968), pp. 109-113.
- (27) Kaufman, Linda, "A Variable Projection Method for Solving Separable Nonlinear Least Squares Problems", *BIT*, 15, (1975), pp. 49-57.
- (28) Kaufman, Linda and Victor Pereyra, "A Method for Separable Nonlinear Least Squares Problems with Separable Nonlinear Equality Constraints", *SIAM Journal on Numerical Analysis*, 15, (1978), pp. 12-20.
- (29) Kirkegaard, Peter, Morton Eldrup, Ole E. Mogensen, and Niels J. Pedersen, "Program System for Analysing Positron Lifetime Spectra and Angular Correlation Curves", *Computer Physics Communications*, 23, (1981), pp. 307-335.
- (30) Krogh, Fred T., "Efficient Implementation of a Variable Projection Algorithm for Nonlinear Squares Problems", *Communications of the ACM*, 17, (1974), pp. 167-169.

- (31) Kruskal, J. B., "Nonmetric Multidimensional Scaling: A Numerical Method", *Psychometrika*, **29**, (1964), pp. 1-27 and 115-129.
- (32) Lasdon, L. S., "Reduced Gradient Methods", in *Nonlinear Optimization 1981*, (M. J. D. Powell, ed.), London, Academic Press, (1982), pp. 243-250.
- (33) Lasdon, L. S., A. D. Waren, A. Jain, and M. Ratner, "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming", *ACM Transactions on Mathematical Software*, **4**, (1978), pp. 34-50.
- (34) Lawson, Charles L. and Richard J. Hanson, *Solving Least Squares Problems*, Englewood Cliffs, New Jersey, Prentice Hall, Inc. (1974).
- (35) Lawton, William H. and Edward A. Sylvestre, "Elimination of Linear Parameters in Nonlinear Regression", *Technometrics*, **13**, (1971), pp. 461-467.
- (36) Lawton, W. H., E. A. Sylvestre, and M. S. Maggio, "Self Modeling Nonlinear Regression", *Technometrics*, **14**, (1972), pp. 513-532.
- (37) Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least Squares", *Quarterly of Applied Mathematics*, **2**, (1944), pp. 164-168.
- (38) McCord, Richard Kenneth, *Minimization With One Linear Equality Constraint and Bounds on the Variables*, Ph.D. Thesis, Stanford University, California, (1979).
- (39) McCormick, Garth P., *Nonlinear Programming*, New York, John Wiley and Sons, (1983).
- (40) Marquardt, D. W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters", *Journal of SIAM*, **11**, (1963), pp. 431-441.
- (41) Moore, R. H. and R. K. Zeigler, "The Solution of General Least Squares Problems with Special Reference to High-Speed Computers", Technical Report LA-2367, U. of California, Los Alamos, (1959).
- (42) Moré, Jorge J., "The Levenberg-Marquardt Algorithm: Implementation and Theory", in *Numerical Analysis*, (G. A. Watson, ed.), Lecture Notes in Mathematics, **630**, Berlin, Springer Verlag, (1978), pp. 105-116.

- (43) Nelson, David L. and Truman O. Lewis, "A Method for the Solution of Non-Linear Least Squares Problems When Some of the Parameters are Linear", *The Texas Journal of Science*, **xxi**, (1970), p. 480.
- (44) Ortega, J. M. and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, New York, Academic Press, (1970).
- (45) Osborne, M. R., "Some Aspects of Non-Linear Least Squares Calculations", in *Numerical Methods for Non-Linear Optimization*, (F. A. Lootsma, ed.), New York, Academic Press, (1972), pp. 171-180.
- (46) Osborne, M. R., "Some Special Nonlinear Least Squares Problems", *SIAM Journal on Numerical Analysis*, **12**, (1975), pp. 571-592.
- (47) Pavel-Parvu, M. and A. Korganoff, "Iteration Functions for Solving Polynomial Equations", in *Constructive Aspects of the Fundamental Theorem of Algebra*, (B. Dejon and P. Henrici, eds.), New York, John Wiley and Sons, (1969), pp. 225-280.
- (48) Penrose, R., "A Generalized Inverse for Matrices", *Proceedings of the Cambridge Philosophical Society*, **51**, (1955), pp. 406-413.
- (49) Pérez, A. and H. D. Scolnik, "Derivatives of Pseudoinverses and Constrained Nonlinear Regression Problems", unpublished manuscript.
- (50) Richards, F. S. G., "A Method of Maximum-Likelihood Estimation", *Journal of the Royal Statistical Society, Series B*, **23**, (1961), pp. 460-475.
- (51) Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or Least Value of a Function", *The Computer Journal*, **3**, (1960), pp. 175-184.
- (52) Ruhe, Axel and Per Åke Wedin, "Algorithms for Separable Nonlinear Least Squares Problems", *SIAM Review*, **22**, (1980), pp. 318-337.
- (53) Schittkowski, K., *Nonlinear Programming Codes*, Springer Verlag Lecture Notes in Economics and Mathematical Systems, **183**, Berlin, (1980).
- (54) Scolnik, Hugo D., "On the Solution of Non-Linear Least Squares Problems", in *Information Processing 71*, (C. V. Freiman, ed.), Amsterdam, North-Holland Publishing Co., (1972), pp. 1258-1265.

- (55) Tseng, Yu. Ya., "Sur les Solutions des Equations Opératrices Fonctionnelles Entre les Espaces Unitaires. Solutions Extrémales. Solutions Virtuelles.", *Comptes Rendu Des Seances De L'Académie Des Sciences, Paris*, 228, (1949), pp. 640-641.
 - (56) Walling, Donald, "Non-Linear Least Squares Curve Fitting When Some Parameters are Linear", *The Texas Journal of Science*, xx, (1968), pp. 119-124.
 - (57) Wedin, Per Åke, "On Pseudoinverses of Perturbed Matrices", Technical Report, Computer Science Department, Lund University, Sweden, (1969).
 - (58) Wedin, Per Åke, "On Surface Dependent Properties of Methods for Separable Non-Linear Least Squares Problems", ITM Arbetsrapport nr. 23, (July, 1974).
 - (59) Wold, H. and E. Lyttkens, "Nonlinear Iterative Partial Least Squares (NIPALS) Estimation Procedures", *Bulletin ISI*, 43, (1969), pp. 29-51.
 - (60) Wolfe, Philip, "Methods of Nonlinear Programming", in *Nonlinear Programming*, (J. Abadie. ed.), Amsterdam, North-Holland Publishing Co., (1967), pp. 97-131
-

