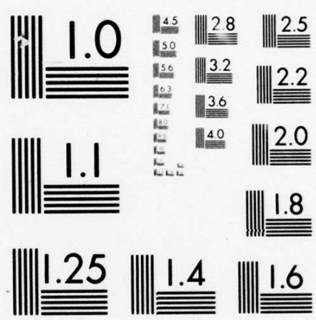
AD-A053 439 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/1
DECOMPOSITION IN FIXED POINT COMPUTATION. (U)
DEC 77 D SOLOW DAAG-29-74-C-0032
UNCLASSIFIED SOL-77-32 NL

1 OF 2
AD
A053439



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DECOMPOSITION IN FIXED POINT COMPUTATION

BY

DANIEL SOLOW

TECHNICAL REPORT SOL 77-32

DECEMBER 1977

Systems Optimization Laboratory

Department of Operations Research

Stanford University

Stanford California 94305

DECOMPOSITION IN FIXED POINT COMPUTATION

by

Daniel Solow

TECHNICAL REPORT SOL 77-32

December 1977

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH

Stanford University
Stanford, California

Research and reproduction of this report were partially supported by the Energy Research and Development Administration Contracts EY-76-S-03-0326 PA #52 and EY-76-S-03-0326 PA #18; the National Science Foundation Grants MCS76-20019 and MCS77-05623; the Army Research Office Contract DAAG-29-74-C-0032; the Office of Naval Research Contract N00014-75-C-0865; the Electric Power Research Institute Contract RP 652-1; and the Institute for Energy Studies at Stanford University.

ACKNOWLEDGMENTS

I would like to thank Professor Romesh Saigal for having introduced me to the area of fixed point computation and Professor B. Curtis Eaves for having served as my advisor throughout the dissertation. I am also indebted to the Systems Optimization Laboratory (and especially Professor George B. Dantzig) for supplying me with a computer budget and with the finest computing facilities I have ever used.

My deepest appreciation goes to my parents for their moral and financial support during my years of academic endeavors, and to my wonderful friend Andrée whose kind words of encouragement and unending patience made this work possible.

TABLE OF CONTENTS

TABLE OF CONTENTS Continued

CHAPTER 1

INTRODUCTION

1.1. Overview and Thesis Organization.

In the past decade, several constructive proofs of the Brouwer and Kakutani fixed point theorems have emerged. These proofs have been developed into algorithms (known in the literature as complementary pivot algorithms) which search for fixed points on unbounded regions. In turn these algorithms have been used to solve problems arising in economics, engineering and other branches of applied mathematics. An important application for which this method was awkward was that of optimizing an objective function subject to both equality and inequality constraints (hereafter referred to as the general constrained optimization problem). One result of the dissertation is the most efficient complementary pivot algorithm to date for handling this problem. The second major contribution of this thesis is a general structure on fixed point problems which, when present, enables one to work in a lower dimensional space. It is shown that the general constrained optimization problem may sometimes be formulated as a fixed point problem possessing this property.

The basic approach adopted in this work for handling the general constrained optimization problem is to use an implicit function (derived from the equality constraints) to solve for some dependent variables in terms of the remaining independent ones. Under certain circumstances, a fixed point algorithm may be used to search for optimal values of the independent variables while Newton's method for solving nonlinear systems of equations is used to determine values of the dependent variables. Theoretical conditions on the original functions are developed to guarantee that the fixed point algorithm converges to a solution and various techniques are devised to enhance the overall efficiency.

To help ascertain the value of this method, comparative computer tests are run against the Generalized Reduced Gradient (GRG) algorithm which is a well established nonlinear programming code. This method was selected as the basis for comparison because, to the author's knowledge, it is the best commercial code for solving the general constrained optimization problem (see Colville [8], Nishiyama, Simkin and Takeuchi [40], Lasdon, Warren, Jain and Ratner [34]). Seventeen test problems were taken from various sources. The fixed point code solved all seventeen and GRG solved sixteen. This supports the robustness of the fixed point approach. As to the computer times, the fixed point code proved to be as fast or faster than GRG on the lower dimensional problems. As the dimension increased, however, the trend reversed and on a forty dimensional problem GRG was approximately eleven times faster. The conclusion is that when the dimension of the original problem can be sufficiently reduced by the equality constraints, the fixed point approach appears to be more effective.

The dissertation consists of seven chapters. Chapter 2 contains the essence of the dimension reducing property along with several examples of where this structure arises in applications (of principal interest is the general constrained optimization problem). Under certain circumstances, the algorithm described in Chapter 3 can be used to obtain a solution to this problem. Some theoretical conditions on the original functions which ensures that the algorithm converges to a solution are established in Chapter 4 while Chapter 5 deals with all of the computational considerations. Chapter 6 proposes various techniques to improve the efficiency. Finally, Chapter 7 presents the results of the computer tests along with the appropriate conclusions.

1.2. Historical Development.

The history of the computation of fixed points dates back to 1929 when Knaster, Kuratowski and Mazurkiewicz [30] gave the first constructive proof of the Brouwer fixed point theorem using Sperner's lemma. It was not until thirty eight years later that Scarf [48], using the ideas of complementary pivot theory developed by Lemke [35] and Lemke and Howson [36] in 1964, produced the first algorithm to approximate fixed points of continuous functions from the simplex into itself. Cohen [7] simultaneously developed a constructive type proof of Sperner's lemma. Many ideas closely related to fixed point computation were anticipated by Hirsch [27] in 1963 wherein he gave an existence proof by using a certain constructive technique to reach a contradiction.

The combinatorial techniques of Scarf's algorithm have various other applications in mathematics and economics as shown in Scarf and Hansen [25]. One example of this is approximating solutions to convex minimization problems in which the feasible region is compact and non-empty. Another example is the special case of Kakutani's fixed point theorem [28] in which the compact convex subset involved is a simplex.

There were, however, several deficiencies with these methods. Theoretically, the general version of Kakutani's theorem could not be proved and computationally there was no way to "continuously" obtain more accurate approximations. Furthermore, in higher dimensions, these algorithms were highly inefficient.

In 1970, Eaves [11,12] developed an algorithm for computing a fixed point of the point to set map in Kakutani's theorem. Then in 1971, Eaves [13] and Eaves and Saigal [15] and, independently, Merrill [38] developed techniques for overcoming many of the computational difficulties. The computer results of these new ideas are reported in Merrill [38], Saigal, Solow and Woolsey [47], Gochet, Loute and Solow [23], Wilmuth [53] and subsequently elsewhere. Since 1973 various other researchers have contributed to the field including Todd [51,52], Kuhn [32,33], Garcia [21,22], Fisher and Gould [17], Gould and Tolle [24], Kojima [31], Engles [16], Friedenfelds [19], Eaves [14], Saigal [44,45,46] and Kellogg, Li and York [29]. Currently there are over one hundred papers relating to fixed point computation and complementary pivot theory, and an extensive bibliography may be found in Eaves [14].

1.3. Notation and Preliminaries

The notation is developed here and is used consistently throughout the thesis. To begin with let

$$R^m = \text{m-dimensional euclidean space} .$$

The following variables will always represent vectors in euclidean space: a, b, u, v, w, x, y, z, and the variables ℓ, m, n and s will denote their various dimensions. Any point $x \in R^m$ should always be thought of as a column vector. The row vector corresponding to x will be denoted by x^T. If $x \in R^m$ and $y \in R^n$ then let (x,y) be the $(m+n)$ column vector whose first m components are those of x and whose last n components are those of y. The ith component of a column vector x will be denoted by a subscript, e.g. x_i, where as superscripts refer to elements in a collection of vectors. Thus n different vectors in R^m might be represented by $x^1, \ldots, x^n$. Whenever possible the superscript k will be used for infinite collections and sequences.

Let $x, y \in R^n$ and $a \in R^1$ with $a > 0$ then

$\langle x,y \rangle = \sum_{i=1}^n x_i y_i$ = inner product,

$x < y$ means $x_i < y_i$ for each $i = 1,\ldots,n$,

$x \leq y$ means $x_i \leq y_i$ for each $i = 1,\ldots,n$,

$\|x\| = [\langle x,x \rangle]^{1/2}$ = euclidean norm,

$B(x,a)$ = open ball of radius a and center x

$= \{z \in R^n / \|x-z\| < a\}$.

Some special notation for matrices is also needed. All matrices will contain real numbers. The variables A, U, V and W will always stand for matrices and I is reserved for the square identity matrix. Its dimension will be implied by the context (as will the size of the vector 0). If U is an $(m \times n)$ matrix and V an $(m \times \ell)$ matrix then by $[U,V]$ is meant then $(m \times (n + \ell))$ matrix whose first n columns are those of U and whose last ℓ columns are those of V. In this context an n-vector may be thought of as an $(n \times 1)$ matrix. The only case in which confusion can possibly arise is when U and V are real numbers with $U < V$. In this case $[U,V]$ can also be the closed interval, i.e. $\{x \in R^1 / U \leq x \leq V\}$, depending on the context. The ith column of U will be denoted by $U_{.i}$ and the ith row of U by $U_{i.}$. Also it will sometimes be necessary to associate a column of a matrix, say A, with a vector x. This will be done by the notation $A_{.(x)}$. Note also that if $y \in R^n$ then Uy is well defined because y is a column vector. Also if W is a $(n \times n)$ matrix its determinant will be denoted by $\det(W)$. Finally define

$$\|U\| = \sup\{\|Uy\| / y \in R^n \text{ and } \|y\| = 1\} .$$

A great many different functions are required for (a) the property of decomposability in Chapter 2 and (b) the various constrained optimization problems of Chapter 4. For this reason, f, g, h, p, q, r, t, F, G, L, P and Q will always represent functions. Observe that if $h: R^m \to R^n$, then for each $x \in R^m$, $h(x)$ generates a column vector

$h(x) \in R^n$ whose ith components is $h_i(x)$. Furthermore, f, g, h, r and F will be related to decomposability whereas p, q, r, t, G, P and Q will be used for the constrained optimization problems. Point to set maps will be represented by H, S and T.

The handling of sequences also requires some discussion, for example, a sequence of vectors will be denoted by $\{x^k\}$ where it is understood that $k = 0,1,\ldots$ or $k = 1,2,\ldots$ as implied by the context. A subsequence will be thought of as a subset K of the positive integers and will be written "$\{x^k\}, k \in K$". If the given sequence converges to x it will be written "$\{x^k\} \to x$ for $k \to \infty$" and for a subsequence, "$\{x^k\} \to x$ for $k \in K$". x is said to be a cluster point of $\{x^k\}$ iff there is a subsequence K such that $\{x^k\} \to x$ for $k \in K$.

For sets, the variables X, Y, Z and O will always be used. If X is a subset of R^m then

$$\begin{aligned}
\text{bd}(X) &= \text{boundary of } X, \\
\text{cl}(X) &= \text{closure of } X, \\
\text{hull}(X) &= \text{convex hull of } X, \\
\text{int}(X) &= \text{interior of } X, \\
X^* &= \text{the set of all non-empty subsets of } X.
\end{aligned}$$

If X and Y are subsets of R^m and if $w \in R^m$ and $f:X \to Y$, $T:X \to Y^*$ then

$$X + Y = \{x + y / x \in X,\ y \in Y\} ,$$

$$X \times Y = \{(x,y) / x \in X,\ y \in Y\} ,$$

$$w + X = \{w\} + X ,$$

$$X \setminus Y = \{x \in X / x \notin Y\} ,$$

$$f(X) = \{f(x) / x \in X\} ,$$

$$T(X) = \bigcup_{x \in X} T(x) .$$

Theorems, lemmas, etc. are numbered sequentially within a chapter and the following conventions have been adopted:

D.i.j = j<u>th</u> definition of Chapter i ,

L.i.j = j<u>th</u> lemma of Chapter i ,

T.i.j = j<u>th</u> theorem of Chapter i ,

P.i.j = j<u>th</u> problem of Chapter i ,

C.i.j = j<u>th</u> corollary of Chapter i ,

A.i.j = j<u>th</u> assumption of Chapter i.

Some preliminary definitions and results will be drawn upon in later chapters. These notions are presented here.

<u>Definition 1.1.</u> Let O be an open subset of R^m. The function $h: O \to R^n$ is said to be <u>differentiable</u> at a point $x \in O$ iff there is an $(n \times m)$ matrix $Dh(x)$ (called the Jacobian matrix) such that

$$\lim_{\substack{\|z-x\| \to 0 \\ z \in O}} \|h(z) - h(x) - Dh(x)(z-x)\| = 0$$

h is said to be <u>differentiable on</u> O iff h is differentiable at each point of O. When h is a function from O into R^1, the $(1 \times n)$ matrix $Dh(x)$ is called the gradient of h at x and its transpose is written $\nabla h(x)$. The function h is said to be <u>twice differentiable</u> iff h is differentiable and if the mapping Dh is also differentiable. The second derivative of h at x is written $D^2h(x)$.

Suppose a function $Q: R^m \times R^n \to R^\ell$ is differentiable at a point $(a,b) \in R^m \times R^n$. The $(\ell \times (m+n))$ matrix $DQ(a,b)$ will frequently be written $[D_xQ(a,b), D_yQ(a,b)]$ where $D_xQ(a,b)$ is the $(\ell \times m)$ matrix corresponding to the derivative of Q with respect to the variables in R^m and $D_yQ(a,b)$ for the same in R^n. A similar idea is applied to gradients.

It will often be necessary to use the derivatives to obtain bounds. Ortega and Rhineboldt [41] provide several such theorems and they are stated for use in this thesis as:

<u>Theorem 1.1</u>. Let $f: X \to R^n$ be differentiable on the convex open set $X \subseteq R^m$. Then

$$\|f(z) - f(x)\| \leq \sup\{\|Df(x + \lambda(z-x))\| / \lambda \in [0,1]\} \; \|z-x\|$$

for all $z, x \in X$.

<u>Proof</u>. See Theorem 3.2.3 of Ortega and Rhineboldt [41]. $\|$

Theorem 1.2. Let $f:X \to R^n$ be twice continuously differentiable on the convex open set $X \subseteq R^m$. Then

$$\|f(z) - f(x) - Df(x)(z-x)\| \leq \sup\{\|D^2f(x + \lambda(z-x))\| / \lambda \in [0,1]\} \|y-x\|^2 .$$

Proof. See Theorem 3.3.6 of Ortega and Rhineboldt [41]. ||

If the functions involved are convex then a slightly weaker notion of differentiation exists. (It is assumed that the reader is familiar with terms such as "convex function," "convex set," etc. Rockafeller [43], Stoer and Witzgall [50] and Mangasarian [37] deal with most of these notions.)

Definition 1.2. Suppose O is a subset of R^m and $h:O \to R^n$ is convex. The $(n \times m)$ matrix W is a subgradient of h at $x \in O$ iff

$$h(z) \geq h(x) + W(z-x) \qquad \text{for all } z \in O .$$

The set of all subgradients of h at x is denoted by $\partial h(x)$.

There is a relation between D.1.1 and D.1.2 which is stated in

Theorem 1.3. Suppose O is an open subset of R^m. A convex function $h:O \to R^n$ is differentiable at $x \in O$ iff $\partial h(x) = \{Dh(x)\}$.

Proof. See Theorem 25.1 of Rockafeller [43]. ||

Corollary 1.1. If a convex function $h:R^m \to R^1$ is differentiable then

$$h(z) \geq h(x) + \langle \nabla h(x), z-x \rangle \qquad \text{for all } z \in R^m .$$

Proof. By T.1.3, $\nabla h(x) \in \partial h(x)$ and the result follows from D.1.2. ||

Some other concepts related to convexity are:

Definition 1.3. A convex function $p:R^m \to R^1 \cup \{\pm \infty\}$ is proper iff there is an $x \in R^m$ such that $p(x) > -\infty$.

Definition 1.4. The epigraph of a convex function $p:R^m \to R^1 \cup \{\pm \infty\}$ is $\{(x,z) \in R^m \times R^1/z \geq p(x)\}$ and is written $\text{epi}(p)$.

Definition 1.5. A convex function $p:R^m \to R^1 \cup \{\pm \infty\}$ is closed iff $\text{epi}(p)$ is a closed set.

Definition 1.6. The domain of finiteness of a convex function $p:R^m \to R^1 \cup \{\pm \infty\}$ is $\{x \in R^m/p(x) < \infty\}$ and is written $\text{dom}(p)$.

Definition 1.7. The vector $d \in R^m$ is said to be a common direction of recession of the convex function $h:R^m \to R^n$ iff there is a scalar b and a vector $x \in R^m$ such that

$$h_i(x + \lambda d) \leq b \quad \text{for all} \quad \lambda \geq 0 \quad \text{and} \quad i = 1,\ldots,n .$$

Point to set maps are required for transforming some optimization problems into fixed point problems. Only certain classes of point to set maps can be used in fixed point computation. Some of these concepts are now developed.

Let $S: R^m \to (R^m)^*$ be a point to set map.

Definition 1.8. S is said to be convex iff $S(x)$ is convex for each $x \in R^m$.

Definition 1.9. S is said to be upper semi-continuous (u.s.c.) iff whenever

(1) $\{x^k\} \to x$

(2) $y^k \in S(x^k)$ for each $k = 1,2,\ldots$

(3) $\{y^k\} \to y$

then $y \in S(x)$.

Definition 1.10. A point to set map S is usable iff S is non-empty convex and u.s.c.

Several results concerning point to set maps were proved by Merrill. They are stated as

Theorem 1.4. Let O be an open subset of R^m and suppose $t: O \to R^1 \cup \{\pm \infty\}$ Then the point to set map $T: \text{dom}(t) \to R^m$ defined by $T(x) = x - \partial t(x)$ is usable.

Proof. See Theorem 10.4 of Merrill [38]. ||

Theorem 1.5. Let X be a subset of R^m. Also let $S: cl(X) \rightarrow (R^m)^*$ and $T: R^m \setminus int(X) \rightarrow (R^m)^*$ be usable point to set maps. Then the point to set map $H: R^m \rightarrow (R^m)^*$ defined by

$$H(x) = \begin{cases} S(x) & \text{if } x \in int(X) \\ hull(S(x) \cup T(x)) & \text{if } x \in bd(X) \\ T(x) & \text{if } x \notin cl(X) \end{cases}$$

is also usable.

Proof. See Theorem 2.6 of Merrill [38]. ||

A final definition which is needed is that of isotonicity.

Definition 1.8. A function $h: R^m \rightarrow R^n$ is isotone iff $h(x) \leq h(z)$ whenever $x, z \in R^m$ and $x \leq z$.

CHAPTER 2

DECOMPOSABILITY IN FIXED POINT PROBLEMS

2.1. General Theory and Examples for Functions.

Given a function F mapping a nonempty subset Z of R^s into itself, the fixed point problem is that of finding a $z \in Z$ with $F(z) = z$. The basic idea behind decomposability is to place a structure on F such that a fixed point may be computed by working in a lower dimensional space. Several examples of where this structure appears in applications is also presented. This structure is described by

Definition 2.1. Let Z be a nonempty subset of R^s and let $F: Z \to Z$. The pair (F,Z) is said to be decomposable iff there are positive integers m and n whose sum is s, nonempty subsets X of R^m and Y of R^n whose cross product is Z together with functions $f: Z \to X$, $g: Z \to Y$ and $h: X \to Y$ such that for each $x \in X$,

(1) $F(x,h(x)) = (f(x,h(x)),g(x,h(x)))$.

(2) If $x = f(x,h(x))$ then $h(x) = g(x,h(x))$.

The first condition states that F may be decomposed into two separate functions f and g with f providing the first m coordinates of F and g providing the remaining n coordinates. The second condition is a special relation between the functions f, g and h which will be used to establish the connection between fixed points of the lower dimensional problem and fixed points of F. The lower dimensional problem will be one of finding a fixed point in R^m. More specifically, defining the function $r:X \to X$ by $r(x) = f(x,h(x))$ the next theorem shows that any fixed point of r yields a fixed point of F. This then is

Theorem 2.1. If (F,Z) is decomposable then $x \in X$ satisfies $x = f(x,h(x))$ iff $(x,h(x)) = F(x,h(x))$ where X and f are obtained from D.2.1.

Proof. Suppose first that $x = f(x,h(x))$. By property (2) of D.2.1, $h(x) = g(x,h(x))$. Thus $(x,h(x)) = (f(x,h(x)),g(x,h(x))) = F(x,h(x))$, the last equality being justified by property (1) of D.2.1. This takes care of the necessary part of the theorem. For the sufficiency part suppose $(x,h(x)) = F(x,h(x))$. From property (1) of D.2.1 it follows immediately that $x = f(x,h(x))$ as desired. ||

Replacing $f(x,h(x))$ by $r(x)$ one may more easily see what T.2.1 is saying. It is saying that if (F,Z) is decomposable then finding a fixed point x of r yields a fixed point of F namely $(x,h(x))$. The importance of this is that finding a fixed point of r involves working in R^m instead of R^s. Some conditions under which F and r may be expected to have fixed points is developed in

Corollary 2.1. Suppose (F,Z) is decomposable and that f, g, h, X and Y are obtained from D.2.1. In addition suppose that f and h are continuous and that X is compact and convex. Under these conditions F has a fixed point.

Proof. Let $r:X \to X$ by $r(x) = f(x,h(x))$ for each $x \in X$. This function is continuous since f and h are continuous and since the composition of continuous functions is continuous. Now apply the Brouwer fixed point theorem to r to obtain the existence of an $x \in X$ such that $r(x) = x$. By T.2.1, $(x,h(x))$ is a fixed point of F. ||

The following examples show the value of this rather straightforward observation. The first example, although hypothetical, shows the potential of decomposability by reducing an (n+1)-dimensional fixed point problem to a 1-dimensional problem. The second example is equality constrained optimization. The third example is a partially linear systems of equations and the last is the Bilinear Complementarity Problem proposed in Wilson [54].

Example 2.1. Let $a, b \in R^1$ with $a < b$. Set $X = [a,b]$ and $Y = R^n$. Thus $Z = X \times Y = [a,b] \times R^n$. F will be constructed to satisfy the hypotheses of C.2.1 in such a way that the function r will be a mapping from $[a,b]$ into $[a,b]$. Thus finding a fixed point of F will be reduced to finding a fixed point of r and that will be a 1-dimensional problem. To actually construct this F let $f:Z \to X$ and $h:X \to Y$ be arbitrary but continuous functions. Define $g:Z \to Y$

by $g(z) = h(f(z))$ for all $z \in Z$. Finally let $F(z) = (f(z), g(z))$ for all $z \in Z$.

Proposition 2.1. (F,Z) is decomposable.

Proof. Condition (1) of D.2.1 is true by construction of F so only condition (2) needs to be verified. To this end let $x \in X$ with $x = f(x,h(x))$. Applying h to both sides of this yields $h(x) = h(f(x,h(x))) = g(x,h(x))$ as desired. ||

Proposition 2.2. F has a fixed point.

Proof. (F,Z) is decomposable and satisfies the hypotheses of C.2.1, thus F has a fixed point. ||

From the proof of C.2.1 it is apparent that in order to compute a fixed point of F one need only be computed for $r: X \to X$, and finding a fixed point of r is one of searching $[a,b]$ as opposed to searching $[a,b] \times R^n$.

Notice that f and h were completely arbitrary except for their continuity, thus they may be made nonseparable, nondifferentiable, etc. Also this example shows that there are problems (F,Z) whose fixed points would not normally be computable because of the large dimensionality yet if (F,Z) is decomposable in the proper way one can find the fixed point in a 1-dimensional space.

Example 2.2. In this example, a nonlinear programming problem (NLP) of the form

(P.2.1)
$$\min \; P(z)$$
$$\text{s.t.} \; G(z) = 0$$
$$z \in R^s$$

where $P: R^s \to R^1$ and $G: R^s \to R^n$ is put into the framework of decomposability. Since this was the problem which motivated the concept of decomposability all of Chapter 4 has been devoted to a complete theoretical analysis of this problem. In this section a connection between the NLP and decomposability is established in loose terms and a rigorous approach is presented in Chapter 4.

In order to show that P.2.1 is in fact a special case of decomposability one must (a) find a fixed point problem which is related to P.2.1 and (b) show that this fixed point problem is decomposable.

Eaves [11] and Merrill [38] have discussed extensively the fixed point formulation when the NLP is in the form of either (i) unconstrained optimization or (ii) inequality constraints with the existence of a point at which all constraints are strictly feasible. Clearly P.2.1 does not fall into either of these categories (even when the constraints $G(z) = 0$ are replaced by $G(z) \leq 0$ and $-G(z) \leq 0$). So the question becomes how to transform an equality constrained problem into an optimization problem of type (i) or (ii) as described above. Three possible methods for doing this are now presented, the last of which led to the concept of decomposability.

Method 1. Constraint relaxation.

In this approach the equality constrained problem is replaced by a sequence of problems of type (ii). This is done by introducing a tolerance for the constraints. Formally let $\{a^k\}$ be a sequence of vectors in R^n with $a^k > 0$ for each $k = 1,2,\ldots$ and with $\{a^k\} \to 0$ for $k \to \infty$. Consider the family of optimization problems.

$$\begin{aligned} &\min && P(z) \\ &\text{s.t.} && -a^k \leq G(z) \leq a^k \qquad \text{for } k = 1,2,\ldots \\ &&& z \in R^s \end{aligned}$$

Under certain circumstances each of these problems is of type (ii) and one could then use existing methods to partially solve the kth problem. Under additional circumstances one might expect the limit of the solutions (assuming such a limit exists) to be a solution to P.2.1. Computational results from this approach reported by Merrill [38] were extremely discouraging and so this approach was discarded.

Method 2. Lagrangian approach.

In this method one defines the Lagrangian function $L: R^s \times R^n \to R^1$ by $L(z,u) = P(z) + \langle u, G(z) \rangle$. One can then show (see Mangasarian [37] for example) that a necessary condition for (z,u) to solve P.2.1 is

$$\nabla_z L(z,u) = \nabla P(z) + u^T DG(z) = 0$$

$$\nabla_u L(z,u) = G(z) = 0$$

assuming of course that P and G are differentiable functions on $R^s \times R^n$. This is a system of $(n+s)$ equations in $(n+s)$ unknowns which may be transformed into a fixed point problem by defining $F: R^s \times R^n \to R^s \times R^n$ by $F(z,u) = (z,u) - (\nabla_z L(z,u), \nabla_u L(z,u))$. This fixed point formulation has increased the dimension of the original problem from s to $(s+n)$. This is exactly contrary to the concept of decomposability and so this approach was also discarded for this work.

Method 3. Decomposability.

This approach is motivated by considering the special form of P.2.1 in which the equality constraints are linear, for then these equations can be used to solve for some of the variables (called the basic variables) in terms of the remaining variables (called the nonbasic variables). If the rank of the linear transformation is n then there will be exactly n basic variables (referred to herein after by the vector $y = (y_1, \ldots, y_n)$). Correspondingly there will be $m = s-n$ non-basic variables (referred to by the vector $x = (x_1, \ldots, x_m)$). These observations are formalized in

Proposition 2.3. Let W be an $(n \times s)$ matrix and w an n-vector such that $G(z) = Wz + w$ for each $z \in R^s$. If $\mathrm{rank}(W) = n$ then there is a function $h: R^m \to R^n$ such that $G(x,h(x)) = 0$ for each $x \in R^m$.

Proof. Since $\mathrm{rank}(W) = n$ there are $(n \times m)$ and $(n \times n)$ matrices U and V respectively such that after permuting the columns of W, (1) $W = [U,V]$ and (2) V is nonsingular. Define $h: R^m \to R^n$ by $h(x) = -V^{-1}(w + Ux)$. To verify that $G(x,h(x)) = 0$ note that

$$
\begin{aligned}
G(x,h(x)) &= W(x,h(x)) + w \\
&= [U,V]\ (x,h(x)) + w \\
&= Ux + Vh(x) + w \\
&= Ux - (w + Ux) + w \\
&= 0 \ . \qquad \qquad \|
\end{aligned}
$$

In the case that the constraints of P.2.1 are nonlinear, conditions may be placed on the function G which ensures the conclusion of Proposition 2.3; namely, that there is an $h: R^m \to R^n$ such that $G(x,h(x)) = 0$ for each $x \in R^m$. In this case the function h is an implicit function and conditions for its existence are established in Chapter 4. From here on, the function h will always be the mapping described above. Furthermore, this function will be used in showing P.2.1 is a special case of decomposability.

Once the existence of this function h has been established one may then eliminate the basic variables (along with equality constraints) from P.2.1. The result is an unconstrained optimization problem which takes the form

(P.2.2) $$\min \quad p(x) = P(x,h(x)) \;.$$
$$\text{s.t.} \quad x \in R^m \quad .$$

Since this is a minimization problem of type (i), there is a fixed point formulation of P.2.2, namely that of finding an $x \in R^m$ such that $x - \nabla p(x) = x$ provided p is differentiable (see Merrill [38]). These remarks are now summarized and stated in

Proposition 2.4. Let $Z = R^s = R^m \times R^n$ and suppose there is an $h: R^m \to R^n$ such that $G(x,h(x)) = 0$ for each $x \in R^m$. Also suppose P, G and h are differentiable on their respective domains. Let $F: Z \to Z$ be defined by $F(x,y) = (x,y) - (\nabla_x P(x,y) + \nabla_y P(x,y)^T Dh(x),\ G(x,y))$. Then (F,Z) is decomposable.

Proof. Let $X = R^m$, $Y = R^n$ and define $f: Z \to X$ by $f(x,y) = x - \nabla_x P(x,y) - \nabla_y P(x,y)^T Dh(x)$ and define $g: Z \to Y$ by $g(x,y) = y + G(x,y)$ for each $(x,y) \in Z$. Now one may verify the conditions of D.2.1 with X, Y, f, g and h. ||

Example 2.3. Partially linear systems of equations. In this example, decomposability will be applied to solving an s by s system of equations in which some of them are linear. Let W be an $(n \times s)$ matrix and w be an n-vector. Also let $E: R^m \times R^n \to R^m$ be an arbitrary function. The original problem may be stated as that of finding a $z \in R^s$ such that

$$E(z) = 0 \;,$$
$$Wz + w = 0 \;.$$

The next proposition gives a condition under which there is a decomposable fixed point problem whose solution solves the original system of equations.

Proposition 2.5. Let $Z = R^s$. If $\text{rank}(W) = n$ then there is a function $F: Z \to Z$ such that (F,Z) is decomposable.

Proof. In order to prove this, the appropriate sets and functions will be created. In particular let $X = R^m$ and $Y = R^n$. Since $\text{rank}(W) = n$ there are $(n \times m)$ and $(n \times n)$ matrices U and V respectively such that after rearranging the columns of W, (i) $W = [U,V]$, and (ii) V is nonsingular. Define $f:Z \to X$, $g:Z \to Y$ and $h:X \to Y$ by $f(x,y) = E(x,y) + x$, $g(x,y) = Ux + (V + I)y + w$, $h(x) = -V^{-1}(Ux + w)$ for each $(x,y) \in Z$. Now D.2.1 may be verified. ||

Example 2.4. The Bilinear Complementarity Problem. This problem arises in economics and was introduced in Wilson [54] and may be stated as that of finding $x, y \in R^n$ such that (a) $x, y \geq 0$, (b) $x = Uy + u$ and (c) $x_i y_i = \langle V_{.i}, y\rangle$ for each $i = 1,\ldots,n$ where U and V are $(n \times n)$ matrices and u is an n-vector. Through the rest of this example it is assumed that

(i) V is nonsingular.

(ii) $Z^i = \{z \in R^n / z_i > 0, \langle V_{.i}, z\rangle \geq 0\}$ is nonempty for each $i = 1,\ldots,n$.

(iii) There is no $z \in R^n$ with $z \geq 0$ and $V^T z > 0$.

Let $z' \in R^n$ be such that $V^T z' > 0$. Now define $f: R^n \times R^n \to R^n$ by $f_i(x,y) = x_i y_i + x_i - \langle V_{.i}, y\rangle$; $g: R^n \times R^n \to R^n$ by $g(x,y) = (V^T)^{-1}(x_1 y_1, \ldots, x_n y_n)$ and $h: R^n \to R^n$ by

$$h(x) = \begin{cases} z \in Z^i & \text{if } i \text{ is the first integer such that } x_i < 0 \\ z' & \text{if } x \geq 0 \text{ and } \{y \geq 0 / x = Uy + u\} \text{ is empty.} \\ y & \text{if } x,\ y \geq 0,\ x = Uy + u \ . \end{cases}$$

As usual define $F: R^n \times R^n \to R^n \times R^n$ by

$$F(x,y) = (f(x,y),\ g(x,y)) \quad \text{for all} \quad (x,y) \in R^n \times R^n \ .$$

Proposition 2.6. If $Z = R^n$ then (F,Z) is decomposable.

Proof. Let $X = Y = R^n$. Then these sets together with the functions f, g and h will be shown to satisfy D.2.1. Only condition (2) of D.2.1 needs verification so let $x \in R^n$ with $f(x,h(x)) = x$. Then by construction,

$$(x_1 h_1(x) + x_1 - \langle V_{.1}, h(x)\rangle,\ \ldots\ ,\ x_n h_n(x) + x_n - \langle V_{.n}, h(x)\rangle) = (x_1, \ldots, x_n)$$

and hence

$$x_1 h_1(x) = \langle V_{.1}, h(x)\rangle,\ \ldots\ ,\ x_n h_n(x) = \langle V_{.n}, h(x)\rangle$$

so

$$(x_1 h_1(x),\ \ldots\ ,\ x_n h_n(x)) = V^T h(x) \ .$$

Multiplying both sides of this by $(V^T)^{-1}$ yields

$$h(x) = (V^T)^{-1}\,(x_1h_1(x),\ \dots\ ,\ x_nh_n(x)) = g(x,h(x))\ . \qquad \|$$

Proposition 2.7. If $f(x,h(x)) = x$ then $x \geq 0$.

Proof. Suppose not. Then there is a j between 1 and n such that $x_j < 0$. Let i be the first subscript with $x_i < 0$. By the definition of h, $h(x) = z$ where $\langle V_{.i}, z\rangle \geq 0$ and $z_i > 0$. From the fact that $f(x,h(x)) = x$ it follows that $x_ih_i(x) = \langle V_{.i}, h(x)\rangle$ but the left side is strictly negative and the right side is greater than or equal to 0. This contradiction shows the claim. $\|$

Proposition 2.8. If $f(x,h(x)) = x$ then $h(x) \geq 0$.

Proof. Suppose not. Then there is an i between 1 and n such that $h_i(x) < 0$. From Proposition 2.7 it may be assumed that $x \geq 0$ and consequently that $h(x) = z'$ where $V^Tz' > 0$. By the fact that $f(x,h(x)) = x$ it follows that $x_ih_i(x) = \langle V_{.i}, h(x)\rangle$ but the right side is strictly positive and the left side is less than or equal to 0. This contradiction shows the claim. $\|$

Proposition 2.9. If $(x,h(x)) = F(x,h(x))$ then $(x,h(x))$ solves the BLCP.

Proof. From Propositions 2.7 and 2.8 both x and $h(x)$ are nonnegative. Using the fact that $f(x,h(x)) = x$ one may conclude that $x_i h_i(x) = \langle V_{i.}, h(x)\rangle$ for each $i = 1,\ldots,n$. Finally since $h(x) \geq 0$ and by assumption (iii) it follows that $x = Uh(x) + u$. Thus in fact $(x,h(x))$ solves the BLCP. ||

In this section the concept of a decomposable fixed point problem was discussed and several examples of this property were presented. The most interesting example was optimization with equality constraints. One would like to be able to handle both equality and inequality constrained problems, however, in order to do this it appears necessary to enter the framework of point to set maps. The next section is devoted to this extension.

2.2. General Theory and Examples for Point to Set Maps.

Given a nonempty subset Z of R^s and a point to set map $S:Z \to Z^*$ the fixed point problem is that of finding a $z \in Z$ such that $z \in S(z)$. The basic idea is to put some structure on S which allows one to solve an equivalent problem but in a lower dimensional setting. This property is a straightforward generalization of the one described in the previous section. Corresponding to D.2.1 is

Definition 2.2. Let Z be a nonempty subset of R^s and let $S:Z \to Z^*$ be a nonempty point to set map. The pair (S,Z) is said to be decomposable iff there are positive integers m and n whose sum is s, nonempty subsets X of R^m and Y of R^n whose cross product is Z together with nonempty point to set maps $S_f: Z \to X^*$, $S_g: Z \to Y^*$ and a function $h: X \to Y$ such that for each $x \in X$,

(1) $S(x,h(x)) = S_f(x,h(x)) \times S_g(x,h(x))$.

(2) If $x \in S_f(x,h(x))$ then $h(x) \in S_g(x,h(x))$.

The first condition states that at each $x \in X$ the set $S(x,h(x))$ may be expressed as the cross product of the sets $S_f(x,h(x))$ and $S_g(x,h(x))$. The second condition is a special relation between S_f, S_g and h which will be used to establish the connection between fixed points of the lower dimensional problem and fixed points of S.

One would expect that D.2.2 reduces to D.2.1 in the special case where $S(z)$ is a set consisting of a single point for each $z \in Z$. This is established in

Proposition 2.10. Let $F: Z \to Z$ and define the point to set map $S: Z \to Z^*$ by $S(z) = \{F(z)\}$ for each $z \in Z$. Then (F,Z) is a decomposable function iff (F,Z) is a decomposable point to set map.

Proof. Assume first that (F,Z) is a decomposable function and let X, Y, f, g and h be obtained from D.2.1. Define the point to set maps $S_f: Z \to X^*$ and $S_g: Z \to Y^*$ by $S_f(z) = \{f(z)\}$ and $S_g(z) = \{g(z)\}$ for each $z \in Z$. It is now an easy matter to verify that X, Y, S_f, S_g

and h satisfy D.2.2. This takes care of the necessary part of the proposition. To go the other way suppose (S,Z) is a decomposable point to set map. Let X, Y, S_f, S_g and h be obtained from D.2.2. To show that (F,Z) is decomposable, functions f and g will be constructed in such a way that together with X, Y and h they will satisfy D.2.1. Simply define $f(z)$ to be any element of $S_f(z)$ and similarly for $g(z)$. To verify condition (1) of D.2.1 let $x \in X$. Then from the decomposability of S,

$$
\begin{aligned}
F(x,h(x)) \in \{F(x,h(x))\} &= S(x,h(x)) = S_f(x,h(x)) \times S_g(x,h(x)) \\
&= \{(f(x,h(x)),\ g(x,h(x)))\} ,
\end{aligned}
$$

the last equality being justified because $S_f(x,h(x))$ and $S_g(x,h(x))$ are sets with only one point. Hence $F(x,h(x)) = (f(x,h(x)),\ g(x,h(x)))$. Condition (2) is trivial to verify. ||

This result has shown that, on the surface, D.2.2 is the proper generalization of D.2.1. The next theorem is the analog to T.2.1.

Theorem 2.2. If (S,Z) is decomposable then $x \in X$ satisfies $x \in S_f(x,h(x))$ iff $(x,h(x)) \in S(x,h(x))$ where X and S_f are obtained from D.1.2.

Proof. Suppose first that $x \in S_f(x,h(x))$. By property (2) of D.2.2, $h(x) \in S_g(x,h(x))$. Thus $(x,h(x)) \in S_f(x,h(x)) \times S_g(x,h(x)) = S(x,h(x))$, the last equality being justified by property (1) of D.2.2. This takes care of the necessary part of the theorem. For the sufficiency part, suppose $(x,h(x)) \in S(x,h(x))$. From property (1) of D.2.2 it follows immediately that $x \in S_f(x,h(x))$ as desired. ||

Defining the point to set map $S_r:X \to X^*$ by $S_r(x) = S_f(x,h(x))$ one may more easily see what T.2.2 is saying. It is saying that if (S,Z) is decomposable then finding a fixed point x of S_r yields a fixed point of S namely $(x,h(x))$. The importance of this is that finding a fixed point of S_r involves working in R^m instead of R^s. The next task is to develop some conditions under which S and S_r may be expected to have fixed points.

Corollary 2.2. Suppose (S,Z) is decomposable and that in addition S_f is usable (i.e. nonempty convex and upper semi-continuous). Suppose also that h is continuous. If X is compact and convex with $\text{int}(X)$ nonempty then S has a fixed point.

Proof. Define $S_r:X \to X^*$ by $S_r(x) = S_f(x,h(x))$. S_r is a usable point to set map since the composition of usable maps is usable (see Theorem 1', p. 113 of Berge [3]). Now apply the Kakutani fixed point theorem [28] to S_r to obtain the existence of an $x \in X$ such that $x \in S_r(x)$. By T.2.2, $(x,h(x))$ is a fixed point of S. ||

Two examples of the property of decomposability are discussed. The first example, although hypothetical, shows the power of decomposability by reducing an (n+1)-dimensional fixed point problem to a 1-dimensional problem. The second example is optimization under both equality and inequality constraints.

Example 2.5. Let $a, b \in R^1$ with $a < b$. Set $X = [a,b]$ and $Y = R^n$. Thus $Z = X \times Y = [a,b] \times R^n$. S will be constructed to satisfy the hypotheses of C.2.2 in such a way that the point to set map S_r will be a mapping from $[a,b]$ into $[a,b]^*$. Thus finding a fixed point of S will be reduced to finding a fixed point of S_r, and that will be a one-dimensional problem. To actually construct this S let $h:X \to Y$ be an arbitrary continuous function and let $S_f:Z \to X^*$ be any usable point to set map. Define $S_g:Z \to Y^*$ by $S_g(x,y) = h(S_f(x,y))$. Finally define $S:Z \to Z^*$ by $S(x,y) = S_f(x,y) \times S_g(x,y)$ for each $(x,y) \in Z$.

Proposition 2.11. (S,Z) is decomposable.

Proof. Condition (1) of D.2.2 is true by construction so only condition (2) needs to be verified. To this end let $x \in X$ with $x \in S_f(x,h(x))$. Applying h to both sides yields $h(x) \in h(S_f(x,h(x))) = S_g(x,h(x))$ as desired. ||

Proposition 2.12. S has a fixed point.

Proof. (S,Z) is decomposable and satisfies the hypotheses of C.2.2, thus S has a fixed point. ||

From the proof of C.2.2 it is apparent that in order to compute a fixed point of S, one need only compute a fixed point of $S_r: X \to X^*$ and this is a problem of searching $[a,b]$ as opposed to $[a,b] \times R^n$. Notice that S_f and h were completely arbitrary except for their continuity properties. This example shows that there are problems (S,Z) whose fixed points would not normally be computable because of the high dimensionality yet if (S,Z) is decomposable in the proper way, one can find the fixed point in a 1-dimensional space.

Example 2.6. With the concept of a decomposable point to set map it is possible to show that, under certain circumstances, the general nonlinear programming problem of the form

$$\begin{array}{llll} \min & P(z) & & P: R^s \to R^1 \\ \text{s.t.} & G(z) = 0 & \text{where} & G: R^s \to R^n \\ & Q(z) \leq 0 & & Q: R^s \to R^\ell \\ & z \in R^s & & \end{array}$$

may be set up as a decomposable point to set map fixed point problem. In order to do this, many of the concepts developed in Chapter 4 are required and in order to avoid duplication, a proof of exactly how this can be done is postponed until Appendix A. Recall however, that decomposability is the property of being able to solve a particular fixed point problem by solving a lower dimensional fixed point problem, so in order to apply this concept to the nonlinear programming problem it will be necessary to (a) find a fixed point problem which is related to the NLP and (b) show that this fixed point problem is decomposable. Chapter 4 deals with (a) and Appendix A deals with (b). The approach is very much related to the one developed in Example 2.2.

In this section the concept of a decomposable point to set map was discussed and several examples were presented. The next section is a further generalization of these notions.

2.3. Generalizations.

A generalization of decomposability for functions and then for point to set maps is developed.

Definition 2.3. Let Z be a nonempty subset of R^s and let $F:Z \to Z$. The pair (F,Z) is weakly decomposable iff there are positive integers m and n whose sum is s, nonempty subsets X of R^m and Y of R^n together with functions $f:Z \to X$, $g:Z \to Y$, $h:X \to Y$ and $c:X \times Y \to Z$ such that for each $x \in X$,

(1) $F(c(x,h(x))) = c(f(c(x,h(x))),g(c(x,h(x))))$.

(2) If $x = f(c(x,h(x)))$ then $h(x) = g(c(x,h(x)))$.

(3) c is 1-1.

Note immediately that when c is the identity map on $X \times Y$ this definition is exactly that of decomposability of (F,Z). Whenever (F,Z) is weakly decomposable, X, Y, f, g, h and c will refer to the corresponding sets and functions derived from D.2.3.

As one would expect there are straightforward generalizations of T.2.1 and C.2.1 which may be stated as

Theorem 2.3. If (F,Z) is weakly decomposable then $x \in X$ satisfies $x = f(c(x,h(x)))$ iff $c(x,h(x)) = F(c(x,h(x)))$.

Proof. Suppose first that $x = f(c(x,h(x)))$. By condition (2) of D.2.3, $h(x) = g(c(x,h(x)))$. Thus

$$c(x,h(x)) = c(f(c(x,h(x))), g(c(x,h(x)))) = F(c(x,h(x))) ,$$

the last inequality being justified by property (1) of D.2.3. This takes care of the necessary part of the theorem. For the sufficiency part, suppose $c(x,h(x)) = F(c(x,h(x)))$. From property (2) of D.2.3, $F(c(x,h(x))) = c(f(c(x,h(x))), g(c(x,h(x))))$ and since c is 1-1 it follows that $x = f(c(x,h(x)))$. ||

Corollary 2.3. Suppose (F,Z) is weakly decomposable. Suppose in addition that f, h and c are continuous and that X is compact and convex. Then F has a fixed point.

Proof. Define $r:X \to X$ by $r(x) = f(c(x,h(x)))$ for each $x \in X$. Note that r is continuous because f, c and h are and because the composition of continuous functions is continuous. Now apply the Brouwer fixed point theorem to r to obtain the existence of an $x \in X$ such that $r(x) = x$. By T.2.3, $c(x,h(x))$ is a fixed point of F. ||

A similar concept exists for the point to set map case.

Definition 2.4. Let Z be a nonempty subset of R^s and let $S:Z \to Z^*$ be a point to set map. The pair (S,Z) is weakly decomposable iff there are positive integers m and n whose sum is s, nonempty subsets X of R^m and Y of R^n, functions $h:X \to Y$ and $c:X \times Y \to Z$ together with point to set maps $S_f:Z \to X^*$ and $S_g:Z \to Y^*$ such that for each $x \in X$,

(1) $S(c(x,h(x))) = c(S_f(c(x,h(x)) \times S_g(c(x,h(x))))$.

(2) If $x \in S_f(c(x,h(x)))$ then $h(x) \in S_g(c(x,h(x)))$.

(3) c is 1-1.

Note that when c is the identity on $X \times Y$ this definition is exactly that of decomposability for (S,Z). Whenever (S,Z) is weakly decomposable, X, Y, S_f, S_g, h and c will refer to the corresponding sets, point to set maps, and functions of D.2.4.

As one would expect there are straightforward generalizations of Proposition 2.10, T.2.3 and C.2.3. They are stated as

Proposition 2.13. Let $F:Z \to Z$ and define the point to set map $S:Z \to Z^*$ by $S(z) = \{F(z)\}$ for each $z \in Z$. Then (F,Z) is a weakly decomposable function iff (S,Z) is a weakly decomposable point to set map.

Proof. Assume first that (F,Z) is a weakly decomposable function and let X, Y, f, g, h and c be the sets and functions obtained from D.2.3. Define the point to set maps $S_f:Z \to X^*$ and $S_g:Z \to Y^*$ by $S_f(z) = \{f(z)\}$ and $S_g(z) = \{g(z)\}$ for each $z \in Z$. It is now an easy matter to show that X, Y, S_f, S_g, h and c satisfy D.2.4. This takes care of the necessary part of the theorem. To go the other way suppose (S,Z) is a weakly decomposable point to set map. Let X, Y, h, c, S_f and S_g be the sets, functions and point to set maps obtained from D.2.4. To show that (F,Z) is a weakly decomposable function, f and g will be constructed in such a way that together with h, c, X and Y they satisfy D.2.3. Simply define $f(z)$ to be any element of $S_f(z)$ and similarly for $g(z)$. To verify condition (1) of D.2.3 let $x \in X$. Then

$$\begin{aligned} F(c(x,h(x))) &\in \{F(c(x,h(x)))\} \\ &= S(c(x,h(x))) \\ &= c(S_f(c(x,h(x))) \times S_g(c(x,h(x)))) \\ &= c(\{f(c(x,h(x)))\} \times \{g(c(x,h(x)))\} \\ &\quad \in \{c(f(c(x,h(x))),\ g(c(x,h(x))))\} \end{aligned}$$

so

$$F(c(x,h(x))) = (f(c(x,h(x))),\ g(c(x,h(x))))$$

as desired. Condition (2) of D.2.3 is trivial to verify and this completes the proof. ||

Theorem 2.4. If (S,Z) is weakly decomposable then $x \in X$ satisfies $x \in S_f(c(x,h(x)))$ iff $c(x,h(x)) \in S(c(x,h(x)))$.

Proof. Suppose first that $x \in S_f(c(x,h(x)))$. By property (2) of D.2.4, $h(x) \in S_g(c(x,h(x)))$. Thus

$$c(x,h(x)) \in c(S_f(c(x,h(x))) \times S_g(c(x,h(x)))) = S(c(x,h(x))) ,$$

the last equality being justified by condition (1) of D.2.4. This takes care of the necessary part of the theorem. For the sufficiency part suppose $c(x,h(x)) \in S(c(x,h(x)))$. From condition (1) of D.2.4 and the fact that c is 1-1 it follows that $x \in S(c(x,h(x)))$ as desired. ||

Corollary 2.4. Suppose (S,Z) is weakly decomposable. Suppose in addition that S_f is a usable point to set map and that h and c are continuous. Suppose also that X is compact and convex with $\text{int}(X)$ nonempty. Under these conditions S has a fixed point.

Proof. Define $S_r : X \to X^*$ by $S_r(x) = S_f(c(x,h(x)))$. S_r is a usable point to set map since the composition of usable maps is usable (see Theorem 1', p. 113 of Berge [3]). Now apply the Kakutani fixed point theorem [28] to S_r to obtain the existence of an $x \in X$ with $x \in S_r(x)$. By T.2.4, $c(x,h(x))$ is a fixed point of S. ||

This chapter has dealt with the structure of decomposability as it applies to functions and point to set maps. Several examples of this were pointed out, the most interesting of which is optimization with both equality and inequality constraints. The next step is to find a method for solving these problems. Chapter 3 develops an algorithm which may sometimes be used to solve the fixed point problem and Chapter 4 shows how to convert the optimization problem, into a fixed point problem.

CHAPTER 3

AN ALGORITHM FOR COMPUTING FIXED POINTS

3.1. Triangulations.

In this chapter an algorithm which attempts to compute fixed points of certain point to set maps is described. It was developed by Merrill [38] in 1971. A slightly more general version was independently and simultaneously developed by Eaves and Saigal [15]. In very general terms the algorithm attempts to compute the fixed points of a sequence of piecewise linear (PL) functions approaching the original function. Under certain hypotheses each of these points may be computed in a finite number of steps. Under additional circumstances there will be a cluster point of the sequence and that will be the desired fixed point. To use this approach it is necessary to have (a) a computerized method for generating the sequence of PL approximations, (b) a systematic procedure for attempting to move through these pieces of linearity in search of a fixed point and (c) some conditions under which one might expect the method to find such points in a finite number of steps. These are the three sections of this chapter.

Let M be a nonempty subset of R^m and $S:M \rightarrow (R^m)^*$ be a usable point to set map whose fixed point is sought. In order to construct a PL approximation f^L to S it will be necessary to break up M into pieces (called simplexes) which fit together in a very special way and on which f^L will actually be linear and continuous. This partition of M is called a triangulation. Before rigorously defining this concept it will be helpful to understand its building blocks, namely, the simplexes.

<u>Definition 3.1</u>. For each $i = 0,\ldots,m$ define an i-<u>simplex</u> in R^m to be the convex hull of $(i+1)$ points of R^m in general position. Given these points, say $x^0,\ldots, x^i$, the i-simplex is then $\text{hull}(\{x^0,\ldots,x^i\})$.

Let $\tau = \text{hull}(\{x^0,\ldots,x^m\})$ be an m-simplex of R^m. The points $x^0,\ldots,x^m$ are called the <u>vertices</u> of τ. Note that they are actually 0-simplexes. One can also generate very natural 1-simplexes from τ by considering any pair of vertices say x^{i_1} and x^{i_2} and then forming $\text{hull}(\{x^{i_1}, x^{i_2}\})$. In general one can generate for each $j = 0,\ldots,i$ a j-simplex from τ by choosing any subset of $(j+1)$ of the original vertices, say $x^{i_0}, \ldots, x^{i_j}$ and forming $\text{hull}(\{x^{i_0}, \ldots, x^{i_j}\})$. These are called the j-<u>faces</u> of τ. With these concepts in hand the notion of a triangulation is quite understandable.

<u>Definition 3.2</u>. Let M be a nonempty subset of R^m and $\mathcal{M}$ be a finite or countable collection of m-simplexes. Let $\mathcal{M}_i$ for $i = 0,\ldots,m$ be the set of i-faces of members of $\mathcal{M}$. Then $\mathcal{M}$ is a <u>triangulation</u> of M iff

(1) $M = \cup\{\tau / \tau \in \mathcal{M}\}$.

(2) Each pair of m-simplexes are either disjoint or meet in a common face.

(3) Each (m-1) simplex in $\mathcal{M}_{m-1}$ belongs to at most two m-simplexes. (Those (m-1) simplexes which belong to exactly one m-simplex are called boundary (m-1)-simplexes.)

(4) Each point in M has a neighborhood meeting only finite many m-simplexes of $\mathcal{M}$.

Property (1) states that M is the union of all the m-simplexes in $\mathcal{M}$. Properties (2) and (3) describe how these m-simplexes must fit together and property (4) guarantees that each bounded subset of M meets only finitely many m-simplexes of $\mathcal{M}$. This is needed for the proof of finite convergence. A generalization of this notion is presented in Eaves [14] and other pertinent references include Cairns [5], Todd [51,52], Kuhn [32], Scarf and Hansen [25].

The triangulation $\mathcal{M}$ can be used to generate a PL approximation to S through the use of the vertices $\mathcal{M}_0$. First consider a map $f:\mathcal{M}_0 \to R^m$ defined by $f(x) \in S(x)$ for each $x \in \mathcal{M}_0$. There is a unique extension of f to a PL map $f^L: M \to R^m$ defined by

$$f^L(x) = \sum_{i=0}^{m} \lambda^i f(x^i)$$

where $x \in \tau = \text{hull}(\{x^0, \ldots, x^m\}) \in \mathcal{M}$ and $x = \sum_{i=0}^{m} \lambda^i x^i$. Note that by property (2) of D.3.2, f^L is well defined since each $x \in M$ has a unique representation in this form. Furthermore, f^L is linear on each $\tau \in \mathcal{M}$. This map f^L is called a PL approximation to S induced by $\mathcal{M}$. A measure of how closely f^L approximates S is given by

$$\text{mesh}(\mathcal{M}) = \sup\{\max\{\|u-v\| / u, v \in \tau\} / \tau \in \mathcal{M}\}.$$

The algorithm is going to attempt to find a fixed point of f^L. If it is successful it will terminate with an m-simplex $\tau = \text{hull}(\{x^0, \ldots, x^m\})$ and an $x \in \tau$ with $f^L(x) = x$. This may be expressed as the existence of multipliers $\lambda^0, \ldots, \lambda^m \in R^1$ such that

$$[(f(x^0) - x^0, 1), \ldots, (f(x^m) - x^m, 1)]\ (\lambda^0, \ldots, \lambda^m) = (0, \ldots, 0, 1)$$

$$\lambda^i \geq 0 \quad \text{for } i = 0, 1, \ldots, m.$$

Setting $x = \sum_{i=0}^m \lambda^i x^i$ yields $f^L(x) = x$. Such an m-simplex is said to be completely labelled. The next question is how to systematically move through the triangulation in search of this special simplex. This is the topic of the next section.

3.2. Moving Through a Triangulation.

No direct search procedure was found; instead, a method based on a triangulation of $R^m \times [0,1]$ (in which all the vertices belong to either $R^m \times \{0\}$ or $R^m \times \{1\}$) was developed. An example of such a triangulation is depicted in Figure 3.1 for the case $m = 1$.

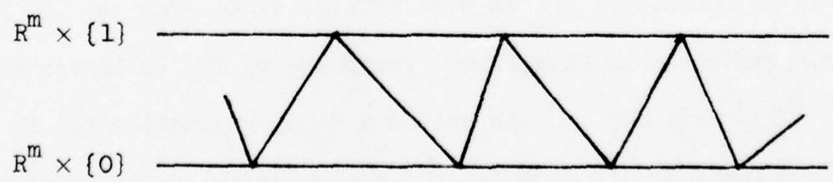


FIGURE 3.1

It is not hard to see that the collection of m-simplexes lying in $R^m \times \{0\}$ induces a triangulation of R^m as does those in $R^m \times \{1\}$. If τ is such an m-simplex let $(\tau)'$ denote the induced m-simplex in R^m obtained by dropping the last coordinate of each point of τ.

Recall that the objective is to find a completely labelled simplex in R^m. The algorithm will be designed in such a way that if there is such an m-simplex it will lie in $R^m \times \{1\}$. More specifically the algorithm will start with a special m-simplex τ^0 of $R^m \times \{0\}$. It will then generate a (possibly infinite) sequence of "adjacent" (m+1) and m-simplexes $\sigma^0, \tau^1, \sigma^1, \tau^2, \ldots$ in such a way that if τ^i lies in $R^m \times \{1\}$ then $(\tau^i)'$ is the desired m-simplex (see Figure 3.2).

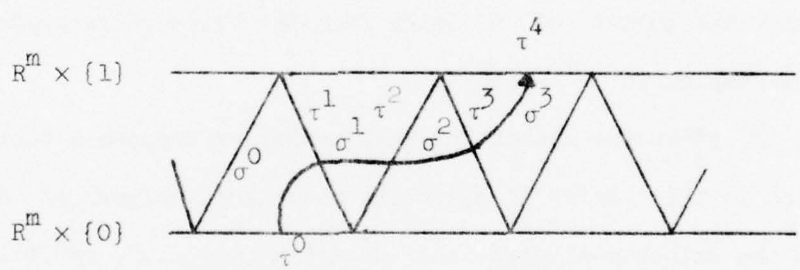


FIGURE 3.2

This movement will now be made mathematically rigorous.

Fix a triangulation $\mathcal{M}$ of $R^m \times [0,1]$ such that each vertex may be expressed as a vector (x,u) where $x \in R^m$ and $u \in \{0,1\}$. For each such vertex define a column vector $A_{\cdot(x,u)} \in R^{m+1}$ by

$$A_{\cdot(x,u)} = \begin{cases} (f(x)-x,\ 1) & \text{if } u = 1 \\ (w-x, 1) & \text{if } u = 0 \end{cases}$$

where $(w,0) \in R^m \times \{0\}$ belongs to the interior of a unique m-simplex τ^0. This vector w is called the starting point, since it is from this initial simplex, τ^0, that the search will begin. Finding an m-simplex in $R^m \times \{1\}$ which contains a linear approximate fixed point is equivalent to finding a basic solution to

$$(*) \qquad A\lambda = (0,\ldots,0,1) \qquad \lambda_i \geq 0 \text{ for } i = 0,\ldots, m$$

such that the columns of A which form the basis corresponds to vertices of an m-simplex in $R^m \times \{1\}$.

To start the search for this m-simplex compute a basic feasible solution to this system of equations which uses columns of A corresponding to the unique m-simplex $\tau^0 = \text{hull}(\{(x^0,0), \ldots, (x^m,0)\})$ in $R^m \times \{0\}$ containing the starting point $(w,0)$. The reader will find it useful to refer frequently to Figure 3.3.

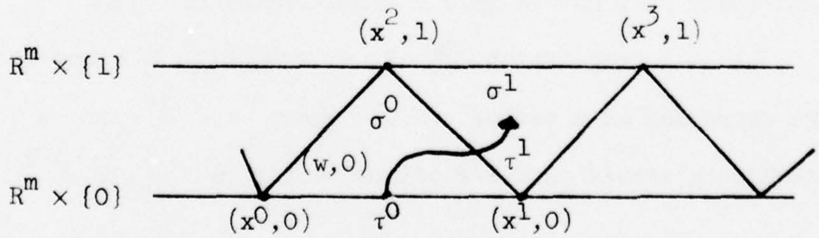


FIGURE 3.3

τ^0 is a boundary m-simplex of the triangulation and by property (3) of D.3.2, it belongs to exactly one (m+1)-simplex, namely, σ^0. Let $(x^{m+1},1)$ be that vertex which when joined to τ^0 produces σ^0. In Figure 3.3 this vertex is $(x^2,1)$. Proceed by bringing $A_{.(x^{m+1},1)}$ into the basis. Some column will drop from the old basis. This vertex corresponds to one of the original vertices $(x^0,0)$, ... , $(x^m,0)$ say $(x^j,0)$. Let τ^1 be the m-simplex obtained from σ^0 by dropping $(x^j,0)$. In Figure 3.3 this corresponds to moving from σ^0 to τ^1 after having dropped $(x^0,0)$. By property (3) of a triangulation since τ^1 is not a boundary m-simplex it must belong to exactly two (m+1)-simplexes. One of them is σ^0. Call the other one σ^1. This again corresponds to a new vertex (x^{m+2},u) for which $A_{.(x^{m+2},u)}$ may be brought into the basis. Continuing in this manner the sequence of m and (m+1)-simplexes τ^0, σ^0, τ^1, ... is generated and each m-simplex has a basic solution to (*). Now only one of four things can happen with this sequence.

Case 1. The algorithm repeats an (m+1)-simplex to which it has already been. Merrill [38] has shown that this cannot happen if the dropping vertex is chosen by a lexicographic resolution technique such as proposed in Dantzig [9] or Gale [20]. Hence cycling may be avoided and this case need not be considered.

Case 2. The algorithm hits a boundary m-simplex in $R^m \times \{0\}$. This case cannot arise becuase the only m-simplex in $R^m \times \{0\}$ with a solution to (*) is τ^0 and Case 1 rules out the possibility of ever returning to σ^0.

Case 3. The algorithm hits a boundary m-simplex in $R^m \times \{1\}$. In this case the algorithm has succeeded and this m-simplex is the one containing the linear approximate fixed point.

Case 4. An infinite sequence of distinct m and (m+1)-simplexes is generated. In this case the algorithm is said to have failed.

Since Case 1 and Case 2 cannot occur, one would like some conditions on the point to set map S which ensures that Case 4 cannot arise. Then only Case 3 can happen and this is what is desired. Before proceeding to give some sufficient conditions on S which rule out Case 4, a subtle point must be cleared up. In describing the algorithm it was stated that a new column $A_{\cdot(x,u)}$ will be brought into the existing basis. The fact that this may always be done is a consequence of the nonnegativity requirement of λ and the boundedness to the solution set to (*) caused by the fact that $\sum_{i=0}^{m} \lambda^i = 1$. The question of finite termination is addressed in the following section.

3.3. Conditions for Finite Convergence.

Recall that the algorithm generates a sequence of m and (m+1)-simplexes $\tau^0, \sigma^0, \tau^1, \ldots$ One way to prevent this sequence from being infinite is to force it to remain inside some compact set, then by property (4) of D.3.2, the sequence must be finite. The way to accomplish this is to give conditions on S which guarantee that outside of some compact set there is no solution to (*), and since each m-simplex τ^i generated by the algorithm satisfies (*), the path cannot leave this compact set. This is the essence of

<u>Theorem 3.1</u>. Let $S: R^m \to (R^m)^*$ be the usable point to set map whose fixed point is sought and let $\mathcal{M}$ be a triangulation of $R^m \times [0,1]$ as described above. Suppose there are positive numbers a and b and a point x' in R^m such that if $z \notin B(x',a)$ then $\langle z'-z, y-x' \rangle < 0$ for all $y \in B(z,b) \setminus B(x',a)$ and $z' \in S(z)$. If $\text{mesh}(\mathcal{M}) \leq b$ then

(a) A linear approximate fixed point of S is computed in a finite number of steps.

(b) Any linear approximate fixed point lies in $B(x', a+b)$.

<u>Proof</u>. See Theorem 5.1 of Merrill [38]. ||

For each application, the hypotheses of the theorem are shown to hold and thus finite convergence is guaranteed. All that remains is to take a sequence $\{\mathcal{M}^k\}$ of triangulations of $R^m \times [0,1]$ with the property that $\text{mesh}(\mathcal{M}^k) \to 0$ as $k \to \infty$. Now a sequence of fixed points will be generated and a convergent subsequence will be enough to yield a fixed point of S as is stated in

<u>Theorem 3.2</u>. Let $\{\mathcal{M}^k\}$ be a sequence of triangulations of $R^m \times [0,1]$ as described above (i.e. $\text{mesh}(\mathcal{M}^k) \to 0$ as $k \to \infty$). If there are positive real numbers a and b together with a point $x' \in R^m$ and

the point to set map S which satisfy the hypotheses of T.3.1 then

(a) Each linear approximate fixed point lies in $B(x', a + b)$ and will be computed in a finite number of steps.

(b) Either $x^k \in S(x^k)$ for some k or any cluster point of $\{x^k\}$ is a fixed point of S, where x^k is the kth linear approximate fixed point.

Proof. See Theorem 5.2 of Merrill [38]. ||

The only thing left to develop is a computerized method for generating the sequence $\{\mathcal{M}^k\}$ of triangulations with $\text{mesh}(\mathcal{M}^k) \to 0$ as $k \to \infty$. Various people have had needs for triangulations and have developed their own, e.g. Kuhn [32], Scarf and Hansen [25], Eaves [15], Merrill [38]. Recently Todd [52] developed a very elegant way to generate a sequence of triangulations as described above. They are referred to as "Union Jack" triangulations. The current version of the computer code uses this method and computational results have indicated that it is quite efficient.

This completes the description of an algorithm which attempts to compute fixed points of usable point to set maps. The next chapter deals with conditions under which this algorithm may be applied to equality constrained optimization problems.

CHAPTER 4

THE APPLICATION OF DECOMPOSABILITY TO EQUALITY CONSTRAINED OPTIMIZATION

4.1. Introduction and Preliminaries.

Consider the problem

$$\text{(P.4.1)} \qquad \begin{aligned} \min\ & P(z) \\ \text{s.t.}\ & G(z) = 0 \\ & Q(z) \leq 0 \\ & z \in R^s \end{aligned} \qquad \text{where} \qquad \begin{aligned} & P: R^s \to R^1 \cup \{\pm\infty\} \\ & G: R^s \to R^n \\ & Q: R^s \to R^\ell \cup \{\pm\infty\} \end{aligned}$$

The objective of this chapter is to state some sufficient conditions on the functions P, G and Q such that (1) P.4.1 may be formulated as a fixed point problem and (2) the algorithm described in Chapter 3 can be used to solve the resulting problem. Since global convergence is sought one might expect that the conditions obtained will be very restrictive. This is in fact the case; however, computational experience is indicating that the method is viable on a reasonable class of problems (see Chapter 7). More specifically the first step will be to develop conditions on P, G and Q so that P.4.1 may be solved by finding the fixed point of some usable point to set map. The second step will be to put additional conditions on these functions so that the resulting point to set map will satisfy the hypotheses of T.3.2. This will ensure global convergence of the algorithm.

Under the next two assumptions, Merrill [38] developed a fixed point formulation for the following special case of P.4.1

$$\text{(P.4.2)} \qquad \begin{array}{rl} \min & p(x) \\ \text{s.t.} & q(x) \leq 0 \\ & x \in R^m \end{array} \quad \text{where} \quad \begin{array}{l} p: R^m \to R^1 \cup \{\pm \infty\} \\ q: R^m \to R^{\ell} \cup \{\pm \infty\} \end{array}$$

<u>Assumption 4.1</u>. Assume that p and q are closed proper convex functions with $\text{dom}(q_i) = R^m$ for each $i = 1, \ldots, \ell$.

<u>Assumption 4.2</u>. Define the function $t: R^m \to R^1$ by $t(x) = \max\{q_i(x)/i = 1, \ldots, \ell\}$. Assume that $\{x \in R^m | t(x) \leq 0\}$ belongs to $\text{int}(\text{dom } p)$. (The function t will have this interpretation for the remainder of the thesis.)

Now one can define the point to set map $S: R^m \to (R^m)^*$ by

$$S(x) = \begin{cases} x - \partial p(x) & \text{if } t(x) < 0 \\ x - \text{hull}(\partial p(x) \cup \partial t(x)) & \text{if } t(x) = 0 \\ x - \partial t(x) & \text{if } t(x) > 0 \end{cases}$$

The first thing to verify is that S is usable.

<u>Theorem 4.1</u>. If A.4.1 and A.4.2 hold then S is usable.

<u>Proof</u>. See Theorem 12.1 of Merrill [38]. ||

The relationship between fixed points of S and solutions to P.4.1 is stated in

<u>Theorem 4.2</u>. If A.4.1 and A.4.2 hold and if $\inf\{t(x)/x \in R^m\} < 0$ then $x' \in S(x')$ iff x' solves P.4.2.

<u>Proof</u>. See Theorem 12.3 of Merrill [38]. ||

Unfortunately this theorem does not guarantee that the algorithm described in Chapter 3 will compute a fixed point of S. In order to accomplish this it is necessary to have

<u>Assumption 4.3</u>. Assume the function q has no common direction of recession other than 0 (see D.1.7).

Now algorithmic convergence is established in

<u>Theorem 4.3</u>. If A.4.1-A.4.3 hold then the hypotheses of T.3.2 are met and in addition if $a = \inf\{t(x)/x \in R^m\}$ then

(a) The condition $a > 0$ may be detected in a finite number of steps.

(b) The condition $a = 0$ implies that any fixed point of S is feasible for P.4.2.

(c) The condition $a < 0$ is a necessary and sufficient condition for a fixed point of S to solve P.4.2 and the algorithm will compute a linear approximate fixed point in a finite number of steps.

<u>Proof</u>. See Theorem 12.3 and Corollary 12.4.1 of Merrill [38]. ||

The next logical step is to find some method of transforming P.4.1 into a problem of the form P.4.2. Perhaps the first thing that comes to mind is to replace the equality constraints $G(z) = 0$ by two equivalent inequality constraints $G(z) \leq 0$ and $-G(z) \leq 0$. Unfortunately, in this case part (b) of T.4.3 says that the algorithm can only be expected to generate a feasible point. Thus it is necessary to find another approach. This is done in the next section.

4.2. Transforming Equality Constrained Optimization into a Fixed Point Problem.

In the spirit of decomposability the idea is to use the equality constraints to solve for some of the variables (called the basic variables) in terms of the rest (called the nonbasic variables). Thus the original variables $z \in R^s$ will often be written (x,y) for $x \in R^m$ and $y \in R^n$, x being the nonbasic variables and y being the basic variables. In order to show how and when this can be done, define a point to set map H on R^m by $H(x) = \{y \in R^n / G(x,y) = 0\}$. Note $H(x)$ might be empty so let $X = \{x \in R^m / H(x) \text{ is nonempty}\}$. The next two assumptions will allow the proper transformation of P.4.1 into P.4.2.

Assumption 4.4. Assume $X = R^m$.

Assumption 4.5. Assume $H(x)$ is a singleton for each $x \in X$.

From A.4.5 it is possible to define a function $h: R^m \to R^n$ by $h(x)$ = that unique element of $H(x)$. Hence $G(x,h(x)) = 0$ for each $x \in R^m$.

From A.4.4 it is possible to define the functions $p:R^m \to R^1$, $q:R^m \to R^\ell$ by $p(x) = P(x,h(x))$ and $q(x) = Q(x,h(x))$, respectively. This is the desired transformation of P.4.1 into P.4.2 as is established in

Theorem 4.4. If A.4.4 and A.4.5 hold and if h, p and q are defined as above then $x \in R^m$ solves P.4.2 iff $(x,h(x))$ solves P.4.1.

Proof. Suppose first that x solves P.4.2. Then $(x,h(x))$ is feasible for P.4.1 since $G(x,h(x)) = 0$ and $Q(x,h(x)) \leq 0$. In order to show that $(x,h(x))$ actually solves P.4.1 let (x',y') be any other feasible solution to P.4.1. It must be shown that $P(x,h(x)) \leq P(x',y')$. Since $y' \in H(x') = \{h(x')\}$ one need only show that $P(x,h(x)) \leq P(x',h(x'))$, but this follows from the optimality of x for P.4.2 since $P(x,h(x)) = p(x) \leq p(x') = P(x',h(x'))$. This proves the necessary part of the theorem. For the sufficiency, suppose $(x,h(x))$ solves P.4.1. Then x is certainly feasible for P.4.2. In order to show that x actually solves P.4.2 let x' be any other feasible solution. It must be shown that $p(x) \leq p(x')$. Since $(x',h(x'))$ is also feasible for P.4.1 it follows that $p(x) = P(x,h(x)) \leq P(x',h(x')) = p(x')$ as desired. ||

Now that the transformation of P.4.1 into P.4.2 has been established it is desirable to have conditions on P, G, and Q which ensures A.4.1-A.4.5. The next theorem puts sufficient conditions on G which makes A.4.4 and A.4.5 hold. It is a modification of the implicit function theorem and is stated as

<u>Theorem 4.5</u>. Let $G:R^m \times R^n \to R^n$ be continuously differentiable on $R^m \times R^n$ and suppose there is a point $(a,b) \in R^m \times R^n$ such that (1) $G(a,b) = 0$ and (2) the matrix $E = D_y G(a,b)$ is nonsingular. Also suppose there is a constant $0 < \lambda < 1$ such that

$$\|I - E^{-1} D_y G(x,y)\| \leq \lambda \quad \text{for all} \quad (x,y) \in R^m \times R^n .$$

Under these conditions A.4.4 and A.4.5 hold.

<u>Proof</u>. It will be shown that there is a unique function $h:R^m \to R^n$ such that $G(x,h(x)) = 0$ for each $x \in R^m$. To this end define a new function $L:R^m \times R^n \to R^n$ by $L(x,y) = y - E^{-1}G(x,y)$. For each fixed value of x, L will be shown to be a contraction mapping and hence will have a unique fixed point $h(x) \in R^n$. Note that $L(x,h(x)) = h(x)$ iff $G(x,h(x)) = 0$. To show that L is a contraction map in the y coordinate, the constant λ in the hypothesis will be used to conclude that

$$\|L(x,y) - L(x,y')\| \leq \lambda\|y-y'\| \quad \text{for all} \quad y,\ y' \in R^n .$$

So it is necessary to bound $\|L(x,y) - L(x,y')\|$. This is precisely the essence of T.1.1. In order to apply it, the function L need only be differentiable in the y-coordinates which of course it is and in fact

$$D_y L(x,y) = I - E^{-1} D_y G(x,y) .$$

Upon applying the bound in T.1.1,

$$\|L(x,y) - L(x,y')\| \leq \sup\{\|D_y L(x,y+\lambda'(y-y'))\| / 0 \leq \lambda' \leq 1\} \|y-y'\|$$

$$= \sup\{\|I - E^{-1} D_y G(x,y+\lambda'(y-y'))\| / 0 \leq \lambda' \leq 1\} \|y-y'\|$$

$$\leq \lambda\|y-y'\| ,$$

the last inequality being justified by the hypothesis. Thus in fact L is a contraction mapping in the y-coordinates for each fixed value of the x-coordinates and this completes the proof. $\|$

__Corollary 4.1__. If G is linear and has rank n then A.4.4 and A.4.5 hold.

__Proof__. Let $G(x,y) = Ux + Vy + w$ where V is an $(n \times n)$ nonsingular matrix, U is an $(n \times m)$ matrix and w is an n-vector. Choose any λ with $0 < \lambda < 1$ and set $(a,b) = (0, -V^{-1}w)$. Now the hypotheses of T.4.5 hold so A.4.4 and A.4.5 do also. $\|$

Supposing now that this function $h(x)$ exists, conditions can be placed on P, Q and h so that A.4.1-A.4.3 hold.

__Proposition 4.1__. Suppose P and Q are closed proper convex functions and that there exists x', $x'' \in R^m$ such that $P(x',h(x')) < \infty$ and $Q(x'',h(x'')) < \infty$. If h is linear then A.4.1 holds.

__Proof__. It must be shown that p and q are closed proper convex functions with $\mathrm{dom}(q_i) = R^m$ for each $i = 1,\ldots,\ell$. Since h is defined on all of R^m (by A.4.4), $\mathrm{dom}(q_i) = R^m$ for each $i = 1,\ldots,\ell$.

Next p is shown to be convex. To do this let $x, w \in R^m$ and $\lambda \in [0,1]$. Then by the linearity of h and the convexity of P it follows that

$$\begin{aligned} p(\lambda x + (1-\lambda)w) &= P(\lambda x + (1-\lambda)w, h(\lambda x + (1-\lambda)w)) \\ &= P(\lambda x + (1-\lambda)w, \lambda h(x) + (1-\lambda)h(w)) \\ &= P(\lambda(x,h(x)) + (1-\lambda)(w,h(w))) \\ &\leq \lambda P(x,h(x)) + (1-\lambda) P(w,h(w)) \\ &= \lambda p(x) + (1-\lambda) p(w) . \end{aligned}$$

That p is closed and proper is straightforward. A similar argument shows q to be a closed proper convex function also. ||

In order to weaken the assumption that h is linear it is necessary to place additional structure on P and Q. This is done in

Proposition 4.2. Suppose P and Q are closed proper convex functions and that there exists $x', x'' \in R^m$ such that $P(x',h(x')) < \infty$ and $Q(x'',h(x'')) < \infty$. Suppose in addition that P and Q are isotone (see D.1.8) in the y-coordinates for each fixed value of the x-coordinates. If h is convex then A.4.1 holds.

Proof. As in the previous proposition, $\mathrm{dom}(q_i) = R^m$ for each $i = 1,\ldots,\ell$. Next p is shown to be convex. To do this let $x, w \in R^m$ and $\lambda \in [0,1]$. From the convexity of h,

$$h(\lambda x + (1-\lambda)w) \leq \lambda h(x) + (1-\lambda) h(w) ,$$

and since P is isotone in the y-coordinates it follows that

$$P(\lambda x + (1-\lambda)w,\ h(\lambda x + (1-\lambda)w)) \le P(\lambda x + (1-\lambda)w,\ \lambda h(x) + (1-\lambda)w,\ \lambda h(w))$$
$$= P(\lambda(x,\ h(x)) + (1-\lambda)(w, h(w)))$$
$$\le \lambda P(x, h(x)) + (1-\lambda)\ P(w, h(w))$$

the last inequality being justified by the convexity of P. The left most side of the inequality is $p(\lambda x + (1-\lambda)w)$ and the right most side is $\lambda p(x) + (1-\lambda)\ p(w)$. Hence p is convex. That p is closed and proper is straightforward. A similar argument shows q to be a closed proper convex function. ||

Proposition 4.3. If $\mathrm{dom}(P) = R^s$ then A.4.2 holds.

Proof. Obvious. ||

Proposition 4.4. If Q has no common direction of recession other than 0 and if h is linear then A.4.3 holds.

Proof. This is done by contradiction so assume $d \in R^m$ is a nonzero common direction of recession of q. Then there is a $b \in R^1$ and an $x \in R^m$ such that $q_i(x + \lambda d) \le b$ for all $\lambda \ge 0$ and $i = 1,\ldots,\ell$. It will be shown that $(d, h(d))$ is a nonzero common direction of recession of Q. Clearly $(d, h(d))$ is nonzero since d is. From the linearity of h it follows that for all $\lambda \ge 0$ and $i = 1,\ldots,\ell$

$$Q_i((x, h(x)) + \lambda(d, h(d))) = Q_i(x + \lambda d, h(x) + \lambda h(d))$$
$$= Q_i(x + \lambda d,\ h(x + \lambda d))$$
$$= q_i(x + \lambda d)$$
$$\le b.$$

Hence $(d, h(d))$ is the desired common direction of recession of Q and this contradiction proves the claim. ||

In order to weaken the assumption that h is linear it is necessary to place additional structure on P and Q. This is done in

Proposition 4.5. Suppose h is convex. Assume also that Q is convex and isotone in the y-coordinates. If Q has no common direction of recession other than 0 then A.4.3 holds.

Proof. Again this will be done by contradiction, so assume $d \in R^m$ is a nonzero common direction of recession of q. Hence there is a $b \in R^1$ and $x \in R^m$ such that $q_i(x + \lambda d) \leq b$ for each $\lambda \geq 0$ and $i = 1,\ldots,\ell$. To construct a nonzero common direction of recession of Q let W (an $(n \times m)$ matrix) be a subgradient of h at x (see D.1.2). It will be shown that (d,Wd) is a nonzero common direction of recession of Q. It is clearly nonzero since d is. From the definition of a subgradient,

$$h(x) + \lambda(Wd) \leq h(x + \lambda d) \quad \text{for all} \quad \lambda \geq 0;$$

and since Q is isotone in the y-coordinates it follows that for all $\lambda \geq 0$ and $i = 1,\ldots,\ell$,

$$\begin{aligned} Q_i(x + \lambda d,\ h(x) + \lambda Wd) &\leq Q_i(x + \lambda d,\ h(x + \lambda Wd)) \\ &= q_i(x + \lambda d) \\ &\leq b\ . \end{aligned}$$

Hence (d,Wd) is the desired common direction of recession of Q and this contradiction proves the claim. ||

In applications it is very often the case that neither A.4.4 nor A.4.5 will hold. Take for example the function $G: R^2 \to R^1$ by $G(x,y) = x^2 + y^2 - 1$. In this case $X = [-1,1]$ and $H(x) = \{\sqrt{1-x^2}, -\sqrt{1-x^2}\}$ for each $x \in X$. One would like to be able to find a fixed point formulation of P.4.1 in which the resulting point to set map is usable. There are several problems that immediately present themselves. The first is that the function h is no longer well defined. Recall that for each $x \in R^m$, $h(x)$ was that unique element such that $G(x,h(x)) = 0$. In the current situation there may be many choices for $h(x)$ (since $H(x)$ need not be a singleton). Even supposing one were able to construct this choice function $h(x)$, the next problem is that h is a mapping from X into R^n. If, as before, one were to define $p(x) = P(x,h(x))$ and $q(x) = Q(x,h(x))$ for each $x \in X$ then A.4.1 will not hold unless $X = R^m$. If $X \neq R^m$ then the resulting point to set map may not be usable (even though X might be compact and convex). No usable point to set map has been found to overcome these difficulties, however, one which is very close and has worked in practice is described in:

<u>Theorem 4.6</u>. Suppose X is a compact convex subset of R^m with $x' \in X$. Suppose further that A.4.5 holds so there is an $h: X \to R^n$ such that $G(x,h(x)) = 0$ for each $x \in X$. Let $p: X \to R^1 \cup \{\pm\infty\}$ be defined by $p(x) = P(x,h(x))$ and let $q: X \to R^{\ell}$ be defined by $q(x) = Q(x,h(x))$ and suppose p and q are closed proper convex functions on $\text{int}(X)$. Then the point to set map $S: R^m \to (R^m)^*$ defined by

$$S(X) = \begin{cases} \{x'\} & \text{if } x \notin \text{int}(X) \\ \begin{cases} x - \partial p(x) & \text{if } t(x) < 0 \\ x - \text{hull}(\partial p(x) \cup \partial t(x)) & \text{if } t(x) = 0 \\ x - \partial t(x) & \text{if } t(x) > 0 \end{cases} & \text{if } x \in \text{int}(X) \end{cases}$$

has the following properties:

(a) If $x' \in \text{int}(X)$ then any fixed point of S belongs to $\text{int}(X)$.

(b) If $x' \in \text{int}(X)$ then any fixed point of S solves the problem

$$\begin{aligned} &\min \quad p(x) \\ &\text{s.t.} \quad q(x) \leq 0 \qquad \text{provided } a = \inf\{t(x)/x \in \text{int}(X)\} < 0 . \\ &\qquad\ \ x \in \text{int}(X) \end{aligned}$$

(c) The algorithm when implemented on S computes either a fixed point of S or a point in $\text{bd}(X)$.

Proof. Part (a). Let $x \in S(x)$. Suppose contrary to the conclusion of (a) that $x \notin \text{int}(X)$, then $S(x) = \{x'\}$ and so $x = x' \in \text{int}(X)$. This contradiction establishes that any fixed point of S belongs to $\text{int}(X)$.

Part (b). Let $x \in S(x)$. First it will be shown that x is feasible, i.e. $q(x) \leq 0$ and $x \in \text{int}(X)$. Part (a) shows that $x \in \text{int}(X)$ and by hypothesis $a < 0$ hence $q_i(x) \leq t(x) \leq 0$ for each $i = 1, \ldots, \ell$. Thus x is feasible. If $t(x) < 0$ then the fact that x is a fixed point of S yields $0 \in \partial p(x)$ and the optimality of x follows from the convexity of p on $\text{int}(X)$. If, on the other hand, $t(x) = 0$ then $0 = \lambda a' + (1-\lambda)b'$ for some $a' \in \partial p(x)$ and $b' \in \partial t(x)$

and $\lambda \in [0,1]$. Since $a < 0$ and $t(x) = 0$ it follows that $\lambda > 0$ whence $a' = [(1-\lambda)/\lambda]b'$. To see that x is actually optimal let $z \in \mathrm{int}(X)$ be any other feasible point. It must be shown that $p(z) \geq p(x)$. By the fact that z is feasible and that $b' \in \partial t(x)$,

$$0 \geq t(z) \geq t(x) + \langle b', z-x \rangle = \langle b', z-x \rangle ,$$

and since $a' \in \partial p(x)$,

$$\begin{aligned} p(z) &\geq p(x) + \langle a', z-x \rangle \\ &= p(x) - [(1-\lambda)/\lambda] \langle b', z-x \rangle \geq p(x) . \end{aligned}$$

This shows that x actually solves the problem.

Part (c). The proof will be done by showing that the algorithm generates a point $x \in X$ with the additional property that if $x \notin \mathrm{bd}(X)$ then $x \in S(x)$. To actually construct this x, consider a sequence of triangulations $\{\mathcal{M}^k\}$ of $R^m \times [0,1]$ whose mesh is going to 0 for $k \to \infty$. For each fixed value of k the algorithm will generate a sequence of m and $(m+1)$-simplexes such that each m-simplex is completely labelled. Since this collection can never leave a compact set it must be finite and so there will be a completely labelled m-simplex, say $\tau^k = \mathrm{hull}(\{(x^{0k},1), \ldots, (x^{mk},1)\})$ consequently there are scalars $\lambda^{0k}, \ldots, \lambda^{mk} \geq 0$ together with points $z^{0k} \in S(x^{0k}), \ldots, z^{mk} \in S(x^{mk})$ such that

$$\sum_{i=0}^{m} \lambda^{ik} z^{ik} = \sum_{i=0}^{m} \lambda^{ik} x^{ik}$$

$$\sum_{i=0}^{m} \lambda^{ik} = 1 .$$

Since the $\text{mesh}(\mathcal{M}^k)$ is going to 0, it follows that there are points $x, z^0, \ldots, z^m \in R^m$ together with nonnegative scalars $\lambda^0, \ldots, \lambda^m$ and a subsequence K such that

(1) $\{x^{ik}\} \to x$ for $k \in K$ and each $i = 0,\ldots,m$.

(2) $\{z^{ik}\} \to z^i$ for $k \in K$ and each $i = 0,\ldots,m$.

(3) $\{\lambda^{ik}\} \to \lambda^i$ for $k \in K$ and each $i = 0,\ldots,m$.

(4) $x = \sum_{i=0}^{m} \lambda^i z^i$.

First it will be shown that $x \in X$. If $x \notin X$ then $\{z^{ik}\} \to x'$ for $k \in K$ and each $i = 0,\ldots,m$. Hence $x = x' \in \text{int}(X)$. This contradiction establishes that $x \in X$.

Next it will be shown that if $x \notin \text{bd}(X)$ then $x \in S(x)$. So suppose $x \notin \text{bd}(X)$. Then $x \in \text{int}(X)$ and hence $S(x)$ is a convex set. To show $x \in S(x)$ it will be shown that $z^i \in S(x)$ for each $i = 0,\ldots,m$ and thus $x = \sum_{i=0}^{m} \lambda^i z^i \in S(x)$. To see that $z^i \in S(x)$ for each $i = 0,\ldots,m$ note that $S(x)$ is usable on $\text{int}(X)$ so by the upper semicontinuity of S and (1) and (2) above, $z^i \in S(x)$ for each $i = 0,\ldots,m$. This concludes the proof. ||

Throughout the entire chapter it has been assumed that A.4.5 always held and an example was presented where A.4.5 did not hold. It will be shown that there is a possible method for dealing with this type of problem but certain aspects of the approach render it computationally infeasible. If A.4.4 holds then the function $h(x)$ will choose a very special element of $H(x)$ in such a way that x solves P.4.2 iff $(x,h(x))$ solves P.4.1. Unfortunately no computational implementation for this choice function has been found. This function is described in

Theorem 4.7. Suppose $H(x)$ is compact and convex for each $x \in R^m$ and that P is continuous. Suppose further that for each $x \in R^m$ there is a $y \in H(x)$ such that $Q(x,y) \leq 0$. Then the choice function $h: R^m \to R^n$ defined by $h(x)$ = any solution to

$$\begin{aligned} \min\ & P(x,y) \\ & Q(x,y) \leq 0 \\ & y \in H(x) \end{aligned}$$

satisfies x solves P.4.2 iff $(x,h(x))$ solves P.4.1.

Proof. Suppose first that x solves P.4.2. Then $(x,h(x))$ is feasible for P.4.1. Let (x',y') be any other feasible solution for P.4.1. Then by the definition of h, $P(x',y') \geq P(x',h(x')) = p(x') \geq p(x) = P(x,h(x))$ and so $(x,h(x))$ actually solves P.4.1. This proves the necessary part of the theorem. For the sufficiency part suppose $(x,h(x))$ solves P.4.1. Then x is certainly feasible for P.4.2. Let x' be any other feasible point for P.4.2. It must be shown that $p(x') \geq p(x)$. Since $(x',h(x'))$ is feasible for P.4.1 it follows that

$$p(x) = P(x,h(x)) \leq P(x', h(x')) = p(x') . \qquad \|$$

This chapter has dealt with theoretical conditions on the original functions P, G and Q which allows the algorithm described in Chapter 3 to solve P.4.1. It is now time to focus on the computational aspects of the problem. It will also be desirable to find some techniques to increase the efficiency of the algorithm. These are the topics of the next two chapters.

CHAPTER 5

COMPUTER CONSIDERATIONS

5.1. Introduction and Preliminaries.

In the previous chapter a theoretical approach was suggested for solving P.4.1. There is however a considerable gap between the theoretical framework and an actual computer implementation. In this chapter a computational method based on the previous theory is developed in detail. The first step in this direction involves a careful analysis of exactly what quantities must be computed and methods are proposed for accomplishing these tasks efficiently. It will become evident that the amount of work required is quite large and one naturally asks if there are ways to reduce the computational effort. This topic however will be reserved for Chapter 6.

Given P.4.1 recall that for each $x \in R^m$, $H(x) = \{y \in R^n / G(x,y) = 0\}$ and $X = \{x \in R^m / H(x) \text{ is nonempty}\}$. At the end of Chapter 4 it was shown that the case where $H(x)$ was not a singleton for each $x \in X$, was a very difficult and seemingly unmanageable problem. Therefore for the duration of this chapter, A.4.5 will be assumed. Also recall that the following point to set map was developed to solve P.4.1,

$$S(x) = \begin{cases} \{x'\} & \text{if } x \notin \text{int}(X) \\ \begin{cases} x - \partial p(x) & \text{if } t(x) < 0 \\ x - \text{hull}\,(\partial p(x) \cup \partial t(x)) & \text{if } t(x) = 0 \\ x - \partial t(x) & \text{if } t(x) > 0 \end{cases} & \text{if } x \in \text{int}(X) \end{cases}$$

where $x' \in X$. Finally, recall how the algorithm of Chapter 3 will attempt to find a fixed point of S. It will use a triangulation of $R^m \times [0,1]$ in which the vertices lie in either $R^m \times \{0\}$ or $R^m \times \{1\}$. The algorithm starts with a special m-simplex, say, $\tau^0 = \text{hull}(\{(x^0,1),\ldots,(x^m,1)\})$ containing the starting point $(w,0) \in R^m \times \{0\}$ in its interior. Consequently there is a basic feasible solution λ to

$$A\lambda = (0,\ldots, 0, 1)$$
$$\lambda \geq 0$$

where

$$A_{\cdot(x,u)} = \begin{cases} (f(x)-x,\ 1) \quad \text{for some } f(x) \in S(x) & \text{if } u = 1 \\ (w-x,1) & \text{if } u = 0\ . \end{cases}$$

The method proceeds to generate a vertex in the triangulation of the form (x,u) with $u \in \{0,1\}$. For each such point it is necessary to compute $A_{\cdot(x,u)}$. This vector will be brought into the current basis and a dropping vertex will be uniquely determined. This in turn will generate a new incoming vertex and the process is repeated. Note that when then incoming vertex (x,u) has $u = 0$, the computation of $A_{\cdot(x,u)}$ is very simple, however when $u = 1$ it is necessary to generate a point $f(x) \in S(x)$. The flow chart of Figure 5.1 shows the necessary computations to accomplish this. The first three computations are discussed in the next section and the other four computations are handled in the last section.

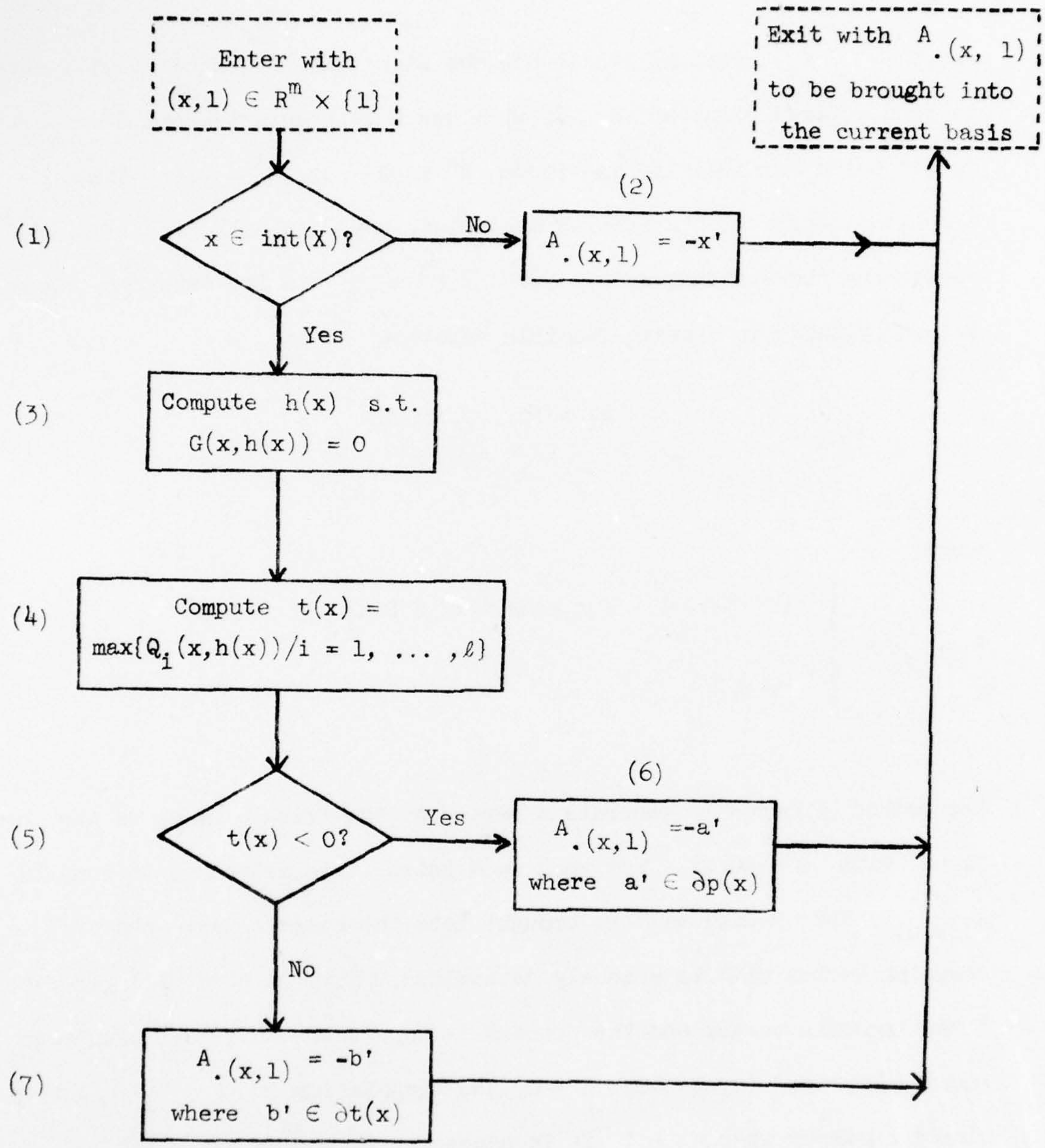

Enter with
(x,1) ∈ R^m × {1}
(1)
x ∈ int(X)?
No
(2)
A.(x,1) = -x'
Exit with A.(x, 1)
to be brought into
the current basis
Yes
(3)
Compute h(x) s.t.
G(x,h(x)) = 0
(4)
Compute t(x) =
max{Q_i(x,h(x))/i = 1, ... ,ℓ}
(5)
t(x) < 0?
Yes
(6)
A.(x,1) =-a'
where a' ∈ ∂p(x)
No
(7)
A.(x,1) = -b'
where b' ∈ ∂t(x)

FIGURE 5.1

5.2. Solving Nonlinear Systems of Equations.

Step (1) of Figure 5.1 is ascertaining if $x \in \text{int}(X)$. Amongst other things, this requires determining if there is a $y \in R^n$ such that $G(x,y) = 0$. Since this is attempting to solve a particular nonlinear system of n equations in n unknowns it is going to be a difficult task. Several computational methods exist for accomplishing this. In the proof of T.4.5, for example, a contraction mapping approach was designed, however it is felt that the hypotheses of this theorem are too strong and that in most applications the function G will not satisfy these conditions. A more stable computational tool was sought. In the event that these constraints satisfy some differentiability conditions, Newton's method may be applied. These conditions are supplied in

Theorem 5.1. Let $x \in X$ be fixed and suppose $y \in R^n$ is such that $G(x,y) = 0$. Suppose G is continuously differentiable (in the y-coordinates) in some neighborhood of y and that $D_y G(x,y)$ is nonsingular at y. Then there is a neighborhood O of y such that for any $y^0 \in O$ the sequence $\{y^k\}$ defined by

$$y^{k+1} = y^k - D_y^{-1} G(x,y^k)\ G(x,y^k), \qquad k = 0,1,\ldots$$

converges to y.

Proof. See Theorem 10.2.2 of Ortega and Rhineboldt [41]. ||

Optimally one would have hoped for a stronger version of T.5.1 in which the sequence $\{y^k\}$ converged to some $y \in R^n$ iff $G(x,y) = 0$, that is to say iff $x \in X$. This of course is not the case as it is possible for Newton's method to diverge and yet $x \in X$. Furthermore there is no way of determining if $x \in \text{int}(X)$. No way around these difficulties is known so in practice when Newton's method fails it is assumed that $x \notin \text{int}(X)$. This action has not caused any difficulties in the test problems reported in Appendix A. Note that when Newton's method fails it is easy to compute $f(x) \in S(x)$ provided $x' \in X$ is available. If such an x' is not available at the start of a problem, a Phase I method for attempting to find such an x' exists. The next proposition shows how to set this up.

Proposition 5.1. Consider the problem

$$\text{(P.5.1)} \qquad \min \sum_{i=1}^{n} z_i$$

$$\text{s.t.} \quad G(x,y) + z = 0$$

$$z \geq 0$$

$$z \in R^n .$$

Then $x' \in X$ iff there is a $y' \in R^n$ such that $(x',y',0)$ solves P.5.1.

Proof. Suppose first that $x' \in X$. This means there is a $y' \in R^n$ such that $G(x',y') = 0$. Thus $(x',y',0)$ is feasible for P.5.1. To see that it is actually optimal note that the objective value is bounded below by 0 and since $(x',y',0)$ actually attains that value it must be optimal. To go the other way suppose $(x',y',0)$ solves P.5.1. From the feasibility conditions, $G(x',y') = 0$ and hence $x' \in X$. ||

As a result of this proposition it will now be assumed that $x' \in X$ is available and returning to Figure 5.1(2), this completes the description of what to do if Newton's method fails to converge.

Note that when implementing Newton's method it is necessary to supply an initial starting point $y^0 \in R^n$ from which to iterate. A good choice for y^0 will hopefully mean that fewer iterations are required for convergence. On the other hand a poor choice might lead to slow convergence or no convergence at all. Under certain differentiability assumptions on G, a tangent plane approximation can sometimes be used to generate a good initial starting point. This is the essence of

Theorem 5.2. Suppose G is twice continuously differentiable and that the point $(a,b) \in R^m \times R^n$ satisfies $G(a,b) = 0$ with $D_yG(a,b)$ nonsingular and let $x \in R^m$. Then there is a constant e, such that

$$\|G(x,y^0)\| \leq e\|(x,y^0) - (a,b)\|^2$$

where

$$y^0 = b - D_y^{-1}G(a,b)(D_xG(a,b)(x-a)) .$$

Proof. Since $G(a,b) = 0$ and by the definition of y^0,

$$\begin{aligned} G(x,y^0) &= G(x,y^0) - G(a,b) - D_xG(a,b)(x-a) - D_yG(a,b)(y^0-b) \\ &= G(x,y^0) - G(a,b) - DG(a,b)(x-a,y^0-b) . \end{aligned}$$

Set $e = \sup\{\|D^2G((x,y^0) + \lambda((a,b) - (x,y^0)))\| / 0 \le \lambda \le 1\} < \infty$. The result now follows from T.1.2. $\|$

The reason this starting point was chosen was because all of the quantities needed to generate it will already have been computed by the algorithm. That is to say, at the moment the algorithm needs a starting point for Newton's method it will have a point (a,b) with $G(a,b) = 0$ and it will have computed $-D_y^{-1}G(a,b)\ D_xG(a,b)$. Therefore, to compute y^0 all that needs to be done is a matrix multiplication and an addition. This concludes the analysis of Newton's method and it will henceforth be assumed that Newton's method has converged to $y \in R^n$ with $G(x,y) = 0$. According to Figure 5.1(4) one must now compute

$$t(x) = \max\{Q_i(x,h(x))/i = 1, \ldots, \ell\} .$$

Following this however it will be necessary to generate either an $a' \in \partial p(x)$ (if $t(x) < 0$) or $b' \in \partial t(x)$ (if $t(x) \geq 0$). It will be shown in the next section that under certain conditions these quantities may be obtained from the original functions P, G and Q.

5.3. Computing Subgradients.

In this section it will be shown that if P, G, Q and h satisfy some differentiability conditions it will be possible to compute subgradients of either $p(x)$ or $t(x)$. First it will be shown how to compute $\nabla p(x)$ and $Dq(x)$ from the derivatives of the initial functions P, G and Q, and they in turn will be used to compute subgradients.

Proposition 5.2. Suppose P, G, Q and h are differentiable at some point $(x,h(x))$ with $x \in \text{int}(X)$. If in addition $D_y G(x,h(x))$ is nonsingular then p and q are differentiable at x and

$$\nabla p(x) = \nabla_x P(x,h(x)) + (\nabla_y P(x,h(x)))^T \, Dh(x)$$

$$Dq(x) = D_x Q(x,h(x)) + D_y Q(x,h(x)) \, Dh(x)$$

with

$$Dh(x) = -D_y^{-1} \, G(x,h(x)) \, D_x G(x,h(x)) \; .$$

Proof. That p and q are differentiable follows from the chain rule. Furthermore the $\nabla p(x)$ and $Dq(x)$ may be written explicitly as

$$\nabla p(x) = \nabla_x P(x,h(x)) + (\nabla_y P(x,h(x)))^T \, Dh(x)$$

$$Dq(x) = D_x Q(x,h(x)) + D_y Q(x,h(x)) \, Dh(x)$$

It remains only to show that $Dh(x) = -D_y^{-1} G(x,h(x)) \, D_x G(x,h(x))$. The function $G(x,h(x))$ is identically 0 on $\text{int}(X)$ so again applying the chain rule yields

$$D_x G(x,h(x)) + D_y G(x,h(x)) \, Dh(x) = 0 \; .$$

By hypothesis, $D_y G(x,h(x))$ is nonsingular, therefore an explicit formula for $Dh(x)$ is given by

$$Dh(x) = -D_y^{-1} G(x,h(x)) \, D_x G(x,h(x)) \; . \qquad \|$$

Lemma 5.1. If p and q are closed proper convex functions on $\text{int}(X)$ and if p and q are differentiable on $\text{int}(X)$ then $\nabla p(x) \in \partial p(x)$ and $Dq(x) \in \partial q(x)$ for each $x \in \text{int}(X)$.

Proof. This is an immediate consequence of T.1.3. ||

Theorem 5.3. Let p and q be closed proper convex functions which are differentiable on $\text{int}(X)$. Suppose i is such that $q_i(x) = \max\{q_j(x)/1 \leq j \leq \ell\} = t(x)$. Then $\nabla q_i(x) \in \partial t(x)$ and $\nabla p(x) \in \partial p(x)$.

Proof. Let $z \in \text{int}(X)$. It will be shown that $t(z) \geq t(x) + \langle \nabla q_i(x), z-x \rangle$ for then $\nabla q_i(x) \in \partial t(x)$. But since $t(x) = q_i(x)$ and since q_i is convex,

$$
\begin{aligned}
t(x) + \langle \nabla q_i(x), z-x \rangle &= q_i(x) + \langle \nabla q_i(x), z-x \rangle \\
&\leq q_i(z) \\
&\leq t(z) \quad ,
\end{aligned}
$$

as desired. The fact that $\nabla p(x) \in \partial p(x)$ is a restatement of L.5.1. ||

With Proposition 5.2 and T.5.2 it is possible to compute either an element of $\partial p(x)$ or $\partial t(x)$ and with these computations under control all of Figure 5.1 has been dealt with.

This chapter has provided an implementation of an algorithm to solve P.4.1. Several difficulties remain unresolved but computational tests have shown this method to be quite reliable (see Chapter 7).

Specifically, a scheme for computing $f(x) \in S(x)$ was analyzed. First it is necessary to attempt to solve a nonlinear system of equations. If this is successful it is then necessary to evaluate partial derivatives and subsequently solve a system of linear equations. This clearly requires a lot of effort. It is therefore desirable to find ways of reducing this work. This is the topic of Chapter 6.

CHAPTER 6

ACCELERATION TECHNIQUES

6.1. Introduction and Preliminaries.

In the previous chapter a computational method was devised to solve P.4.1. It was observed that each time a point $(x,u) \in R^m \times \{1\}$ was generated a large effort was required to compute $f(x) \in S(x)$. The purpose of this chapter is to describe various ways of improving the overall efficiency of the algorithm.

An obvious approach is to try to reduce the amount of work needed to generate $f(x) \in S(x)$. One way of doing this is to approximate S by a sequence $\{S^k\}$ of point to set maps. Then instead of generating fixed points to PL approximations of S (as the algorithm currently does) one would generate fixed points of PL approximations to S^k. The idea is that it will be less expensive to generate an element of $S^k(x)$ than one of $S(x)$. Another possible way to save work in generating $f(x) \in S(x)$ is to reduce the amount of work needed to compute $h(x)$ by using a Simplified Newton Method (SNM) (see Ortega and Rhineboldt [41] for example) or perhaps a Quasi-Newton Method (QNM) (see Murray [39] or Broyden, Dennis and More [4]). Whereas NM requires a matrix inversion at each of its interations, SNM requires only one matrix inversion for the entire procedure. Still another idea for saving work is to avoid computing $t(x) = \max\{q_i(x)/i = 1,\dots, \ell\}$ which requires evaluating all of the constraints. Instead one might hope to be able to evaluate the constraints sequentially and stop as soon as a violated constraint is encountered. This was observed by Merrill [38] and he presented an example consisting of convex functions for which the above idea did not work. A way around this difficulty has been found and has been implemented in the computer code.

All of the previous ideas have involved reducing the amount of work needed to compute $f(x) \in S(x)$. Another valuable approach might be to somehow reduce the total number of times $f(x) \in S(x)$ needs to be computed. One possibility is to use additional structure of P.4.1 such as upper and lower boundes on the variables. A triangulation of the hyperrectangle defined by these upper and lower bounds has been developed. Intuitively this would appear to be good since outside of the hyperrectangle no solution can possibly exist. Yet another possible way to reduce the total number of times $f(x) \in S(x)$ need be computed is suggested by Saigal [46]. His analysis can only be applied to P.2.1 in which there are no inequality constraints and requires some differentiability conditions on the functions. In this case the rate at which the mesh of the triangulation goes to zero can be greatly increased. Each of these ideas is now made mathematically concrete.

6.2. Approximation Techniques.

Consider the problem of finding a fixed point of a function $f: R^m \to R^m$. In many applications the evaluation of f can be very time consuming. It would therefore seem reasonable to attempt to reduce the amount of work needed to evaluate f. This will be accomplished by replacing f by a sequence of functions $\{f^k\}$. Under certain circumstances the algorithm of Chapter 3 may be used to compute a linear approximate fixed point of f^k in a finite number of steps, and if the sequence of points thus generated has a cluster point then this can be shown to be a fixed point of f. The motivation for this approach lies in the fact that each f^k is easier to evaluate than the original f. The circumstances under which this approach will work is now developed. It is necessary to introduce

<u>Definition 6.1</u>. A sequence $\{f^k\}$ of functions is said to be <u>weakly equicontinuous</u> iff for each $\epsilon > 0$ there is a $\delta > 0$ and an integer $N > 0$ such that $\|x-z\| < \delta$ implies $\|f^k(x) - f^k(z)\| < \epsilon$ for all $k > N$.

The next two lemmas will be used in the proof of the main theorem.

<u>Lemma 6.1</u>. If $\{f^k\}$ converges pointwise to f and if $\{f^k\}$ is weakly equicontinuous then for any sequence of points $\{x^k\}$ converging to $x \in R^m$, the sequence $\{f^k(x^k)\}$ converges to $f(x)$.

<u>Proof</u>. Let $\epsilon > 0$. By D.6.1 there is a $\delta > 0$ and an integer $N_1 > 0$ such that $\|x-y\| < \delta$ implies $\|f^k(x) - f^k(y)\| < \epsilon/2$ for all $k > N_1$. Since $\{x^k\}$ converges to x one may choose an integer $N_2 > 0$ such that $\|x^k - x\| < \delta$ for all $k > N_2$. Also since $\{f^k\}$ converges pointwise to f there is an integer $N_3 > 0$ such that $\|f^k(x) - f(x)\| < \epsilon/2$ for all $k > N_3$. Set $N = \max(N_1, N_2, N_3)$. Then for any $k > N$ it follows that

$$\|f^k(x^k) - f(x)\| \leq \|f^k(x^k) - f^k(x)\| + \|f^k(x) - f(x)\|$$

$$\leq \epsilon/2 + \epsilon/2$$

$$= \epsilon \ .$$

The first term is made small by the weak equicontinuity of $\{f^k\}$ and the second term is made small by the pointwise convergence of $\{f^k\}$ to f. $\|$

Lemma 6.2. Let $\{\mathcal{M}^k\}$ be a sequence of triangulations of R^m such that $\text{mesh}(\mathcal{M}^k) \to 0$ as $k \to \infty$. Suppose further that $\{f^k\}$ is a weakly equicontinuous sequence of functions. Let f^{kL} be the PL approximation to f^k induced by $\mathcal{M}^k$ (see Chapter 3). Then for any sequence of points $\{x^k\}$ and $\epsilon > 0$ there is an integer $N > 0$ such that $\|f^{kL}(x^k) - f^k(x^k)\| < \epsilon$ for all $k > N$.

Proof. The proof is done by showing that for large enough N, the vertices of the m-simplex containing x^k are sufficiently close to x^k to apply the properties of weak equicontinuity. Formally then let $\epsilon > 0$ be given. Let $\tau^k = \text{hull}(\{x^{0k}, \ldots, x^{mk}\})$ be an m-simplex in $\mathcal{M}^k$ containing x^k. Hence there are nonnegative multipliers $\lambda^{0k}, \ldots, \lambda^{mk}$ which sum to 1 such that $x^k = \sum_{j=0}^m \lambda^{jk}x^{jk}$. Let $\delta > 0$ and N_1 be chosen by the definition of weak equicontinuity of $\{f^k\}$. Choose N_2 such that $\|x^{jk} - x^k\| < \delta$ for each $j = 0,\ldots,m$ and $k > N_2$. This may be done since $\text{mesh}(\mathcal{M}^k) \to 0$ as $k \to \infty$. Setting $N = \max(N_1,N_2)$ it follows that for all $k > N$,

$$\|f^{kL}(x^k) - f^k(x^k)\| = \|f^{kL}(\sum_{j=0}^{m} \lambda^{jk}x^{jk}) - f^k(x^k)\|$$

$$= \|\sum_{j=0}^{m} \lambda^{jk}(f^k(x^{jk}) - f^k(x^k))\|$$

$$\leq \sum_{j=0}^{m} \lambda^{jk}\|f^k(x^{jk}) - f^k(x^k)\|$$

$$\leq \sum_{j=0}^{m} \lambda^{jk} \epsilon$$

$$= \epsilon \; . \qquad \|$$

Theorem 6.1. Let $\{f^k\}$ be a weakly equicontinuous sequence of functions which converges pointwise to f. Let $\{\mathcal{M}^k\}$ be a sequence of triangulation of R^m such that $\text{mesh}(\mathcal{M}^k) \to 0$ for $k \to \infty$. Also let f^{kL} be the PL approximation to f^k induced by $\mathcal{M}^k$. If $\{x^k\}$ is a sequence of fixed points of f^{kL} such that $\{x^k\} \to x$ for $k \in K$ (some subsequence), then x is a fixed point of f.

Proof. It will be shown that $\|f(x)-x\| = 0$ so let $\epsilon > 0$. Then for each $k \in K$,

$$\|f(x)-x\| \leq \|f(x)-f^k(x^k)\| + \|f^k(x^k) - f^{kL}(x^k)\| + \|f^{kL}(x^k)-x^k\| + \|x^k-x\| \; .$$

Each of these terms can be made less than $\epsilon/4$ for sufficiently large k. The first term can because of L.6.1 as can the second by L.6.2. The third term is 0 because x^k is the fixed point of f^{kL} for each $k \in K$. Finally, the fourth term can be made less than $\epsilon/4$ since $\{x^k\} \to x$ for $k \in K$ by hypothesis. The result now follows by letting $\epsilon \to 0$. $\|$

The next proposition shows that under certain circumstances a sequence of PL approximations to a uniformly continuous function can generate a weakly equicontinuous sequence of functions.

Proposition 6.1. Let $f: R^m \to R^m$ be uniformly continuous and let $\{\mathcal{M}^k\}$ be a sequence of triangulations of R^m such that $\text{mesh}(\mathcal{M}^k) \to 0$ for $k \to \infty$. Also let f^{kL} be the PL approximation to f induced by $\mathcal{M}^k$, then $\{f^{kL}\}$ is an equicontinuous sequence of functions.

Proof. Let $\epsilon > 0$. First choose $\delta > 0$ such that $\|x-y\| < \delta$ implies $\|f(x)-f(y)\| < \epsilon/3$ by the uniform continuity of f. Next choose N such that $\text{mesh}(\mathcal{M}^k) < \delta$ for all $k > N$. It will be shown that δ and N satisfy the definition of equicontinuity. To see this, let $k > N$ and $x, y \in R^m$ with $\|x-y\| < \delta$. Also let $\text{hull}(\{x^{0k}, \dots, x^{mk}\})$, $\text{hull}(\{y^{0k}, \dots, y^{mk}\})$ be simplexes in $\mathcal{M}^k$ containing x and y respectively. Hence there are nonnegative multipliers $\lambda^{0k}, \dots, \lambda^{mk}$ and $u^{0k}, \dots, u^{mk}$ which sum to 1 such that $x = \sum_{i=0}^m \lambda^{ik} x^{ik}$ and $y = \sum_{i=0}^m u^{ik} y^{ik}$. Consequently

$$\begin{aligned}
\|f^{kL}(x)-f^{kL}(y)\| &\leq \|f^{kL}(x)-f(x)\| + \|f(x)-f(y)\| + \|f(y)-f^{kL}(y)\| \\
&= \left\|\sum_{i=0}^{m} \lambda^{ik}(f(x^{ik})-f(x))\right\| + \|f(x)-f(y)\| + \left\|\sum_{i=0}^{m} u^{ik}(f(y)-f(y^{ik}))\right\| \\
&\leq \sum_{i=0}^{m} \lambda^{ik}\|f(x^{ik})-f(x)\| + \|f(x)-f(y)\| + \sum_{i=0}^{m} u^{ik}\|f(y)-f(y^{ik}))\| \\
&\leq \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \\
&= \epsilon . \quad \|
\end{aligned}$$

In particular, if f is a continuous function from a compact set into itself then f will be uniformly continuous and consequently a sequence of PL approximations to f will yield a weakly equicontinuous sequence of functions.

A theorem similar to T.6.1 for the point to set map case can be developed. For this it is necessary to have

Definition 6.2. Let $S^k: R^m \to (R^m)^*$ be a nonempty point to set map for each $k = 1,2,\dots$. Then $\{S^k\}$ is said to converge equicontinuously to the nonempty point to set map S if whenever

(1) $\{x^k\} \to x$,

(2) $z^k \in S^k(x^k)$ for all $k = 1,2,\dots,$ and

(3) $\{z^k\} \to z$ then $z \in S(x)$.

Theorem 6.2. Suppose $\{S^k\}$ is a sequence of nonempty point to set maps which converges equicontinuously to a usable point to set map S. Also suppose there is a compact set C with $\cup_k S^k(R^m) \subseteq C$. Let $\{\mathcal{M}^k\}$ be a sequence of triangulations of R^m with $\text{mesh}(\mathcal{M}^k) \to 0$ for $k \to \infty$ and let f^{kL} be the PL approximation to S^k induced by $\mathcal{M}^k$ with x^k a fixed point of f^{kL}. If x is a cluster point of $\{x^k\}$ then $x \in S(x)$.

Proof. The proof is done by showing that x may be expressed as a convex combination of points in $S(x)$. The result will then follow since $S(x)$ is convex. To begin with, x^k is a fixed point of f^{kL}. Let $\tau^k = \text{hull}(\{x^{0k}, \dots, x^{mk}\})$ be an m-simplex containing x^k. Hence there are nonnegative multipliers $\lambda^{0k}, \dots, \lambda^{mk}$ which sum to 1 such that $\sum_{i=0}^{m} \lambda^{ik} x^{ik} = x^k$. To say that x^k is a fixed point of f^{kL} means that there are points $z^{0k} \in S^k(x^{0k}), \ldots, z^{mk} \in S^k(x^{mk})$ for which

$$\sum_{i=0}^{m} \lambda^{ik} x^{ik} = \sum_{i=0}^{m} \lambda^{ik} z^{ik} .$$

One would like to be able to take limits in this equation therefore a subsequence K will be found for which $\{z^{ik}\}$, $\{x^{ik}\}$, $\{\lambda^{ik}\}$ all converge for $k \in K$. To do this it will be shown that $\{z^{ik}\}$, $\{x^{ik}\}$, and $\{\lambda^{ik}\}$ each lie in different but compact sets. The compact set containing $\{z^{ik}\}$ is C by hypothesis. The sequence $\{x^{ik}\}$ actually converges to x for each $i = 0, \ldots, m$. This is because $\text{mesh}(\mathcal{M}^k) \to 0$ for $k \to \infty$. Finally the $\{\lambda^{ik}\}$ lie inside a simplex. Hence there is a subsequence K along which $\{z^{ik}\} \to z^i$ for $k \in K$, $\{x^{ik}\} \to x$ for $k \in K$, and $\{\lambda^{ik}\} \to \lambda^i$ for $k \in K$ and this is true for each $i = 0, \ldots, m$. Now one may take limits in the above equation to yield

$$\sum_{i=0}^{m} \lambda^i x = \sum_{i=0}^{m} \lambda^i z^i$$

or equivalently

$$x = \sum_{i=0}^{m} \lambda^i z^i .$$

Hence x is a convex combination of $z^0, \ldots, z^m$. All that remains to be shown is that $z^i \in S(x)$ for each $i = 0, \ldots, m$, and this follows from D.6.2 since for each i,

(1) $\{x^{ik}\} \to x$ for $k \in K$,

(2) $z^{ik} \in S^k(x^{ik})$ for $k \in K$,

(3) $\{z^{ik}\} \to z^i$ for $k \in K$,

therefore $z^i \in S(x)$ for each $i = 0, \ldots, m$. ||

This concludes the approximation section, however, one word of caution is in order when implementing the idea. If the early approximations are not good the algorithm might become "trapped" in the wrong region and require a lot of additional work to get back to the correct answer, thus negating the savings. Hence one may wish to think of this as a "tail end" procedure depending of course on the specific approximation.

6.3. Various Modified Newton Methods.

In this section the Simplified Newton Method (SNM) and the Quasi-Newton Methods (QNM) are described as possible alternatives to using Newton's Method (NM) for solving nonlinear systems of equations.

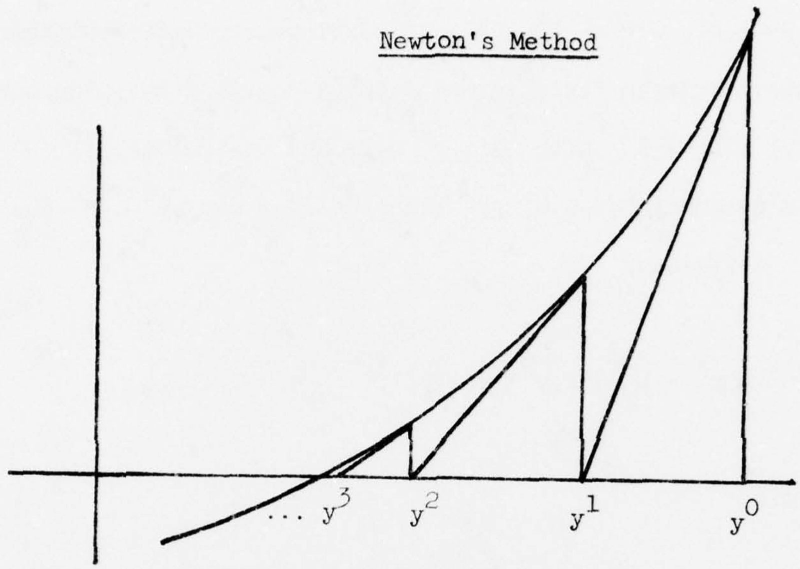
Recall how NM will work. For a given point $x \in R^m$ it will take a starting point y^0 and generate the sequence $\{y^k\}$ defined by

$$y^{k+1} = y^k - D_y^{-1}G(x,y^k)\ G(x,y^k) \quad \text{for} \quad k = 0,1,\ldots .$$

Notice that it is necessary to recompute $D_y^{-1}G(x,y^k)$ at each iteration. The idea behind SNM is to compute the matrix $W = D_y^{-1}G(x,y^0)$ once and for all and then to generate the sequence $\{y^k\}$ defined by

$$y^{k+1} = y^k - WG(x,y^k) \quad \text{for} \quad k = 0,1,\ldots .$$

Figure 6.1 shows the difference between these two methods in the 1-dimensional case. The next theorem gives conditions under which SNM may be expected to converge.


Newton's Method
... y^3
y^2
y^1
y^0

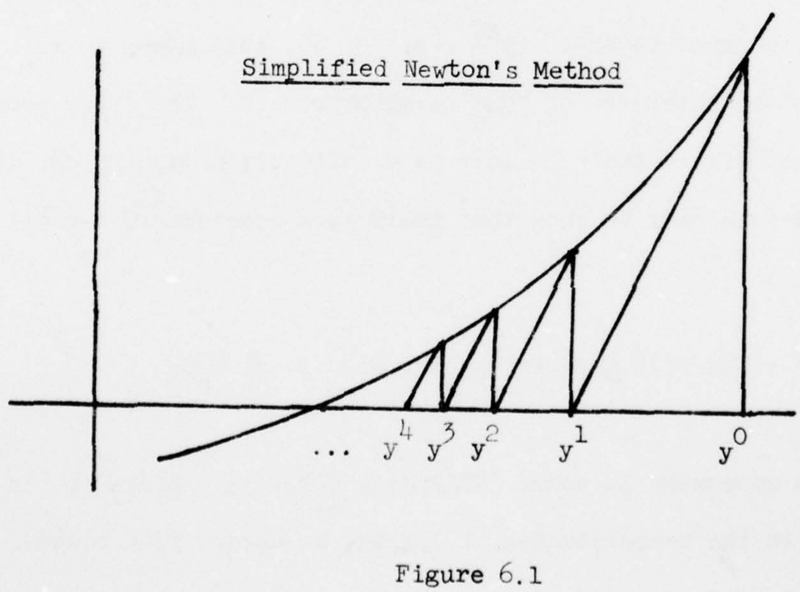

Simplified Newton's Method
... y^4
y^3
y^2
y^1
y^0

Figure 6.1

Theorem 6.3. Suppose $G: R^m \times R^n \to R^n$ is continuously differentiable in the y-coordinates for each fixed value of the x-coordinates. Suppose also that $(x,y) \in R^m \times R^n$ with $G(x,y) = 0$, and $\det(D_y G(x,y)) \neq 0$. Then there is a neighborhood 0 of y such that for all $y^0 \in 0$ the sequence $\{y^k\}$ defined by

$$y^{k+1} = y^k - D_y^{-1}G(x,y^0)\ G(x,y^k), \qquad k = 0,1,\ldots \ .$$

converges to y.

Proof. The proof is done, by showing that there is a neighborhood 0 of y such that for each $y^0 \in 0$ the function $L: R^m \times c|(0) \to c|(0)$ defined by $L(x,z) = z - D_y^{-1}G(x,y^0)\ G(x,z)$ is a contraction mapping and hence has a unique fixed point namely y. Consequently, since $y^{k+1} = L(x,y^k)$, it must be that $\{y^k\} \to y$. So all that needs to be demonstrated is the existence of this neighborhood 0. The first property that 0 must satisfy is that L must be a contraction mapping on $c|(0)$. Thus it will be necessary to show that there is a constant $0 < c < 1$ such that

$$\|L(x,z) - L(x,z')\| \leq c\|z-z'\| \quad \text{for all} \quad z,\ z' \in 0.$$

Therefore it is necessary to bound $\|L(x,z) - L(x,z')\|$. Since L is differentiable in the y-coordinates, T.1.1 may be used. This bound specifies that

$\|L(x,z)-L(x,z')\| \leq \sup\{\|D_yL(x,z+\lambda(z'-z))\|/\lambda \in [0,1]\}\|z-z'\|$. Hence O will be chosen so that

$$\sup\{\|D_yL(x,\ z+\lambda(z'-z))\|/\lambda \in [0,1]\} \leq c$$

for some c between 0 and 1. From the definition of L,

$$\begin{aligned}
\|D_yL(x,z)\| &= \|I - D_y^{-1}G(x,y^0)D_yG(x,z)\| \\
&\leq \|D_y^{-1}G(x,y)\ D_yG(x,y) - D_y^{-1}G(x,y^0)\ D_yG(x,y)\| \\
&\quad + \|D_y^{-1}G(x,y^0)\ D_yG(x,y) - D_y^{-1}G(x,y^0)\ D_yG(x,z)\| \\
&\leq \|D_yG(x,y)\|\ \|D_y^{-1}G(x,y) - D_y^{-1}G(x,y^0)\| \\
&\quad + \|D_y^{-1}G(x,y^0)\|\ \|D_yG(x,y) - D_yG(x,z)\| .
\end{aligned}$$

The set O will be chosen to make each of these final two terms $\leq c/2$, and it will follow that

$$\sup\{\|D_yL(x,z+\lambda(z'-z))\|/\lambda \in [0,1]\} \leq c .$$

The first term can be made less than $c/2$ by using the continuity of $D_yG(x,y)$ to choose a $\delta_0 > 0$ such that

$$\|D_y^{-1}G(x,y) - D_y^{-1}G(x,y^0)\| \leq c/(2\|D_yG(x,y)\|)$$

whenever $\|y-y^0\| < \delta_0$. Also a $\delta_1 < \delta_0$ may be chosen together with a constant e such that

$$\|D_y^{-1}G(x,y^0)\| \leq e$$

whenever $\|y-y^0\| \leq \delta_1$. Finally there is a $\delta_2 < \delta_1$ such that

$$\|D_yG(x,y) - D_yG(x,z)\| \leq c/(2e)$$

whenever $\|z-y\| \leq \delta_2$. Defining $0 = B(y,\delta_2)$ does the trick. All that remains to be verified is that L in fact maps $R^m \times c|(0)$ into $c|(0)$. To this end let $z \in c|(0)$. It msy be shown that $\|L(x,z) - y\| \leq \delta_2$. By T.1.1 and the construction of 0,

$$\begin{aligned} \|L(x,y)-z\| &= \|L(x,z) - L(x,y)\| \\ &\leq \sup\{\|D_yL(x, z + \lambda(y-z))\| / \lambda \in [0,1]\} \|z-y\| \\ &\leq c\|z-y\| \\ &\leq \|z-y\| \\ &\leq \delta_2. \end{aligned}$$

This completes the proof. ||

Other variants of NM are the Quasi-Newton Methods (QNM). Basically these methods differ from Newton's Method in that they do not require a direct computation of $D_y^{-1}G(x,y^k)$. Instead this information is approximated by local data. The approximation is continually updated throughout the solution procedure. Davidon [10] and Fletcher and Powell [18] were amongst the first to study this approach. Only the (n+1)-point sequential secant method predated their work.

6.4. Use of Upper and Lower Bounds.

In this section a different point to set mapping is designed to solve P.4.2. The difference is that in the new formulation it will not be necessary to compute $t(x) = \max\{q_i(x)/1 \leq i \leq \ell\}$. Instead it will be possible to check each constraint for feasibility and stop as soon as the first infeasible constraint is detected. To do this it will be necessary to assume A.4.1. Then it is possible to define the following collection of convex functions $t^i : R^m \to R^1$ by $t^i(x) = \max\{q_j(x)/1 \leq j \leq i\}$ for each $i = 1, \ldots, \ell$, and the following usable point to set maps $T^i : R^m \to (R^m)^*$ by $T^i(x) = x - \partial t^i(x)$ for each $i = 1, \ldots, \ell$. Letting $S^\ell : R^m \to (R^m)^*$ by $S^\ell(x) = x - \partial p(x)$ then one may inductively define $S^{i-1} : R^m \to (R^m)^*$ for $i = \ell, \ell-1, \ldots, 1$ by

$$S^{i-1}(x) = \begin{cases} S^i(x) & \text{if } t^i(x) < 0 \\ \text{hull}(S^i(x) \cup T^i(x)) & \text{if } t^i(x) = 0 \\ T^i(x) & \text{if } t^i(x) > 0 . \end{cases}$$

S^0 is the point to set map whose fixed point will be shown to solve P.4.2. First however it is necessary to establish that S^0 is usable.

<u>Theorem 6.4</u>. If A.4.1 holds and if $\{x/t^\ell(x) \leq 0\} \subseteq \text{int}(\text{dom } p)$ then S^0 is usable.

<u>Proof</u>. S^ℓ is usable by T.1.4. By T.1.5 one can conclude that S^{i-1} is usable. Iterating backwards finally yields S^0 is usable. ||

The relationship between fixed points of S^0 and solutions to P.4.2 is established in

Theorem 6.5. If $u = \inf\{t^\ell(z)/z \in R^m\} < 0$ then $x \in S^0(x)$ iff x solves P.4.2.

Proof. Assume first that $x \in S^0(x)$. It is necessary to show that (a) x is feasible and (b) if z is any other feasible point then $p(z) \geq p(x)$. The feasibility of x will be established by contradiction, so suppose $x \in S^0(x)$ is infeasible. It will be shown that $t^\ell(z) \geq 0$ for all z and this will contradict the fact that $u < 0$. Since $x \in S^0(x)$ there is a $\lambda^1 \in [0,1]$, $x^1 \in S^1(x)$, $z^1 \in T^1(x)$ such that

$$x = \lambda^1 x^1 + (1-\lambda^1) z^1 .$$

Since $z^1 \in T^1(x)$ there is a $b^1 \in \partial t^1(x)$ with $z^1 = x-b^1$, hence,

$$x = \lambda^1 x^1 + (1-\lambda^1)(x-b^1) .$$

Note that if $t^1(x) > 0$ then $\lambda^1 = 0$ and if $t^1(x) < 0$ then $\lambda^1 = 1$. Since $x^1 \in S^1(x)$ there is a $\lambda^2 \in [0,1]$, $x^2 \in S^2(x)$, $z^2 \in T^2(x)$ such that

$$x^1 = \lambda^2 x^2 + (1-\lambda^2) z^2 .$$

Since $z^2 \in T^2(x)$ there is a $b^2 \in \partial t^2(x)$ with $z^2 = x-b^2$, hence

$$x^1 = \lambda^2 x^2 + (1-\lambda^2)(x-b^2).$$

Substituting this into the expression for x yields

$$x = \lambda^1\lambda^2x^2 + \lambda^1(1-\lambda^2)x - \lambda^1(1-\lambda^2)b^2 + (1-\lambda^1)x - (1-\lambda^1)b^1 .$$

Note again that if $t^2(x) > 0$ then $\lambda^2 = 0$ and if $t^2(x) < 0$ then $\lambda^2 = 1$. Continuing in this manner yields that for each $i = 1,\ldots, \ell$ there is a $\lambda^i \in [0,1]$, $b^i \in \partial t^i(x)$ together with a $c \in \partial p(x)$ such that

$$x = \lambda'(x-c) + (1-\lambda')x - \sum_{i=1}^{\ell} e^i b^i$$

where

$$\lambda' = \lambda^1 \cdot \lambda^2 \cdot \ldots \cdot \lambda^\ell \quad \text{and} \quad e^i = (1-\lambda^i)\left(\sum_{j=1}^{i-1} \lambda^j\right) .$$

(It is understood that $e^1 = (1-\lambda^1)$.) Simplifying this one has

(a) $$\lambda' c + \sum_{i=1}^{\ell} e^i b^i = 0$$

and as noted above

(b) $t^i(x) > 0$ implies $\lambda^i = 0$, for each $i = 1,\ldots,\ell$; and

(c) $t^i(x) < 0$ implies $\lambda^i = 1$, for each $i = 1,\ldots,\ell$.

Recall that x is assumed to be infeasible hence for some k, $t^k(x) > 0$ and consequently by (b), $\lambda' = 0$. Thus (a) reduces to (a') $\sum_{i=1}^{\ell} e^i b^i = 0$. To establish the desired contradiction it is necessary to use the convexity of t^i and the fact that $b^i \in \partial t^i(x)$ so

$$t^i(z) \geq t^i(x) + \langle b^i, z-x \rangle \qquad \text{for all } z \in R^m .$$

Multiplying both sides of this by the nonnegative number e^i and summing over i yields

$$\text{(d)} \quad \sum_{i=1}^{\ell} e^i t^i(z) \geq \sum_{i=1}^{\ell} e^i t^i(x) + \langle \sum_{i=1}^{\ell} e^i b^i, z-x \rangle \quad \text{for all } z \in R^m$$

$$= \sum_{i=1}^{\ell} e^i t^i(x) ,$$

The last equality being justified by (a'). Note that $t^i(x)$ is non-decreasing in i so

$$\text{(e)} \quad \sum_{i=1}^{\ell} e^i t^i(z) \leq (\sum_{i=1}^{\ell} e^i)\, t^{\ell}(z) \quad \text{for all } z \in R^m .$$

Also note that

$$\text{(f)} \quad \sum_{i=1}^{\ell} e^i t^i(x) \geq 0$$

because $e^i \geq 0$ and if $t^i(x) < 0$ then by (c) $e^i = 0$. Combining (e) and (f) into (d) yields

$$(\sum_{i=1}^{\ell} e^i)\, t^{\ell}(z) \geq 0 \quad \text{for each } z \in R^m.$$

Finally observe that $\sum_{i=1}^{\ell} e^i = (1-\lambda') + \lambda' = 1$ and hence $t^{\ell}(z) \geq 0$ for each $z \in R^m$. This is the long awaited for contradiction. Thus far it has been shown that x is feasible and consequently that $\lambda' > 0$. Now let $z \in R^m$ be any other feasible point for P.4.2. It must be shown that $p(z) \geq p(x)$. This will follow from the convexity of p and the fact that $c \in \partial p(x)$ since

$$p(z) \geq p(x) + \langle c, z-x \rangle .$$

Hence all that needs to be shown is that $\langle c, z-x \rangle \geq 0$, and this will be derived from the feasibility of z and the convexity of t^i. Since $b^i \in \partial t^i(x)$,

(g) $\quad t^i(z) \geq t^i(x) + \langle b^i, z-x \rangle$ for each $i = 1, \ldots, \ell$.

Multiplying both sides of (g) by the nonnegative scalars e^i/λ' and summing over i yields

(h) $\quad \sum_{i=1}^{\ell} (e^i/\lambda')\, t^i(z) \geq \sum_{i=1}^{\ell} (e^i/\lambda')\, t^i(x) + \langle \sum_{i=1}^{\ell} (e^i/\lambda^i) b_i, z-x \rangle$

and from (a),

(h') $\quad \sum_{i=1}^{\ell} (e^i/\lambda')\, t^i(z) \geq \sum_{i=1}^{\ell} (e^i/\lambda')\, t^i(x) - \langle c, z-x \rangle$.

On the right side of (h') note that $\sum_{i=1}^{\ell} (e^i/\lambda')\, t^i(x) = 0$ since x is feasible and since $t^i(x) < 0$ implies $e^i = 0$ by (c). On the left side of (h') note that $\sum_{i=1}^{\ell} (e^i/\lambda')\, t^i(z) \leq \sum_{i=1}^{\ell} (e^i/\lambda')\, t^{\ell}(z)$ by the monotonicity of t^i in i. Combining these two into (h') yields

(i) $\sum_{i=1}^{\ell}(e^{i}/\lambda')\ t^{\ell}(z) \geq -\langle c, (z-x)\rangle$.

Since z is a feasible point, $t^{\ell}(z) \leq 0$ and hence $\langle c, z-x\rangle \geq 0$ as desired. This proves the necessary part of the theorem. To establish the sufficiency, assume x solves P.4.2. It will be shown that $x \in S^{\ell-1}(x)$ and an argument is given to show that $x \in S^{\ell-2}(x)$. This argument may be repeated to establish that $x \in S^{0}(x)$. Note that $x \in S^{\ell-1}(x)$ by T.1.5. To see that $x \in S^{\ell-2}(x)$ consider

Case 1: $t^{\ell-1}(x) < 0$.
In this case, $S^{\ell-2}(x) = S^{\ell-1}(x)$ and therefore $x \in S^{\ell-2}(x)$.

Case 2: $t^{\ell-1}(x) = 0$.
In this case $S^{\ell-2}(x) = \text{hull}(S^{\ell-1}(x) \cup T^{\ell-1}(x))$. But $x \in S^{\ell-1}(x)$ so in particular, $x \in S^{\ell-2}(x)$. Since x is feasible for P.4.2 these are the only two possible cases and that establishes the proof. ||

At this point one would hope to be able to apply the algorithm of Chapter 3 to this point to set map. Unfortunately there is no guarantee that a linear approximate fixed point of S^{0} will be computed in a finite number of steps. Merrill [38] observed this difficulty and he constructed an example where the constraints were all convex as was the objective function yet the algorithm failed because it did not find a linear approximate fixed point in a finite number of steps. A way around this difficulty is to ensure that the algorithm never searches for a fixed point of the PL approximation outside of some compact set. A very natural choice would be the hyperrectangle defined by the upper and lower bounds since no solution to P.4.2 can lie outside this region anyway. The only problem is that P.4.2 may have some variables which have no bounds. In this case it is necessary to impose arbitrary upper and lower bounds. Each time a new linear approximate fixed point is computed, the artificial bounds will be expanded. This is the essence of

<u>Theorem 6.5</u>. Let $\{u^k\}$ and $\{v^k\}$ be sequences of vectors in R^m with $u^k < v^k$ and $\{u^k\} \to (-\infty, -\infty, \dots, -\infty)$ and $\{v^k\} \to (+\infty, +\infty, \dots, +\infty)$ as $k \to \infty$. Define a sequence of problems

$$
\begin{aligned}
P^k: \quad & \min p(x) \\
& \text{s.t. } q(x) \leq 0 \\
& \qquad u^k \leq x \leq v^k \\
& \qquad x \in R^m
\end{aligned}
$$

Then x solves P.4.2 iff there is an N such that x solves P^k for $k > N$.

<u>Proof</u>. Suppose first that x solves P.4.2. Choose N such that $k > N$ implies $u^k \leq x \leq v^k$. If z is any other feasible point of P^k for $k > N$, then $p(z) \geq p(x)$ since x solves P.4.2. This establishes the necessary part of the theorem. Suppose now that x solves P^k for all $k > N$. It must be shown that x solves P.4.2. Clearly x is

AD-A053 439 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/1

DECOMPOSITION IN FIXED POINT COMPUTATION.(U)

DEC 77 D SOLOW DAAG-29-74-C-0032

UNCLASSIFIED SOL-77-32 NL

2 OF 2

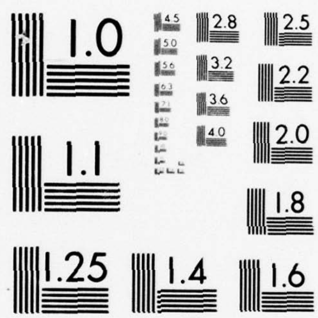
AD A053439

END

DATE FILMED

6-78

DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

feasible for P.4.2 since $q(x) \leq 0$. If z is any other feasible point for P.4.2, then choose $k > N$ such that z is feasible for P^k. From the fact that x solves P^k, $p(x) \leq p(z)$. This establishes that x is optimal for P.4.2. ||

An additional advantage to using a hyperrectangle is that it may be easily triangulated. The philosophy here is that "if there is no reason to search outside a given region then don't." Also one would like the ability to control the "size" of these simplexes. To show precisely how this is done let $u, v \in R^m$ with u strictly less than v in all coordinates (u will be the lower bounds and v the upper bounds). Also let $N^1, \ldots, N^m$ be positive integers. Set $M = X_{i=1}^m [0, N^i]$ and $C = X_{i=1}^m [u_i, v_i]$. The idea is to create a linear homeomorphism, L, from M into C. Then any triangulation of M yields one of C by applying L to each simplex. Michael Todd [52] has developed a triangulation of R^m which when restricted to M yields a triangulation of M. This combined with the mapping L will provide the desired triangulation of C. It remains only to specify the mapping $L: M \to C$, so let $(i_1, \ldots, i_m) \in M$ then define

$$L(i_1, \ldots, i_m) = u + W(i_1, \ldots, i_m)$$

where

$$W = \begin{bmatrix} \frac{(v_1-u_1)}{N^1} & 0 & \cdots & 0 \\ 0 & \frac{(v_2-u_2)}{N^2} & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & \cdots & 0 & \frac{(v_m-u_m)}{N^m} \end{bmatrix}$$

This has been implemented into the computer code and choices for $N^1, \ldots, N^m$ have been left to the user's discretion.

6.5. Other Methods and Future Research.

One of the more effective acceleration techniques is that of Saigal [44,45]. It applies to the special case of P.4.2 in which there are no inequality constraints, that is to say, unconstrained optimization. Furthermore the hypotheses require that p be a strongly convex twice continuously differentiable function. In this case he was able to obtain an increased convergence rate by increasing the rate at which the mesh of the triangulations goes to zero.

Since the area of acceleration techniques appears to be the most important, several unsolved problems will be presented in the hopes that if they can be solved, further improvements will be obtained.

For example, in Chapter 3, it was mentioned that various researchers have developed computerizable triangulations of R^m and of the unit simplex, and with the results of section 6.4 it is now possible to triangulate a hyperrectangle. The next generalization would be to develop such a triangulation of a compact polyhedral set. This would enable one to solve a linear programming problem in a finite number of steps. Another interesting possibility would be to find a sequence of linear transformations, which, when applied to the original triangulation of R^m yields a new triangulation of R^m with perhaps some beneficial new properties.

Another area which could use some work is the way in which the basic and nonbasic variables are chosen. This problem has not been dealt with here and it can only be said that in the test problems of Chapter 7 a proper choice was always evident. In general one will not be that lucky and it will be necessary to find a method for doing this. The choice can be critical in the success of the algorithm. The following example shows that a poor choice of the basic and nonbasic variables can cause the algorithm to fail.

Example 6.1. Consider the problem

$$\begin{aligned} \min \quad & y - x \\ \text{s.t.} \quad & y - x^2 = 0 \\ & x, y \in R^1 . \end{aligned}$$

The solution to the problem is $x = 1/2$, $y = 1/4$ and the algorithm will compute this answer provided the x variable is chosen as the nonbasic variable, whereas if y is so chosen, the algorithm can be caused to converge to $x = y = 0$ by choosing an initial point $x' = -1$, $y' = 1$ with $G(x',y') = 0$.

CHAPTER 7

COMPUTER RESULTS

7.1. Designing the Test.

All of the previous work has been used to develop an algorithm for solving P.4.1 under rather hypothetical conditions. Although convergence has been theoretically established in Chapter 4 the ultimate value of this (or for that matter any other) approach can only be determined by its actual performance. Therefore seventeen test problems have been solved and the results are compared against those obtained from GRG (Generalized Reduced Gradient Method) proposed originally by Abadie [1]), which, to the author's knowledge, is the best commercial code for solving P.4.1.

The first difficulty in designing such a test is to determine a measure of computational efficiency. Some items which should be considered are robustness, "cost" to compute a solution, accuracy of the solution with respect to the optimal objective value and with respect to the optimal solution vector, and ease of implementation. Of these, robustness and overall expense are often considered to be the most important. Robustness is the ability to handle a wide variety of problems under various starting conditions. For this reason seventeen distinct problems are presented and solved, sometimes from different starting points.

Having settled on costs as one of the measures of efficiency the next task is to determine how it is to be calculated. One possibility is to define costs as the number of function evaluations. The reasoning is that very often this will be the most expensive operation. There are several drawbacks to using this statistic as the sole measure of costs. One problem is that it totally ignores the amount of work performed between function evaluations. Also, function evaluations in one method may be quite different from those of another method. For example, the fixed point code will require a simplex pivot in R^m after each gradient evaluation, but it will never evaluate the objective function. Furthermore, due to some of the techniques of Chapter 6, not all gradient evaluations will require a matrix inverse. On the other hand, with GRG, the objective function is evaluated many times but no pivots need to be performed. All these different statistics are reported in the ensuing tables; however, the conclusion is that it is necessary to let the computer determine the total costs via the CPU time.

Even this is very unsatisfactory in that the way the algorithms are programmed, the access of data and the amount of printing can radically affect the CPU time. Several precautions were taken to reduce some of these variances. To begin with, both codes were run on the same computer (IBM 370/165) and on identical partitions of core. Both were compiled under Fortran H, OPT2 and CPU time was measured from the very first executable statement to the very last. The fixed point code was programmed by the author and the GRG code was obtained from Leon Lasdon and Arvind Jain through the Systems Optimization Laboratory at Stanford University.

The final step was to find seventeen test problems. One of the major difficulties here was that many of the problems in the literature had only inequality constraints. Any problem of this kind was discarded immediately since the entire objective was to see how this algorithm would perform under equality constraints. Since problems of this nature are, in general, very difficult to come by they will be printed here along with their sources, known solutions, and various starting points in the hopes that other researchers will be able to use them.

7.2. Tables of Results and Conclusions.

This section presents the results of the fixed point code and GRG on the seventeen problems in Section 7.3. Table 1 gives a summary of the characteristics of the problems and Tables 2 and 3 give a summary of the fixed point and GRG methods respectively. In Table 2, the letter following a number refers to a different starting point as shown in the problem description. Also in Table 2, the number of gradient calls requiring a matrix inverse is reported whereas Table 3 reports all gradient calls. GRG appears to be somewhat faster in solving problems 1, 7, 9, 12, 13 and 15, whereas the fixed point code seems slightly faster on problems 2, 3, 4, 5, 6, 8, 10, 11 and 14. At first glance there is reason to suspect that the dimension of the problem is a key factor. A possible explanation for this is that as the dimension goes up, the number of simplexes traversed by the fixed point algorithm goes up greatly. In order to test this possibility a forty dimensional problem was invented (Problem 16) and GRG proved to be approximately eleven times faster in solving it. Note that this problem has no inequality constraints. Since inequality constraints add to the complexity, Problem 16 was modified by adding upper bounds. This formed Problem 17. An interesting result was that the fixed point code required almost twice as many pivots (and consequently almost twice as much time) to solve Problem 17 as it did to solve Problem 16. GRG, on the other hand, actually took less time to solve Problem 17 than it did to solve Problem 16. This opens up another area for future investigation.

Nonetheless, the conclusion from these tests is that when the dimension of the original problem is reduced sufficiently by the equality constraints, the fixed point approach appears to be more effective than GRG.

TABLE 1

PROBLEM DESCRIPTION

Problem Number	Number of Variables	Number of Equality Constraints	Lower Bounds	Upper Bounds	Nonlinearity of Objective Function	Nonlinearity of Constraints
1	2	1	None	None	Quadratic	Quadratic
2	3	1	None	None	Mild	Mild
3	3	2	None	None	Strong	Quadratic
4	3	2	None	None	Cubic	Quadratic
5	3	2	None	None	Strong	Strong
6	4	2	None	None	Strong	Strong
7	5	2	None	None	Strong	Strong
8	5	2	None	None	Strong	Strong
9	5	2	All	All	Mild	Mild
10	5	3	None	None	Strong	Strong
11	5	3	None	None	Strong	Strong
12	6	4	All	All	Discontinuous	Strong
13	10	3	All	None	Strong	Linear
14	11	6	All	None	Strong	Linear
15	16	8	All	All	Quadratic	Linear
16	40	20	None	None	Quadratic	Mild
17	40	20	All	All	Quadratic	Mild

TABLE 2

FIXED POINT RESULTS

Problem Number *	Objective Value Obtained	Number of Newton Calls	Number of Newton Iterations	Number of Gradient Calls**	Number of Pivots Required	CPU Time (in secs)
1	1.394	24	1	6	24	.07997
2	.000005	41	40	9	42	.11625
3(a)	961.717	20	59	9	26	.09547
3(b)	961.717	20	59	9	26	.09547
4	117.062	16	23	16	16	.08295
5	.1655	33	19	33	33	.10946
6	-4.4969	52	106	29	63	.14749
7	-2500.57	40	60	40	44	.13192
8	- 210.024	86	132	26	107	.20190
9	-30665.3	109	71	34	165	.20931
10(a)	.05395	28	38	9	29	.11132
10(b)	.05395	26	50	6	28	.10978
11(a)	.02932	40	39	24	44	.13036
11(b)	.02932	37	27	21	38	.12151
11(c)	27.552	44	56	28	49	.13700
11(d)	27.552	44	56	28	49	.13700
11(e)	27.552	44	56	28	49	.13700
12	8827.6	58	71	40	67	.23425
13	-47.760	1	1	1	385	.56465
14	.000120	1	1	1	115	.28489
15	244.9	1	1	1	967	1.99
16	.72842	427	390	265	577	16.01778
17	.73027	785	717	785	1084	28.0649

*The letters in parentheses refer to the different starting points.

**Only those gradient calls requiring a matrix inverse are reported.

TABLE 3

GRG RESULTS

Problem Number *	Objective Value Obtained	Number of Newton Calls	Number of Newton Iterations	Number of Gradient Calls	Number of Objective Function Calls	CPU Time (in secs)
1	1.39336	13	7	4	25	.0715
2	0.00000	47	86	9	134	.1139
3(a)	961.715	30	61	9	101	.11429
3(b)	961.715	40	86	10	135	.12521
4	117.056	28	21	6	50	.09472
5	.1655	13	37	4	51	.07924
6	-4.4969	51	126	14	179	.16045
7	-2500.63	29	34	8	65	.1284
8	-210.408	73	175	15	253	.2036
9	-30655.5	7	7	6	16	.0874
10(a)	.053948	24	42	8	68	.11519
10(b)	**	--	--	--	--	--
11(a)	.02931	34	76	10	113	.13434
11(b)	.02931	29	51	9	84	.12562
11(c)	44.022	44	30	10	78	.13835
11(d)	27.872	48	111	11	160	.15617
11(e)	607.017	76	152	15	239	.20074
12	8827.6	7	5	5	14	.08870
13	-47.760	94	1	20	101	.32137
14	**	--	--	--	--	--
15	244.9	156	1	37	162	.92266
16	.72839	56	73	130	11	1.4873
17	.73017	3	13	5	25	.4072

*The letters in parentheses refer to the different starting points.

**GRG failed to obtain correct solution (could be user error).

7.3. The Test Problems.

Since equality constrained optimization problems are, in general, difficult to find in the literature, the seventeen problems which were used for the comparative tests are documented here along with their sources, known solutions and suggested starting points.

Problem 1.

$$\min \quad P(x_1,x_2) = (x_1-2)^2 + (x_2-1)^2$$

$$\text{s.t.} \quad x_1 - 2x_2 + 1 = 0$$

$$\frac{x_1^2}{4} + x_2^2 - 1 \leq 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (.823, \ .911)$$

$$P(x') = 1.393$$

Suggested Starting Point.

$$x = (2,2)$$

Source. Himmelblau [26].

Problem 2.

$$\min \quad P(x_1,x_2,x_3) = (x_1-x_2)^2 + (x_2-x_3)^4$$

$$\text{s.t.} \quad x_1 + x_1x_2^2 + x_3^4 - 3 = 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (1,1,1)$$

$$P(x') = 0$$

Suggested Starting Point.

$$x = (2.4, \ .5, \ 0)$$

Source. Avriel [2].

Problem 3.

$$\min \quad P(x_1,x_2,x_3) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

$$\text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 25 = 0$$

$$8x_1 + 14x_2 + 7x_3 - 56 = 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (3.51212, \ .216988, \ 3.55217)$$

$$P(x') = 961.715$$

Suggested Starting Points.

(a) $x = (10, \ 10, \ 10)$

(b) $x = (-5, \ -10, \ 5)$

Source. Margaret Wright [55].

Problem 4.

$$\min \quad P(x_1,x_2,x_3) = -(x_1 + x_2 + x_3 - 7)^3$$

$$\text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 - 2 = 0$$

$$x_2 - \exp(x_1) = 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (.17906,\ 1.1961,\ .7330)$$

$$P(x') = 117.062$$

Suggested Starting Point.

$$x = (0,1,1)$$

Source. Richard Asmuth--Private Communication.

Problem 5.

$$\min \quad P(x_1,x_2,x_3) = \exp(x_1 x_2 - x_3^2)$$

$$\text{s.t.} \quad x_1^2 + x_3^4 - 2 = 0$$

$$x_1 x_2 - x_2^3 + x_3 = 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (-.682494,\ .820716,\ 1.11294)$$

$$P(x') = .1655032$$

Suggested Starting Point.

$$x = (-1,1,1)$$

Source. Richard McCord--Private Communication.

Problem 6.

$$\min \quad P(x_1,x_2,x_3,x_4) = -x_1^2x_4 + (x_1-1)^4 + (x_2-x_3)^4 + (x_3-1)^2$$

$$\text{s.t.} \quad x_1x_4^2 + \sin(x_4 - x_3) - 4 = 0$$

$$x_2^2 + x_3^2x_4^4 \qquad -10 = 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (2.033936,\ 1.591623,\ 1.392401,\ 1.400879)$$

$$P(x') = -4.496925$$

Suggested Starting Point.

$$x = (3.159,\ 3.162,\ 0,1)$$

Source. Richard McCord--Private Communication.

Problem 7.

$$\min \quad P(x_1,x_2,x_3,x_4,x_5) = 10x_1x_4-6x_3x_2^2+x_2x_1^3+9\sin(x_5-x_3) + x_5^4x_4^2x_2^3$$

$$\text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 20 = 0$$

$$x_1^2x_3 + x_5x_4 \qquad + 2 = 0$$

$$5 - x_2^2x_4 - 10x_1x_5 \qquad \leq 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (1.47963,\ -2.63661,\ 1.05467,\ -1.61151,\ 2.67388)$$

$$P(x') = -2500.55$$

Suggested Starting Point.

$$x = (1.091,\ -3.174,\ 1.214,\ -1.614,\ 2.134)$$

Source. Modified from Himmelblau [26].

Problem 8.

$$\min \quad P(x_1,x_2,x_3,x_4,x_5) = 10x_1x_4 - 6x_3x_2^2 + x_2x_1^3 + 9\sin(x_5-x_3) + x_5^4x_4^2x_2^3$$

$$\text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 20 = 0$$

$$x_2^2x_4 + 10x_1x_5 - 5 = 0$$

$$-x_1^2x_3 - x_5x_4 - 2 \leq 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (-.0814522,\ 3.69238,\ 2.48741,\ .377134,\ .173983)$$

or

$$x' = (.0320,\ 3.6914,\ 2.4807,\ .35993,\ .29777)$$

$$P(x') = -210.024$$

Suggested Starting Point.

$$x = (1,1,1,1,1)$$

Source. Modified from Himmelblau [26].

Problem 9.

$$\min \quad P(x_1,x_2,x_3,x_4,x_5) = 5.3578547x_3^2 + .8356891x_1x_5 + 37.293239x_1 - 40792.141$$

$$\text{s.t.} \quad 85.334407 + .0056858x_2x_5 + .0006262x_1x_4 - .0022053x_3x_5 - 92 = 0$$

$$9.300961 + .0047026x_3x_5 + .0012547x_1x_3 + .0019085x_3x_4 - 20 = 0$$

$$80.51249 + .0071317x_2x_5 + .0029955x_1x_2 + .0021813x_3^2 - 100 \leq 0$$

$$-80.51249 - .0071317x_2x_5 - .0029955x_1x_2 - .0021813x_3^2 + 90 \leq 0$$

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_3 \leq 45$$

$$27 \leq x_4 \leq 45$$

$$27 \leq x_5 \leq 45$$

Optimal Solution and Optimal Objective Value.

$$x' = (78, 33, 29.995, 45, 36.776)$$

$$P(x') = -30665.5$$

Suggested Starting Point

$$x = (78.62, 33.44, 31.07, 44.18, 35.22)$$

Source. Modified from Colville [8].

Problem 10.

$$\min \quad P(x_1, x_2, x_3, x_4, x_5) = \exp(x_1 x_2 x_3 x_4 x_5)$$

$$\text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$x_2 x_3 - 5 x_4 x_5 = 0$$

$$x_1^3 + x_2^3 + 1 = 0$$

Optimal Solution and Optimal Objective Value.

$$x' = (-1.71714, 1.59571, 1.82725, -.763643, -.763643)$$

$$P(x') = .0539499$$

Suggested Starting Points.

(a) $x = (-2, 2, 2, -1, -1)$

(b) $x = (-1, -1, -1, -1, -1)$

Source. Powell [42] also Margaret Wright [55].

Problem 11.

$$\min \quad P(x_1,x_2,x_3,x_4,x_5) = (x_1-1)^2 + (x_1-x_2)^2 + (x_2-x_3)^3 + (x_3-x_4)^4 + (x_4-x_5)^4$$

$$\text{s.t.} \quad x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0$$

$$x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0$$

$$x_1 x_5 - 2 = 0$$

Optimal Solution and Optimal Objective Values.

$x' = (1.11663, 1.22044, 1.53779, 1.97277, 1.79110)$

$P(x') = .029318$

$x' = (-2.79807, 3.00414, .205371, 3.87474, -.716623)$

$P(x') = 607.036$

$x' = (-1.27305, 2.41035, 1.19486, -.154239, -1.57103)$

$P(x') = 27.8719$

$x' = (-.703393, 2.63570, -.0963618, -1.79799, -2.84336)$

$P(x') = 44.021.$

Suggested Starting Points.

(a) $x = (1,1,1,1,1)$

(b) $x = (2,2,2,2,2)$

(c) $x = (-1,3,-1/2,-2,-3)$

(d) $x = (-1,2,1,-2,-2)$

(e) $x = (-2,-2,-2,-2,-2)$

Source. Miele, Moseley, Levy and Coggins [6]; also Margaret Wright [55].

Problem 12.

$$\min \quad P(x_1,x_2,x_3,x_4,x_5,x_6) = P'(x_1) + P''(x_2)$$

$$\text{s.t.} \quad x_1 - c + (x_3x_4 \cos(b-x_6) - x_3^2 A \cos(b-a))/B = 0$$

$$x_2 + (x_3x_4 \cos(b+x_6) - x_4^2 A \cos(b-a))/B = 0$$

$$(x_3x_4 \sin(b-x_6) - x_3^2 A \sin(b-a))/B - D = 0$$

$$x_5 + (x_3x_4 \sin(b+x_6) - x_4^2 A \sin(b-a))/B = 0$$

$0 \leq x_1 \leq 400$	where	$A = .90798$
$0 \leq x_2 \leq 1000$		$B = 131.078$
$340 \leq x_3 \leq 420$		$a = .00889$
$340 \leq x_4 \leq 420$		$b = 1.48477$
$-1000 \leq x_5 \leq 1000$		$c = 300$
$0 \leq x_6 \leq .5236$		$D = 200$

and

$$\frac{\partial P'}{\partial x_1}(x_1) = \begin{cases} 30 & \text{if } 0 \leq x_1 < 300 \\ 31 & \text{if } 300 \leq x_1 \leq 400 \end{cases}$$

and

$$\frac{\partial P''}{\partial x_2}(x_2) = \begin{cases} 28 & \text{if } 0 \leq x_2 < 100 \\ 29 & \text{if } 100 \leq x_2 < 200 \\ 30 & \text{if } 200 \leq x_2 < 300 \end{cases}$$

Optimal Solution and Optimal Objective Value.

$$x' = (107.834,\ 196.295,\ 373.836,\ 420.002,\ 21.293,\ .15327)$$

$$P(x') = 8827.595$$

Suggested Starting Point.

$$x = (390,\ 1000,\ 419.5,\ 340.5,\ 198.175,\ .5)$$

Source. Coleville [8].

Problem 13.

$$\min P(x_1, \ldots, x_{10}) = \sum_{i=1}^{10} x_i (c_i + \ell n(x_i / \sum_{j=1}^{10} x_j))$$

$$\text{s.t. } x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0$$

$$x_4 + 2x_5 + x_6 + x_7 - 1 = 0$$

$$x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0$$

where

$$c = (-6.089, -17.164, -34.054, -5.914, -24.721, -14.986, -24.1, -10.708, -26.662, -22.179)$$

Optimal Solution and Optimal Objective Value.

$$x' = (.0406, .1477, .7832, .0014, .4853, .0007, .0274, .0180, .0375, .0969)$$

$$P(x') = -47.761$$

Suggested Starting Point.

$$x = (.1, \ldots, .1)$$

Source. Himmelblau [26].

Problem 14.

$$\min \quad P(c_0, \ldots, c_6, k_1, \ldots, k_5) = - \sum_{i=0}^{5} (c_i - 15)^{1-v}/(1-v)$$

s.t. $k_0 = k_6 = 100$

$$k_{i+1} = 1.3k_i - c_i, \qquad i = 0, \ldots, 5$$

where

$$v = 4.5026$$

Optimal Solution and Optimal Objective Value.

$$(c',k') = (28.474, 29.282, 30.139, 31.047, 32.010, 33.031, \\ 101.526, 102.702, 103.374, 103.339, 102.330)$$

$$P(c',k') = .000120804$$

Suggested Starting Point.

$$(c,k) = (101, 101, \ldots, 101)$$

Source. Professor Alan Manne--Private Communication.

Problem 15.

$$\min \quad P(x_1,\ldots,x_{16}) = \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1)$$

$$\text{s.t.} \qquad \sum_{j=1}^{16} b_{ij} x_j = c_i \;, \qquad\qquad i = 1,\ldots,8$$

where

$$0 \leq x_j \leq 5 \qquad\qquad j = 1,\ldots,16$$

A =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1			1			1	1								1
2		1	1				1			1						
3			1				1		1	1				1		
4				1			1				1				1	
5					1	1				1		1				1
6						1		1							1	
7							1				1		1			
8								1		1					1	
9									1			1				1
10										1				1		
11											1		1			
12												1		1		
13													1	1		
14														1		
15															1	
16																1

$c = (2.5,\ 1.1,\ -3.1,\ -3.5,\ 1.3,\ 2.1,\ 2.3,\ -1.5)$ and

B =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	.22	.2	.19	.25	.15	.11	.12	.13	1							
2	-1.46		-1.3	1.82	-1.15		.8			1						
3	1.29	- .89			-1.16	- .96		-.49			1					
4	-1.1	-1.06	.95	- .54		-1.78	- .41					1				
5				-1.43	1.51	.59	- .33	-.43					1			
6		-1.72	- .33		1.62	1.24	.21	-.26						1		
7	1.12			.31			1.12		-.36						1	
8		.45	.26	-1.1	.58		-1.03		.1							1

<u>Optimal Solution and Optimal Objective Value</u>.

$$x' = (.04, .792, .203, .844, 1.27, .935, 1.682, .155, 1.568, 0, 0, 0, .66, 0, .674, 0)$$

$$P(x') = 244.9$$

<u>Source</u>. Himmelblau [26].

Problem 16.

$$\min \quad P(x_1,\ldots,x_{40}) = \frac{1}{2} \sum_{i=1}^{20} x_i^2 + \sum_{j=21}^{40} x_j$$

$$\text{s.t.} \quad x_{i+20} \exp\left(- \sum_{j=1}^{20} x_j\right) - 1 = 0, \qquad 1 \le i \le 20$$

Optimal Solution and Optimal Objective Value.

$$x_i' = \begin{cases} -.22438 & \text{if } 1 \le i \le 20 \\ .011254 & \text{if } 21 \le i \le 40 \end{cases}$$

$$P(x') = .728399$$

Suggested Starting Point.

$$x_i = \begin{cases} 0 & \text{if } 1 \le i \le 20 \\ 1 & \text{if } 21 \le i \le 40 \end{cases}$$

Source. Proposed by author.

Problem 17.

$$\min \quad P(x_1,\ldots,x_{40}) = \frac{1}{2}\sum_{i=1}^{20} x_i^2 + \sum_{j=21}^{40} x_j$$

$$\text{s.t.} \quad x_{i+20}\exp\left(-\sum_{j=1}^{20} x_j\right) - 1 = 0 \qquad 1 \le i \le 20$$

$$x_{i+20} \le .01 \qquad 1 \le i \le 20$$

Optimal Solution and Optimal Objective Value.

$$X_i = \begin{cases} -.23025 & \text{if } 1 \le i \le 20 \\ .01 & \text{if } 21 \le i \le 40 \end{cases}$$

Suggested Starting Point.

$$x_i = 0 \qquad 1 \le i \le 40$$

Source. Proposed by author.

APPENDIX A

With the theory of Chapters 1 and 4 it is now possible to show that P.4.1 is in fact a special case of decomposability of a point to set map. This is formally stated and proved in

<u>Theorem</u>. Suppose A.4.1-A.4.5 hold and that in addition P, G, h and Q are all differentiable on their respective domains. Define the function $L: R^m \to R^n$ by $L(x) = \nabla_x P(x,h(x)) + \nabla_y P(x,h(x))^T Dh(x)$. Then the point to set map $S: R^m \times R^n \to (R^m \times R^n)^*$ defined by

$$S(x,y) = \begin{cases} (x,y) - \{(L(x),G(x,y))\} & \text{if } t(x) < 0 \\ (x,y)\text{-hull } (\partial t(x) \cup \{L(x)\}) \times \{G(x,y)\} & \text{if } t(x) = 0 \\ (x,y) - \partial t(x) \times \{G(x,y)\} & \text{if } t(x) > 0 \end{cases}$$

satisfies $(S, R^m \times R^n)$ is decomposable.

<u>Proof</u>. Let $X = R^m$, $Y = R^n$, $Z = X \times Y$. Define $S_f: Z \to (X)^*$ by

$$S_f(x,y) = \begin{cases} x - \{L(x)\} & \text{if } t(x) < 0 \\ x\text{-hull}(\partial t(x) \cup \{L(x)\}) & \text{if } t(x) = 0 \\ x - \partial t(x) & \text{if } t(x) > 0 \end{cases}$$

and $S_g: Z \to (Y)^*$ by

$$S_g(x,y) = y - \{G(x,y)\} .$$

Property (1) of D.2.2 holds by construction. Furthermore, property (2) of D.2.2 holds trivially since for each $x \in R^m$,

$$S_g(x,h(x)) = h(x) - \{G(x,h(x))\} = h(x) - \{0\} = \{h(x)\}. \quad \|$$

It is of course important to note that under the conditions developed in Chapter 4, a solution to P.4.1 may be obtained by finding a fixed point of the point to set map $S_r : X \to (X)^*$ defined by $S_r(x) = S_f(x,h(x))$.

BIBLIOGRAPHY

[1] Abadie, J. and J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in Optimization, R. Fletcher, ed., Academic Press (1969), pp. 37-47.

[2] Avriel, M., Nonlinear Programming Analysis and Methods, Prentice-Hall Inc., Englewood, N.J. (1976).

[3] Berge, C., Topological Spaces (translated by E. M. Patterson), New York, The MacMillan Company (1963).

[4] Broyden, C.G., J.E. Dennis, Jr., and J.J. More, "On the Local and Superlinear Convergence of Quasi-Newton Methods," J. Inst. Maths. Applics. 12 (1973), pp. 223-245.

[5] Cairns, S. S., Introductory Topology, New York, The Ronald Press Company (1961).

[6] Coggins, G.M., A.V. Levy, A. Miele and P.E. Moseley, "On the Method of Multipliers for Mathematical Programming Problems," Journal of Optimization Theory and Applications, Vol. 10, No. 1, July 1972, pp. 1-33.

[7] Cohen, D.I.A., "On the Sperner Lemma," J. Comb. Theory 2 (1967), pp. 585-587.

[8] Colville, A.R., "A Comparative Study of Nonlinear Programming Codes," TR-320-2949, IBM New York Scientific Center, New York (1968).

[9] Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton, N.J. (1963), pp. 1-621.

[10] Davidon, W., "Variable Metric Methods for Minimization," A.E.C. Res. and Develop. Rept. ANL-5990, Argonne National Lab., Argonne, Illinois.

[11] Eaves, B.C., "Nonlinear Programming Via Kakutani Fixed Points," Working Paper No. 294, Center for Research in Management Science, University of California, Berkeley (1970).

[12] Eaves, B.C., "Computing Kakutani Fixed Points," SIAM J. of Appl. Math. 21 2 (1971), pp. 236-244.

[13] Eaves, B.C., "Homotopies for Computation of Fixed Points," Mathematical Programming 3 1 (1972), pp. 1-22.

[14] Eaves, B.C., "A Short Course in Solving Equations with PL Homotopies," SIAM-AMS Proc. 9 (1976), pp. 73-143.

[15] Eaves, B.C. and R. Saigal, "Homotopies for Computation of Fixed Points on Unbounded Regions," Mathematical Programming 3 2 (1972) pp. 225-237.

[16] Engels, C., "Economic Equilibrium Under Deformation of the Economy," Ph.D. Thesis, Department of Operations Research, Stanford University, Stanford, Ca. (1976), to appear.

[17] Fisher, M.L. and F.J. Gould, "An Algorithm for the Nonlinear Complementarity Problem," University of Chicago: Center for Math Studies in Business and Economics, Report 7311 (1973).

[18] Fletcher, R. and M. Powell, "A Rapidly Convergent Descent Method for Minimization," Comput. J. 6, pp. 163-168.

[19] Freidenfelds, J., "A Fixed Point Algorithm and Almost-Complementary Sets," Operations Research Department, Stanford University, Stanford, Ca. (April 1971).

[20] Gale, D., The Theory of Linear Economic Models, New York, McGraw-Hill Book Co. (1960).

[21] Garcia, C.B., "Computation of Solutions to Nonlinear Equations Under Homotopy Invariance," Graduate School of Business, University of Chicago (1975), 8 pp.

[22] Garcia, C.B., "A Global Existence Theorem for the Equation $Fx = y$," Center for Mathematical Studies in Business and Economics, Report 7527, University of Chicago (1975), 17pp.

[23] Gochet, W., E. Loute and D. Solow, "Comparative Computer Results of Three Algorithms for Solving Prototype Geometric Programming Problems," CORE Reprint 212, Belgium (1974).

[24] Gould, F.J. and J.W. Tolle, "A Unified Approach to Complementarity in Optimization," University of Chicago, Department of Economics, Report 7312 (1973).

[25] Hansen, T. and H. Scarf, "On the Applications of a Recent Combinatorial Algorithm," Cowles Foundation Discussion Paper No. 272, Yale University (1969).

[26] Himmelblau, D.M., Applied Nonlinear Programming, McGraw-Hill (1972).

[27] Hirsch, M.W., "A Proof of the Nonretractibility of a Cell onto its Boundary," Proc. of AMS 14 (1963), pp. 364-365.

[28] Kakutani, S., "A Generalization of Brouwer's Fixed Point Theorem," Duke Math. J. 8 (1941), pp. 457-459.

[29] Kellogg, R.B., T.Y. Li and J. Yorke, "A Constructive Proof of the Brouwer Fixed Point Theorem and Computational Results," SIAM J. Numer. Anal. 13 4 (1976), pp. 473-483.

[30] Knaster, B.C., Kuratowski and S. Mazurkiewicz, "Ein Beweis des Fixpuntsatzes fur n-dimensionale Simplexe," Fund. Math. 14 (1929), pp. 132-137.

[31] Kojima, M., "On the Homotopic Approach to Systems of Equations with Separable Mappings," TR-B26, Department of Information Science, Tokyo Institute of Technology (September 1975).

[32] Kuhn, H.W., "Simplicial Approximations of Fixed Points," Proc. Nat. Acad. Sci., U.S.A. 61 (1968), pp. 1238-1242.

[33] Kuhn, H.W., "A New Proof of the Fundamental Theorem of Algebra," Mathematical Programming Study 1 (1974), pp. 148-158.

[34] Lasdon, L.S., A. Waren, A. Jain and M.W. Ratner, "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," TR SOL 76-3, Department of Operations Research, Stanford University, Stanford, Ca. (February 1976).

[35] Lemke, C.E., "Bimatrix Equilibrium Points and Mathematical Programming," Mangement Science 11 (1965), pp. 681-689.

[36] Lemke, C.E. and J.T. Howson, Jr., "Equilibrium Points of Bimatrix Games," SIAM J. on Appl. Math. 12 2 (1964), pp. 413-423.

[37] Mangasarian, O.L., Nonlinear Programming, New York, McGraw-Hill Book Company (1969).

[38] Merrill, O.H., "Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Upper Semi-Continuous Point to Set Mappings," Ph.D. Thesis, Dept. of Ind. Engineering, University of Michigan (1972), 228 pp.

[39] Murray, W., Numerical Methods for Unconstrained Optimization, Academic Press, London (1972).

[40] Nishiyama, H., M .G. Simkin and H. Takeuchi, "A Controlled Test of Convergence Efficiencies of GRG, FLT and SUMT," Presented at the 1976 Joint National Meeting of ORSA/TIMS (1976).

[41] Ortega, J.M. and W.C. Rhineboldt, Iterative Solutions of Nonlinear Equations in Several Variables, Academic Press, New York-London (1970).

[42] Powell, M., "A Method for Nonlinear Constraints in Minimization Problems," in Optimization (edited by R. Fletcher), Academic Press, New York-London (1969), pp. 283-298.

[43] Rockafeller, R.T., Convex Analysis, Princeton, Princeton University Press (1970).

[44] Saigal, R., "Investigations into the Efficiency of Fixed Point Algorithms," Fixed Points: Algorithms and Applications, Ed: S. Karamardian (1974).

[45] Saigal, R., "On the Convergence Rate of Algorithms that are Based on Methods of Complementary Pivoting," Bell Laboratories, Holmdel, New Jersey (undated), 60 pp.

[46] Saigal, R., "On Paths Generated by Fixed Point Algorithms," Mathematics of Operations Research 1 4 (1976), pp. 359-380.

[47] Saigal, R., D. Solow and L.A. Woolsey, "A Comparative Study of Two Algorithms to Compute Fixed Points over Unbounded Regions," presented at the 8th Mathematical Programming Symposium, Stanford (1973).

[48] Scarf, H., "The Approximation of Fixed Points of a Continuous Mapping," SIAM J. of Appl. Math. 15 5 (1967), pp. 1328-1343.

[49] Scarf, H., "The Core of an N Person Game," Econometrica 35 1 (1967), pp. 50-69.

[50] Stoer, J. and C. Witzgall, Convexity and Optimization in Finite Dimensions I, New York, Springer-Verlag (1970).

[51] Todd, M.J., "On Triangulations for Computing Fixed Points," TR 234, Department of Operations Research, Cornell University (1974), 23 pp.

[52] Todd, M.J., "Union Jack Triangulations," TR 220, Dept. of Operations Research, Cornell University, Fixed Points: Algorithms and Applications, Ed: S. Karamardian (1975).

[53] Wilmuth, R.J., "The Computation of Fixed Points," Ph.D. Thesis, Department of Operations Research, Stanford University (1973).

[54] Wilson, R., "The Bilinear Complementarity Problem and Competitive Equilibria of Linear Economic Models, TR SOL 76-2, Department of Operations Research, Stanford University (1976), 28pp.

[55] Wright, M., "Numerical Methods for Nonlinearly Constrained Optimization," SLAC Report No. 193, Stanford University (March 1976).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (14) SOL-77-32	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): (6) DECOMPOSITION IN FIXED POINT COMPUTATION	5. TYPE OF REPORT & PERIOD COVERED: (9) Technical Report
	6. PERFORMING ORG. REPORT NUMBER: SOL 77-32
7. AUTHOR(s): (10) Daniel Solow	8. CONTRACT OR GRANT NUMBER(s): (15) DAAG-29-74-C-0032, N00014-75-C-0865
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Operations Research--SOL, Stanford University, Stanford, CA 94305	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-143
11. CONTROLLING OFFICE NAME AND ADDRESS: Mathematics Division, U.S. Army Research Office, Box CM, Duke Station, Durham, N.C. 27706; Operations Research Program--ONR, Department of the Navy, 800 N. Quincy Street, Arlington, VA 22217	12. REPORT DATE: (11) Dec [redacted] 77
	13. NUMBER OF PAGES: (12) 128 P.
	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fixed Points
Equality Constraints
Complementarity
Point to Set Maps
Nonlinear Programming
Inequality Constraints
Simplicial Subdivisions

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE ATTACHED

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

hc

408765

SOL 77-32 "Decomposition in Fixed Point Computation"

In the past decade, several constructive proofs of the Brouwer and Kakutani fixed point theorems have emerged. These proofs have been developed into algorithms (known in the literature as complementary pivot algorithms) which search for fixed points on unbounded regions. In turn these algorithms have been used to solve problems arising in economics, engineering and other branches of applied mathematics. An important application for which this method was cumbersome and inefficient to use was that of optimizing an objective function subject to both equality and inequality constraints (hereafter referred to as the general constrained optimization problem). One result of ~~the dissertation~~ is the most efficient complementary pivot algorithm to date for handling ~~this~~ problem. The second major contribution ~~of this thesis~~ is a general structure on fixed point problems which, when present, enables one to work in a lower dimensional space. It is shown that the general constrained optimization problem may sometimes be formulated as a fixed point problem possessing this property.

The basic approach adopted in this work for handling the general constrained optimization problem is to use an implicit function (derived from the equality constraints) to solve for some dependent variables in terms of the remaining independent ones. Under certain circumstances, a fixed point algorithm may be used to search for optimal values of the independent variables while Newton's method is used to determine values of the dependent variables. Theoretical conditions on the original functions are developed to guarantee that the fixed point algorithm converges to a solution and various techniques are devised to enhance the overall efficiency.

To help ascertain the value of this method, comparative computer tests are run against the Generalized Reduced Gradient (GRG) algorithm which is a well established nonlinear programming code. This method was selected as the basis for comparison because, to the author's knowledge, it is the best commercial code for solving the general constrained optimization problem. Seventeen test problems were taken from various sources. The fixed point code solved all seventeen and GRG solved sixteen. This supports the robustness of the fixed point approach. As to the computer times, the fixed point code proved to be as fast or faster than GRG on the lower dimensional problems. As the dimension increased, however, the trend reversed and on a forty dimensional problem GRG was approximately eleven times faster. The conclusion is that when the dimension of the original problem can be sufficiently reduced by the equality constraints, the fixed point approach appears to be more effective.