ALGORITHMS FOR COMPLEMENTARITY PROBLEMS AND GENERALIZED EQUATIONS

By

Stephen C. Billups

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy
(Computer Sciences)

at the
UNIVERSITY OF WISCONSIN – MADISON
1995

Abstract

Algorithms for Complementarity Problems and Generalized Equations
Stephen C. Billups
Under the Supervision of Associate Professor Michael C. Ferris
at the University of Wisconsin–Madison

Recent improvements in the capabilities of complementarity solvers have led to an increased interest in using the complementarity problem framework to address practical problems arising in mathematical programming, economics, engineering, and the sciences. As a result, increasingly more difficult problems are being proposed that exceed the capabilities of even the best algorithms currently available. There is, therefore, an immediate need to improve the capabilities of complementarity solvers. This thesis addresses this need in two significant ways. First, the thesis proposes and develops a proximal perturbation strategy that enhances the robustness of Newton-based complementarity solvers. This strategy enables algorithms to reliably find solutions even for problems whose natural merit functions have strict local minima that are not solutions. Based upon this strategy, three new algorithms are proposed for solving nonlinear mixed complementarity problems that represent a significant improvement in robustness over previous algorithms. These algorithms have local Q-quadratic convergence behavior, yet depend only on a pseudo-monotonicity assumption to achieve global convergence from arbitrary starting points. Using the MCPLIB and GAMSLIB test libraries, we perform extensive computational tests that demonstrate the effectiveness of these algorithms on realistic problems.

Second, the thesis extends some previously existing algorithms to solve more general problem classes. Specifically, the NE/SQP method of Pang & Gabriel (1993), the semismooth equations approach of De Luca, Facchinei & Kanzow (1995), and the infeasible-interior point method of Wright (1994) are all generalized to the mixed complementarity problem framework. In addition, the pivotal method of Cao & Ferris (1995*b*), which solves affine variational inequalities, is extended to solve affine generalized equations. To develop this extension, the piecewise-linear homotopy framework of Eaves (1976) is used to generate an algorithm for finding zeros of piecewise affine maps. We show that the resulting algorithm finds a solution in a finite number of iterations under the assumption that the piecewise affine map is coherently oriented.

Acknowledgements

One of the greatest rewards of pursuing a Ph.D. has been the chance to work closely with my advisor, Michael Ferris. Over the past four years, he has been a constant source of enthusiasm, advice, guidance, and ideas. I am grateful for all he has done for me, and even more grateful for his friendship.

I would like to thank Robert Meyer and John Strikwerda for serving on my Ph.D. committee and Olvi Mangasarian and Stephen Robinson for serving as readers. I am especially grateful to Professor Robinson, who provided numerous comments that greatly improved the thesis and to Professor Mangasarian for encouraging me to come to Wisconsin. I am also grateful to Professor Strikwerda for serving as my advisor during my first year here.

In the course of my education, I have benefited from the hard work and dedication of over 100 teachers. I am grateful to all of them, but there are two that I particularly want to thank. First is my high school math team coach, André Samson. I am but one of many students he turned on to mathematics, and for that I will always be grateful. I would also like to thank Layne Watson, who was my advisor for my Master's degree at Virginia Tech.

In addition to teachers, there have been several people that have been tremendously helpful during my stay in Wisconsin. I thank Lorene Webber for taking care of countless administrative details, always with a smile on her face. I am also extremely grateful to the computer lab for their outstanding service.

Finally, I would like to acknowledge the support I have received from my family. I thank my parents for bringing me up in a happy and secure home, for encouraging me to achieve, and for making me tremendously proud to be their son. I thank my brothers for providing some of my early motivation to study mathematics and for continuing to be role models. But most importantly, I thank my wife, Sarah. She is the most important part of my life, and I thank her for making me happy.

This research was partially funded by a Fellowship from the Wisconsin Alumni Research Foundation, as well as by the National Science Foundation under Grant CCR–9157632, the Department of Energy under Grant DE-FG03-94ER61915, and the Air Force Office of Scientific Research under Grants AFOSR–89–0410 and F49620-94-1-0036.

Contents

List of Figures

Glossary

AGE	affine generalized equation.
AVI	affine variational inequality.
GE	generalized equation.
HLCP	horizontal linear complementarity problem.
LCP	linear complementarity problem.
LMCP	linear mixed complementarity problem.
MCP	mixed complementarity problem.
NCP	nonlinear complementarity problem.
NE/SQP	algorithm for MCP based on Gauss-Newton method (Pang & Gabriel 1993).
NE/NEWT	algorithm for MCP based on Newton method.
PATH	algorithm for MCP based on path search strategy (Dirkse & Ferris 1995*b*).
PROXI	NE/NEWT with proximal perturbation strategy.
QPCOMP	NE/SQP with proximal perturbation strategy.
SEMICOMP	SEMISMOOTH with proximal perturbation strategy.
SEMISMOOTH	algorithm for MCP based on semismooth equations (De Luca et al. 1995).
SMOOTH	algorithm for MCP based on smoothing technique (Chen & Mangasarian 1995*a*).
VI	variational inequality.

Chapter 1

Introduction

Over the last thirty years, the class of problems known as complementarity problems has become increasingly popular as a tool for addressing practical problems arising in mathematical programming, economics, engineering, and the sciences (Ferris & Pang 1995, Dirkse & Ferris 1995*a*). It is not surprising then that the mathematical programming community has devoted considerable energies toward developing robust and efficient algorithms for solving these problems. Conversely, it is the considerable success of some of these algorithms (Dirkse & Ferris 1995*b*, Rutherford 1993, Chen & Mangasarian 1995*a*, Pang & Gabriel 1993, Harker & Xiao 1990) that has generated much of the recent enthusiasm for applying the complementarity framework to new applications.

Although the effectiveness of complementarity algorithms has improved substantially in recent years, the fact remains that increasingly more difficult problems are being proposed that are exceeding the capabilities of these algorithms. As a result, there is an immediate need to improve the capabilities of complementarity solvers. This thesis addresses this need in two significant ways. First, the thesis proposes and develops a proximal perturbation strategy that enhances the robustness of Newton-based complementarity solvers. This strategy enables algorithms to reliably find solutions even for problems whose natural merit functions have strict local minima that are not solutions. Based upon this strategy, three new algorithms are proposed for solving nonlinear mixed complementarity problems that represent a significant improvement in robustness over previous algorithms. These algorithms have local Q-quadratic convergence behavior, yet depend only on a pseudo-monotonicity assumption to achieve global convergence from arbitrary starting points.

Second, the thesis extends some previously existing algorithms to solve more general problem classes. Specifically, the NE/SQP method of Pang & Gabriel (1993), the semismooth equations approach of De Luca et al. (1995), and the infeasible-interior point method of Wright (1994) are all generalized to the mixed complementarity problem framework. In addition, the pivotal method of Cao & Ferris (1995*b*), which solves affine variational inequalities, is extended to solve affine generalized equations. To develop this extension, the piecewise-linear homotopy framework of Eaves (1976) is used to generate an algorithm for finding zeros of piecewise affine maps. We show that the resulting algorithm finds a solution in a finite number of iterations under the assumption that the piecewise affine map is coherently oriented.

1.1 Notation and Definitions

When discussing matrices, vectors and vector-valued functions, subscripts are used to indicate components, whereas superscripts are used to indicate the iteration number or some other label. For example $A_{i\cdot}, A_{\cdot j}, A_{ij}$ refer to the ith row, jth column, and (i,j)th entry of A, respectively, whereas x^k typically represents the kth iterate generated by an algorithm. In contrast to the above, for scalars or scalar-valued functions, we use subscripts to refer to labels so that superscripts can be reserved for exponentiation. Index sets can also be used to refer to subsets of the components. For example if J and K are subsets of $\{1,\ldots,n\}$, then $M_{J,K}$ denotes the $|J|\times|K|$ submatrix of M consisting of the elements $M_{jk}, j \in J, k \in K$. In concatenating matrices and vectors, we use the MATLAB notation of a comma to separate columns and a semicolon to separate rows. For example, if we have the expressions $A = (x, y)$ and $w = (x; y)$, then A is a matrix with columns x and y, whereas w is the vector formed by concatenating x and y. The vector of all ones is represented by e and the jth unit vector is represented by e^j.

Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm. Inner products are denoted by $\langle\cdot,\cdot\rangle$. We use the notation $(\cdot)_+$, $(\cdot)_-$, and $|\cdot|$ to represent the plus, minus, and absolute value operators, respectively, for vectors. That is, $x_+ := (\max(x_1, 0); \ldots; \max(x_n, 0))$, $x_- := (\max(-x_1, 0); \ldots; \max(-x_n, 0))$, and $|x| := (|x_1|; \ldots; |x_n|)$.

The symbols R_+ and R_{++} refer to the nonnegative real numbers and the positive real numbers, respectively. The extended real numbers are denoted by $\overline{\mathrm{R}} := \mathrm{R} \bigcup \{-\infty, +\infty\}$. The vectors l and $u \in \overline{\mathrm{R}}^n$, specify a set of lower and upper bounds. Throughout this thesis we assume that $l < u$. The symbol B represents the box defined by $\mathrm{B} := [l, u] := \{x \mid l \leq x \leq u\}$.

For a function $f : C \subset \mathrm{R}^n \to \mathrm{R}^m$, we define $\nabla_i f_j(x) := \partial f_j(x)/\partial x_i$. $\nabla f(x)$ is the $n \times m$ matrix whose ijth element is $\nabla_i f_j(x)$. Thus, if f is a scalar valued function, then $\nabla f(x)$ is a *column* vector. Finally, we define $f'(x) = \nabla f(x)^\top$.

The directional derivative of f evaluated at the point x in the direction d is denoted by

$$f'(x;d) := \lim_{\lambda \downarrow 0} \frac{f(x+\lambda d) - f(x)}{\lambda},$$

provided the limit exists. Note that if x is a stationary point of f on C, then $f'(x;d) = 0$ for all d such that $x + d \in C$.

The notation $\rightrightarrows$ is used to indicate a point to set mapping or *multifunction*. Thus, $T : \mathrm{R}^n \rightrightarrows \mathrm{R}^m$ indicates that T is a multifunction, which maps points in R^n to subsets of R^m. In discussing multifunctions, we may refer either to a point to set mapping, $T(\cdot)$, or to the graph of that mapping, which is the set $T := \{(x,y) \in \mathrm{R}^n \times \mathrm{R}^m \mid y \in T(x)\}$. The expression $T^{-1}(\cdot)$ is defined as a set inverse; i.e., $T^{-1}(y) := \{x | (x,y) \in T\}$. Further, $T^{-1} := \{(y,x) | (x,y) \in T\}$. The effective domain of T, is defined by $\mathrm{dom}(T) := \{x | T(x) \neq \emptyset\}$.

For a set C, $\mathrm{aff}(C)$, $\mathrm{co}(C)$, $\mathrm{int}(C)$, $\mathrm{ri}(C)$, $\mathrm{rec}(C)$, $\dim(C)$, and $\delta(\cdot|C)$ refer to the affine hull, convex hull, interior, relative interior, recession cone, dimension, and indicator function of C, respectively (see Rockafellar (1970) for definitions of these terms). The projection operator for the set C is denoted by $\pi_C(\cdot)$. That is $\pi_C(x)$ represents the projection (with respect to the Euclidean norm) of x onto the set C.

In several convergence proofs, we use the notation $\mathcal{O}(\cdot)$ and $o(\cdot)$ as follows: given a sequence $\{u^k\}$, we use the expression $\{\mathcal{O}(u^k)\}$ to represent any sequence $\{v^k\}$ satisfying

$$\limsup_{k \to \infty} \frac{\left\| v^k \right\|}{\|u^k\|} < \infty.$$

Given a function $h : \mathrm{R}^n \to \mathrm{R}^m$, we use the expression $o(h(x))$ to represent any function $g : \mathrm{R}^n \to \mathrm{R}^s$ satisfying

$$\lim_{\|x\| \to 0} \frac{\|g(x)\|}{\|h(x)\|} = 0.$$

In proving convergence results, we will refer to the following convergence rates. A sequence $\{x^k\}$ is said to converge to x^* *Q-linearly* if

$$0 < \limsup_{k \to \infty} \frac{\left\| x^{k+1} - x^* \right\|}{\|x^k - x^*\|} < 1.$$

The sequence converges *Q-superlinearly* if

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

The sequence is said to converge *Q-quadratically* if

$$\limsup_{k\to\infty} \frac{\left\|x^{k+1} - x^*\right\|}{\left\|x^k - x^*\right\|^2} < \infty.$$

The sequence converges *Q-subquadratically* if

$$\limsup_{k\to\infty} \frac{\left\|x^{k+1} - x^*\right\|}{\left\|x^k - x^*\right\|^p} < \infty \quad \forall\, p \in [1, 2).$$

Finally, several types of monotonicity need to be defined:

Definition 1.1.1 *A function $f : X \subset R^n \to R^n$ is said to be*

1. monotone *if*
$$\langle f(x) - f(y), x - y\rangle \geq 0 \quad \forall x, y \in X,$$

2. strongly monotone *with modulus $\alpha > 0$ if*
$$\langle f(x) - f(y), x - y\rangle \geq \alpha \left\|x - y\right\|^2 \quad \forall x, y \in X, x \neq y$$

1.2 Problem Classes

Most of the literature about complementarity problems is concerned with the standard forms given by the nonlinear complementarity problem (NCP) and the linear complementarity problem (LCP): Given a function $f : \Omega \to \mathrm{R}^n$, where $\mathrm{R}^n_+ \subset \Omega \subset \mathrm{R}^n$, the nonlinear complementarity problem is defined by

$$\begin{array}{ll} \mathrm{NCP}(f): & \text{Find } x \in \mathrm{R}^n_+ \text{ such that} \\ & f(x) \geq 0 \text{ and } \langle x, f(x)\rangle = 0. \end{array}$$

When f is an affine function, i.e. $f(x) := Mx + q$, with $M \in \mathrm{R}^{n\times n}$ and $q \in \mathrm{R}^n$, $\mathrm{NCP}(f)$ reduces to the linear complementarity problem:

$$\begin{array}{ll} \mathrm{LCP}(M, q): & \text{Find } x \in \mathrm{R}^n_+ \text{ such that} \\ & Mx + q \geq 0, \text{ and } \langle x, Mx + q\rangle = 0. \end{array}$$

A generalization of the LCP is the horizontal linear complementarity problem given by

$$\begin{array}{ll} \mathrm{HLCP}(M, N, q): & \text{Find } (x, y) \in \mathrm{R}^n_+ \times \mathrm{R}^n_+ \text{ such that} \\ & Mx + Ny = q, \text{ and } \langle x, y\rangle = 0, \end{array}$$

where $M, N \in \mathbb{R}^{n \times n}$.

While the standard forms are convenient from a theoretical viewpoint, many practical problems are more naturally formulated using the framework of the mixed complementarity problem (MCP). The MCP is defined in terms of a box $\mathrm{B} := [l, u] \subset \overline{\mathbb{R}}^n$, with $l < u$, and a function $f : \Omega \to \mathbb{R}^n$, where $\mathrm{B} \subset \Omega \subset \mathbb{R}^n$:

$$\text{MCP}(f, \mathrm{B}): \qquad \text{Find } x \in \mathrm{B} \text{ such that} \\ \langle x - l, f(x)_+ \rangle = 0 \text{ and } \langle u - x, f(x)_- \rangle = 0.$$

Note that the assumption that $l < u$ represents no loss of generality, since if $l_i = u_i$ then the corresponding variable x_i must be fixed and can therefore be removed from the problem. In the special case where f is the affine function $f(x) := Mx + q$, we get the linear mixed complementarity problem given by

$$\text{LMCP}(M, q, \mathbf{B}): \qquad \text{Find } x \in \mathrm{B} \text{ such that} \\ \langle x - l, (Mx + q)_+ \rangle = 0 \text{ and } \langle u - x, (Mx + q)_- \rangle = 0.$$

In the above definitions, note that components of the bounds l and u may be infinite. We therefore adopt the convention that $\pm\infty \times 0 = 0$. Using this convention, it is easily seen that the NCP is the special case of the MCP given by letting $l = 0$, and $u = \infty$. Stated another way, NCP(f) is equivalent to MCP($f, \mathbb{R}^n_+$).

Conversely, the MCP can be reformulated as an NCP (Ferris & Pang 1995). Thus, one way of solving a mixed complementarity problem is simply to reformulate it and solve it as an NCP. However, this is often a tedious and error prone task. Moreover, even if the reformulation is performed automatically, efficiency may be lost because the resulting NCP usually involves a larger number of variables, and although a great deal of structure will exist in the reformulated problem, the NCP solver will be unable to exploit this structure. We therefore contend that algorithms for solving complementarity problems should be aimed at solving problems in the MCP format rather than the standard NCP or LCP format.

Note that by setting $\Omega = \mathrm{B}$, the definition of MCP requires only that f is defined on B. However, for theoretical purposes, we shall assume throughout this thesis that Ω is an open set and that f is continuous on Ω.

The NCP and MCP are special cases of the variational inequality problem, which is defined in terms of a set $C \subset \mathbb{R}^n$ and a function $f : \Omega \to \mathbb{R}^n$, where $C \subset \Omega \subset \mathbb{R}^n$:

$$\text{VI}(f, C): \qquad \text{Find } x \in C \text{ such that} \\ \langle f(x), y - x \rangle \geq 0 \quad \forall y \in C$$

It is easily shown that NCP(f) is equivalent to VI($f, \mathbb{R}^n_+$) and MCP(f, B) is equivalent to VI(f, B) (Dirkse 1994, Theorem 2).

Another way of writing the variational inequality makes use of the concept of a *normal cone.*

Definition 1.2.1 *Given a closed convex set* $C \subset \boldsymbol{R}^n$*, the* normal cone $N_C(x)$ *to* C *at a point* $x \in \boldsymbol{R}^n$ *is defined by*

$$N_C(x) := \begin{cases} \{w \mid \langle w, y - x\rangle \leq 0, \quad \forall y \in C\} & x \in C \\ \emptyset & x \notin C \end{cases}$$

By comparing the definitions, it is easily seen that x solves VI(f, C) if and only if $-f(x) \in N_C(x)$. Thus, the variational inequality can be restated as the problem of finding $x \in C$, such that

$$0 \in f(x) + N_C(x). \tag{1}$$

This equation is a special case of the generalized equation (Robinson 1979*b*), which is defined in terms of maximal monotone multifunctions (Rockafellar 1970).

Definition 1.2.2 *A multifunction* $T : \boldsymbol{R}^n \rightrightarrows \boldsymbol{R}^n$ *is said to be* monotone *if for each* (x^1, y^1)*,* (x^2, y^2) *in the graph of* T*,*

$$\left\langle x^1 - x^2, y^1 - y^2 \right\rangle \geq 0.$$

T *is* maximal *if its graph is not properly contained in that of any other monotone multifunction.* T *is* polyhedral *if its graph is the union of finitely many polyhedral convex sets.*

Given a maximal monotone multifunction T, a set $\Omega \supset \text{dom}(T)$ and a function $f : \Omega \to \mathrm{R}^n$, the generalized equation GE(f, T) is to

$$\text{GE}(f,T): \qquad \begin{array}{l} \text{find } x \in \text{dom}(T) \text{ such that} \\ 0 \in f(x) + T(x). \end{array}$$

If f is an affine function, and if T is polyhedral, then we get the *affine generalized equation.*

It is well-known (Rockafellar 1970) that the normal cone is a maximal monotone multifunction. Thus, in light of (1), the variational inequality VI(f, C) is equivalent to the generalized equation GE(f, N_C).

1.3 Solution Methods for Nonlinear Problems

This section describes several techniques for solving nonlinear complementarity problems that will serve as background for the algorithms presented in Chapters 2 and 3 of this thesis. Later, in Section 1.4, we will discuss solution methods for linear complementarity problems.

There is currently a wide variety of techniques for solving nonlinear complementarity problems. Some of the older approaches include classical pivotal methods (Cottle 1966, Habetler & Kostreva 1978) and fixed-point homotopy methods (Garcia & Zangwill 1981, Todd 1976). More recent approaches include differentiable optimization based methods (Fukushima 1992, Kanzow 1994, Mangasarian 1976, Mangasarian & Solodov 1993, Tseng, Yamashita & Fukushima 1994), projection and proximal methods (Auslander 1993, Auslander & Haddou 1994, Ferris & Ralph 1995, Moré 1994, Solodov & Tseng 1994), and interior point methods, (Chen & Harker 1995, Güler 1993, Kojima, Megiddo & Noma 1991, Kojima, Mizuno & Noma 1989, Monteiro, Pang & Wang 1995, Wang, Monteiro & Pang 1994, Wright & Ralph 1993). In addition to these, Ferris & Pang (1995) list several other classes of algorithms, which we shall discuss in detail in the context of Newton-based methods.

1.3.1 Newton-Based Methods

While many different approaches are available, the fastest algorithms for solving nonlinear complementarity problems are all based upon Newton's method. Newton's method is an iterative technique for solving the system $f(x) = 0$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable. The iterates $\{x^k\}$ are generated by solving a sequence of linear approximations to f given by a first order Taylor series approximation. Specifically, at the kth iteration, the *Newton direction* d^k is computed by solving the following linear subproblem for the vector d:

$$f(x^k) + \nabla f(x^k)^\top d = 0. \tag{2}$$

The next iterate x^{k+1} is then set equal to $x^k + d^k$. This method is known to be Q-quadratically convergent in a neighborhood of a solution x^* if $\nabla f(x^*)$ is nonsingular and if ∇f is Lipschitzian in a neighborhood of x^* (Ortega & Rheinboldt 1970).

This fast local convergence rate is an attractive feature of Newton's method; however, to ensure convergence starting far away from a solution, a globalizing strategy needs to be performed. There are two main techniques for doing this, which are based upon linesearch strategies and trust-region strategies. Our interest here is in linesearch techniques, which give us the *damped Newton methods*;

for information on the trust-region methods, see Fletcher (1987, Chapter 5).

The linesearch strategy is based upon forcing decrease of a *merit function* $\theta : \mathbb{R}^n \rightarrow \mathbb{R}_+$, which in some way measures how close a point is to a solution. Normally, the merit function is nonnegative, and is equal to zero only at solutions to the problem. For example, we might choose $\theta := \|f(\cdot)\|^2/2$. The linesearch strategy is then to use this merit function to choose a step length $\alpha_k \in (0,1]$ so that the new iterate $x^{k+1} := x^k + \alpha_k d^k$ produces an acceptable decrease in the merit function.

A popular linesearch technique is the backtracking scheme based on the work of Armijo (1966) and Goldstein (1967). Given a parameter $\rho \in (0,1)$, the strategy is to evaluate the merit function at the sequence of points $\{x^k + \rho^j d^k : j = 0, 1, \ldots\}$, until a steplength ρ^m is found such that $x^k + \rho^m d^k$ produces "sufficient" decrease in the merit function. Under appropriate criteria for what constitutes "sufficient" decrease, global convergence results can be established, which guarantee that the iterates will either be unbounded, or will converge to a local minimum of the merit function θ. Note that these global convergence results do not guarantee that a solution will be found from arbitrary starting points. For example, a "globally convergent" algorithm might converge to a local minimum that is not a global minimum.

Newton's method has the limitation that the direction finding subproblem (2) may not always be solvable. In particular, if $\nabla f(x^k)$ is singular, then the Newton direction cannot be calculated unless $f(x^k) \in \ker(\nabla f(x^k)^\top)$. This difficulty is avoided by the *Gauss-Newton* method. In this method, the *Gauss-Newton direction* is chosen by solving the least squares problem given by

$$(3) \qquad \begin{array}{cc} \min & \frac{1}{2}\left\| f(x^k) + \nabla f(x^k)^\top d \right\|^2 \\ \text{subject to} & x^k + d \in \operatorname{dom}(f). \end{array}$$

This problem always has a solution, although not necessarily a unique one. Moreover, if ∇f is non-singular, then the Newton direction will be the *unique* solution to the least squares problem.

1.3.2 Reformulations of Complementarity Problems

Since the publication of Mangasarian (1976), numerous algorithms have been proposed based upon reformulating nonlinear complementarity problems as systems of nonlinear equations $H(x) = 0$. Many of these proposals have involved smooth reformulations, that is, where the function H is continuously differentiable. A simple example comes from Mangasarian (1976)

$$H_i(x) = (f_i(x) - x_i)^2 - |f(x)_i| f_i(x) - |x_i| x_i.$$

Other examples are given in Watson (1979), Ferris & Lucidi (1991), Fukushima (1992), Mangasarian & Solodov (1993), and Subramanian (1993). The chief advantage of smooth reformulations is that the Newton-type methods can be employed without modification.

However, smooth reformulations suffer from a fundamental flaw: fast local convergence is achieved only if the problem is nondegenerate, that is, if at a solution x^*, $f_i(x^*) = 0 \Rightarrow x_i^*$ is not at a bound. The reason for this is that, at a degenerate solution x^*, $\nabla H(x^*)$ is singular, so the fast local convergence rates of Newton's method are lost.

In contrast, a considerable number of algorithms have been proposed which instead use a nonsmooth reformulation. See for example, De Luca et al. (1995), Dirkse & Ferris (1995*b*), Facchinei & Soares (1994), Facchinei & Soares (1995), Ferris & Ralph (1995), Fischer & Kanzow (1994), Geiger & Kanzow (1994), Han, Pang & Rangaraj (1992), Harker & Xiao (1990), Moré (1994), Pang (1990), Pang (1991), Pang & Gabriel (1993), Ralph (1994), Robinson (1992), Xiao & Harker (1994*a*), and Xiao & Harker (1994*b*). These nonsmooth reformulations do not force the Jacobian of H to be singular at degenerate solutions. Thus, fast local convergence can be achieved even to degenerate solutions. However, because the function H is nonsmooth, the classical Newton type methods cannot be used directly; instead, modifications must be made to handle the nonsmoothness.

There are three nonsmooth reformulations for NCP(f) that we will be interested in. The first, called the *minimum map* is defined as follows:

$$H_i(x) := \min(x_i, f_i(x)). \tag{4}$$

This reformulation is the basis of algorithms presented in Chen (1995) Chen & Mangasarian (1995*a*), Gabriel & Pang (1992), Han et al. (1992), Pang (1990), Pang (1991), Pang, Han & Rangaraj (1991), and Pang & Gabriel (1993).

It is easily seen that $H(x) = 0$ if and only if x solves NCP(f). However, the (Fréchet) derivative of the function is not defined whenever $x_i = f_i(x)$ for some i, so classical Newton methods cannot be employed. Two different approaches have been taken to handle this difficulty. In Chen & Mangasarian (1995*a*), a solution is found by solving a sequence of smooth approximations to the equation $H(x) = 0$. This technique was implemented in a computer code called SMOOTH. Computational tests on this software (see Chapter 3) demonstrate that this approach is remarkably robust, as well as extremely efficient. We note that smoothing techniques have also been studied by Chen & Harker (1993), Chen & Harker (1995), Kanzow (1994), and Qi & Chen (1993).

Another technique incorporates the notion of a Bouligand derivative, or B-derivative for short (Robinson 1987)

Definition 1.3.1 *A function $H : X \to \boldsymbol{R}^n$ is said to be* B-differentiable *at a point x if H is Lipschitz continuous in a neighborhood of x and there exists a positively homogeneous function $BH(x) : \boldsymbol{R}^n \to \boldsymbol{R}^n$, such that*

$$\lim_{v \to 0} \frac{H(x+v) - H(x) - (BH(x))(v)}{\|v\|} = 0.$$

The function $BH(x)$ is called the B-derivative *of f at x. f is said to be B-differentiable in a set X if f is B-differentiable at all points $x \in X$.*

Shapiro (1988) showed that if H is locally Lipschitz, then it is B-differentiable at a point x if and only if it is directionally differentiable at x. When it exists, the B-derivative is unique (Pang 1990, Proposition 1). Thus, using the B-derivative, Newton's method can be generalized by replacing (2) by

$$H(x^k) + (BH(x^k))(d) = 0. \tag{5}$$

If this equation has a solution d^k, it is shown in Pang (1990) that if $H(x^k) \neq 0$, then d^k is a descent direction for $\theta := \|H(\cdot)\|^2/2$.

Pang (1990) uses this approach, in conjunction with a linesearch strategy to produce an algorithm that is globally convergent under certain relatively strong regularity conditions. Local Q-quadratic convergence for this approach was proved in Pang (1991), again under fairly strong regularity conditions.

It is also possible to use B-derivatives to form a generalization of the Gauss-Newton method by replacing (3) with

$$\begin{array}{cc} \min & \frac{1}{2}\left\|H(x^k) + BH(x^k)d\right\|^2 \\ \text{subject to} & x^k + d \geq 0. \end{array} \tag{6}$$

This generalization is the basis for the NE/SQP algorithm (Pang & Gabriel 1993, Gabriel & Pang 1992). Pang and Gabriel proved global convergence and local Q-quadratic convergence for this algorithm using weaker regularity conditions than were needed for the Newton-based approach. A desirable feature of this algorithm is that the Gauss-Newton subproblems are always solvable, thus contributing to the robustness of the algorithm. However, as we shall show in Chapter 2, the robustness of the NE/SQP algorithm is disappointing.

The minimum map reformulation is the basis for the algorithms QPCOMP and PROXI, which will be presented in Chapters Chapter 2 and Chapter 3. In Chapter 2, we define a generalization of the minimum map whose zeros are solutions to the MCP. Using this reformulation, we develop the QPCOMP algorithm, which is based largely on NE/SQP. This algorithm improves on NE/SQP in two ways.

First, it generalizes the NE/SQP method to the MCP framework, and second, it is considerably more robust than NE/SQP. In Chapter 3, we present an algorithm called PROXI, which is similar to QPCOMP, however, it used a Newton method rather than a Gauss-Newton method to solve the generalized minimum map.

The second reformulation we shall consider is based on the semismooth function $\phi : \mathrm{R}^2 \to \mathrm{R}$ defined by

$$\phi(a,b) = \sqrt{a^2+b^2} - (a+b).$$

This function was first introduced by Fischer (1992), and has been further studied by Facchinei & Soares (1994), Facchinei & Soares (1995), Geiger & Kanzow (1994), Kanzow (1994), Kanzow (1995), Pang & Qi (1993), Qi (1993), Qi & Sun (1993), Qi & Jiang (1994), and Tseng (1994). An up to date review of the uses of this function is given by Fischer (1995).

This function has the property that

$$\phi(a,b) = 0 \Longleftrightarrow a \geq 0, b \geq 0, ab = 0. \tag{7}$$

Note also that ϕ is continuously differentiable everywhere except at the origin.

Several algorithms for solving NCP(f) have been proposed based on the ϕ function (Facchinei & Soares 1994, Facchinei & Soares 1995, Geiger & Kanzow 1994, Kanzow 1994). These algorithms assume that f is continuously differentiable on all of R^n. The NCP is then reformulated as the zero finding problem $\Phi(x) = 0$, where $\Phi : \mathrm{R}^n \to \mathrm{R}^n$ is defined by

$$\Phi_i(x) := \phi(x_i, f_i(x)). \tag{8}$$

Based upon (7), it is easily seen that x solves NCP(f) if and only if $\Phi(x) = 0$.

This semismooth reformulation has two advantages over the minimum map. First, it always incorporates both the boundary and the function information at every iteration. This is in marked contrast to the minimum map, which uses information only from one or the other. Second, the natural merit function $\theta(x) := \|H(x)\|^2/2$ is continuously differentiable. De Luca et al. (1995) use this reformulation to propose an algorithm that is proven to be globally convergent and locally Q-quadratically convergent based on considerably weaker regularity conditions than those required by the NE/SQP method.

In Chapter 3 we shall generalize the function Φ to the MCP framework. We will then use this generalization to develop a a Newton-based algorithm for the MCP, which is considerably more robust than the original algorithm of De Luca et al. (1995)

The third reformulation we consider is of interest not as a basis for a nonlinear MCP algorithm, but rather as motivation for the algorithms presented in Chapters

4 and 5, which solve the LMCP and the affine generalized equation, respectively. This reformulation is based upon the normal map (Eaves 1971, Minty 1962, Robinson 1992). Given a closed convex set $C \subset \mathrm{R}^n$ and a function $f : C \to \mathrm{R}^n$, the normal map $F_C : \mathrm{R}^n \to \mathrm{R}^n$ is defined by

$$F_C(x) := f(\pi_C(x)) + x - \pi_C(x). \tag{9}$$

In the case where $C := \mathbb{B}$, finding a zero of this equation is equivalent to solving MCP$(f, \mathbb{B})$ in the following sense: if $F_{\mathbb{B}}(x) = 0$, then $z := \pi_{\mathbb{B}}(x)$ solves MCP$(f, \mathbb{B})$. Conversely, if z is a solution to MCP$(f, \mathbb{B})$, then $x := z - f(z)$ is a zero of $F_{\mathbb{B}}$.

Several algorithms have been developed based on this normal map formulation. Harker & Xiao (1990) present a B-differentiable equation approach to finding a zero of the normal map. However, the convergence theory for this algorithm is somewhat restrictive.

A more successful approach has been based upon the well-established theory of generalized equations (Robinson 1980). The basic idea is to generalize Newton's method by solving a sequence of piecewise-linear approximations to $F_{\mathbb{B}}$, which are formed by replacing f with its first order Taylor approximation at the current iterate. Specifically, at the kth iteration, the Newton point $\bar{x}$ is calculated by solving the piecewise affine equation $M^k_{\mathbb{B}}(x) = 0$, where

$$M^k_{\mathbb{B}}(x) := f(\pi_{\mathbb{B}}(x^k)) + \nabla f(x^k)^\top(\pi_{\mathbb{B}}(x) - \pi_{\mathbb{B}}(x^k)) + x - \pi_{\mathbb{B}}(x).$$

But this is equivalent to solving LMCP$(M^k, q^k, \mathbb{B})$, where

$$M^k := \nabla f(x^k)^\top \quad \text{and} \quad q^k := f(\pi_{\mathbb{B}}(x^k)) - \nabla f(x^k)^\top \pi_{\mathbb{B}}(x^k).$$

Thus, at each iteration, the Newton point is calculated by solving an LMCP that approximates the nonlinear problem near the current iterate. It is for this reason that this approach is often referred to as a successive linear complementarity problem (SLCP) method (Mathiesen 1987, Mathiesen 1985).

Josephy (1979) proved local quadratic convergence for an algorithm based on the above ideas, assuming a strong regularity condition (Robinson 1980) at a solution. However, global convergence results for Josephy's algorithm were not established. To achieve global convergence, a globalization strategy is needed. Rutherford (1993) describes an algorithm called MILES that globalizes the Josephy method by performing a simple linesearch between the current iterate and the Newton point. This approach works well in practice, but is theoretically unjustified since the linesearch direction is not necessarily a descent direction for the natural merit function given by $\theta(x) := \left\|F_{\mathbb{B}}(x)\right\|^2/2$.

A theoretically sound approach was proposed by Ralph (1994). In this approach, a *pathsearch* is performed, seeking reduction of the merit function along a piecewise linear path connecting the current iterate with the Newton point. This path is exactly the search path that was constructed by the LMCP solver in solving the linear subproblem. Following this path is guaranteed to produce descent of the merit function as long as the current iterate is not a stationary point of the merit function. Thus, Ralph was able to prove global convergence of this algorithm.

A sophisticated version of Ralph's pathsearch algorithm has been implemented in a computer code called PATH (Dirkse 1994, Dirkse & Ferris 1995*b*, Dirkse & Ferris 1995*c*). This algorithm is comparable to SMOOTH in terms of robustness and efficiency, as will be shown in Chapter 3.

Due in large part to the success of PATH and MILES, the economics community has been developing extremely large economic models in the mixed complementarity framework (Harrison, Rutherford & Tarr 1995). Because of this, it is clear that techniques for improving the efficiency of these algorithms on very large scale problems need to be explored. One possible avenue is to use interior-point methods to solve the LCP subproblems. But for such a technique to be useful, the interior-point method needs to be easily warm-started, since the solution to one subproblem will often be near the solution to the next. This consideration is what motivated Chapter 4 of this thesis, which presents an infeasible interior-point method for solving the LMCP.

1.3.3 Robustness Issues

All of the algorithms discussed so far are examples of *descent methods*, which work to minimize the merit function. As such, their global convergence behavior is limited by a fundamental difficulty: the merit function may have local minima that are not solutions of the complementarity problem. This difficulty manifests itself in different ways for different algorithms. In PATH and MILES, it arises as a rank-deficient basis or as a linear complementarity subproblem that is not solvable. In SMOOTH, it appears as a singular Jacobian matrix. In NE/SQP it arises as convergence to a point that fails some regularity condition.

Because of this difficulty, the best these algorithms can hope for, in terms of global convergence behavior, is to guarantee finding a solution only when the merit function has no strict local minimizers that are not global minimizers. In general, this means that the function f must be monotonic in order to guarantee convergence from arbitrary starting points.

Another class of algorithms is based on numerical continuation. Examples of this class are given in Pang & Wang (1990), Reinoza (1985), Sellami (1994), Watson (1979). Theoretically, these algorithms are extremely robust, depending only on very weak assumptions. However, they do not have the fast local convergence rates enjoyed by the Newton-based algorithms.

The challenge then, is to develop an algorithm that rivals the continuation methods in terms of robustness, but which is still competitive with the descent methods in terms of local convergence rates. This challenge is addressed in Chapters 2 and 3, where a proximal perturbation strategy is proposed and implemented. This strategy allows Newton-based methods to achieve global convergence depending only on a pseudo-monotonicity assumption at a solution. Under this pseudo-monotonicity assumption, convergence from arbitrary starting points is achieved even in the presence of strict local minima of the merit function.

1.4 Solution Methods for Linear Problems

This section describes two common techniques for solving linear complementarity problems: interior point methods, and pivoting methods. These two techniques serve as the basis for the algorithms presented in Chapters 4 and 5. In describing these methods, it is convenient to state the LCP in an alternative form given by:

$$\begin{array}{ll} \text{LCP}(M,q): & \text{Find } (x,y) \in \mathrm{R}^n_+ \times \mathrm{R}^n_+ \text{ such that} \\ & y = Mx + q, \quad \text{and } \langle x, y \rangle = 0. \end{array}$$

1.4.1 Interior Point Methods

Since the landmark work of Karmarkar (1984), the mathematical programming community has witnessed an explosion of research devoted to interior point methods. The reason for this enthusiasm is that interior point algorithms have been demonstrated to have polynomial complexity, which means that the worst-case execution times of these algorithms is a polynomial function of the problem size. This contrasts sharply with the exponential complexity of pivoting schemes. But this is not just a theoretical advantage; when put into practice, run times for interior-point algorithms have been observed to grow relatively slowly as problem sizes are increased (Lustig, Marsten & Shanno 1994). Thus, interior point methods are viewed by many as the method of choice for very large scale problems.

Most of the early work on interior-point methods was devoted to solving linear programming problems. However, more recently, these techniques have also been shown to be effective for monotone linear complementarity problems. There are many variations of interior point methods. The algorithm we shall develop in Chapter 4 is an example of a path-following method. The basic idea behind path-following methods is to reformulate the LCP as a minimization problem augmented with a barrier term to ensure that the iterates remain strictly feasible. For example, LCP(f) may be approximated by the following minimization problem:

$$\text{MP}(\mu): \qquad \begin{array}{cc} \min & \langle x, y\rangle - \mu \sum_{i=1}^{n} (\log x_i + \log y_i) \\ \text{subject to} & y = Mx + q. \end{array}$$

For any positive value of μ, the log terms force the solution $(x(\mu), y(\mu))$ of the above minimization problem to be strictly positive. As μ is decreased continuously to 0, the corresponding solutions $(x(\mu), y(\mu))$ to MP(μ) trace out a *central path* that leads to the solution $(\bar{x}, \bar{y})$ of the LCP. The aim of the path-following method is to follow this central path to the solution. The algorithm is started with a positive value of μ, and a strictly feasible point $(x^0, y^0) \in \mathbf{R}^n_{++} \times \mathbf{R}^n_{++}$, which satisfies the equality constraint $y^0 = Mx^0 + q$. A sequence of strictly feasible iterates is then generated, where at each iteration, a single constrained damped Newton step is taken toward the solution of MP(μ), and then the value of μ is decreased.

Until recently, interior-point methods were not amenable to warm-starting. This was unfortunate since our main interest in solving linear complementarity problems is in the context of solving nonlinear problems. In particular, SLCP schemes work by solving a sequence of LCP subproblems, whose solutions converge to the solution of the nonlinear problem. In general, the solution to one LCP is very near the solution to the next. Thus, we would like to warm-start the LCP algorithm with this "good" starting point.

The reason interior point methods could not be easily warm-started was that the starting point needed to be strictly feasible, meaning that it needed to strictly satisfy the inequality constraints $(x, y) \geq 0$ as well as satisfying the equality constraints $y = Mx + q$. Unfortunately, in an SLCP scheme, the solution to one subproblem is not, in general, feasible for the next subproblem. Thus, it would need to be modified to make it feasible, a process that would generally carry it far away from the solution.

More recently, *infeasible* interior-point methods have been developed, which differ from (feasible) interior-point methods by allowing the iterates to violate the equality constraints while strictly satisfying the inequality constraints. Put differently, the iterates are infeasible with respect to the equality constraints, but are interior to the region defined by the inequality constraints. We call this region the *inequality-feasible region* or more simply the *i-feasible region*.

A typical implementation of an infeasible interior-point method replaces the minimization problem MP(μ) with the following unconstrained problem:

$$\min \langle x, y \rangle + \|y - Mx + q\| - \mu \sum_{i=1}^{n} (\log x_i + \log y_i)$$

In this problem, there is no requirement that the equality constraints $y = Mx + q$ be satisfied until the solution is found. The relaxation of the feasibility requirements allows these infeasible interior-point methods to handle warm starts quite effectively. Typically, the problems can be formulated in such a way that the only difference between the current problem and the nearby problem is in the equality constraints. Thus, with only a slight modification, the solution to the nearby problem can be used as the starting point for the current problem. This slight modification is needed simply to move the point from the boundary of the i-feasible region to the interior of the i-feasible region, and can be arbitrarily small.

A significant amount of work has been devoted to the development of infeasible interior-point algorithms. This line of research first produced practical algorithms along with numerical tests and comparisons, which demonstrated superior practical performance of this class of algorithms. (Anstreicher 1991, Anstreicher 1989, Lustig, Marsten & Shanno 1991, Lustig, Marsten & Shanno 1992, McShane, Monma & Shanno 1989, Mehrotra 1992, Kojima, Mizuno & Todd 1992, Mizuno 1993, Potra 1992*a*, Potra 1992*c*).

More recently, a number of theoretical papers have been written which analyze convergence and complexity behavior of various algorithms (Anitescu, Lesaja & Potra 1994, Kojima, Megiddo & Mizuno 1991, Kojima, Shindo & Hara 1994, Mizuno, Jarre & Stoer 1994, Potra 1994, Potra & Bonnans 1994, Potra & Sheng 1994*b*, Potra & Sheng 1994*a*, Shanno & Simantiraki 1995, Stoer 1994, Wright 1994, Wright 1993, Wright & Zhang 1994, Zhang 1994, Zhang & Zhang 1992).

Of particular importance is the paper by Zhang (1994). In it, Zhang demonstrates global Q-linear convergence and polynomial complexity for a class of infeasible interior-point methods for a generalization of the linear complementarity problem called the horizontal linear complementarity problem. This work is particularly significant because the class of algorithms Zhang analyzes is closely related to already existing algorithms with proven effectiveness. More recently Wright (1994) and Wright (1993) extended Zhang's algorithm to produce two algorithms that achieve local Q-subquadratic convergence.

Unfortunately, both Zhang and Wright place a restriction on the starting point that will pose problems when warm-starting the algorithms. Their restriction is very easy to satisfy if we are allowed to vary the starting point. However, this completely defeats the purpose of warm-starting, since changing the starting point may take us farther away from the solution. Fortunately, this restriction on the starting points is unnecessary. The proof of this fact is one of the main results of Chapter 4. After this restriction is removed, it will then be a straightforward task to generalize Wright's algorithm to solve the LMCP.

The issue of relaxing the restrictions on the starting points for infeasible interior-point algorithms has been addressed in several other papers. For linear programming, global convergence from any positive starting point is proved by Potra (1992*b*). A similar result is given by Potra & Sheng (1994*b*) for the linear complementarity problem. Monteiro & Wright (1993) propose an algorithm for the linear complementarity problem that achieves superlinear convergence from any positive starting point. Their analysis is significant in that they prove superlinear convergence without assuming the existence of a strictly feasible point. Another algorithm by Wright (1993) achieves similar convergence results for the mixed monotone linear complementarity problem.

Güler (1995) proves that any interior-point algorithm for the linear complementarity problem can be extended to an interior-point algorithm for the horizontal linear complementarity problem with identical convergence results. Thus, in principle, any of the methods cited above could be extended to the horizontal linear complementarity problem to provide an algorithm that converges from arbitrary starting point. However, because of the fundamental importance of Zhang's algorithm, a direct generalization of his analysis is appropriate.

1.4.2 Pivoting Methods

The final class of algorithms we are interested in is the pivoting methods. The most famous of these approaches is Lemke's method (Lemke & Howson 1964). Numerous other pivoting strategies have been proposed, which are described in detail by Cottle, Pang & Stone (1992). Perhaps the most important of these methods are the principal pivoting algorithms, which were originally proposed by Zoutendijk (1960) and Bard (1972), and are described in their general form by Cottle (1966), Cottle (1968), Cottle & Dantzig (1968), and Dantzig & Cottle (1967). However, Lemke's method is still the preferred technique.

Lemke's method is typically described in terms of generating a sequence of almost complementary vector pairs $\{(y^k, x^k)\}$ that are extreme points of the feasible region $\{(y, x) \mid y \geq 0, x \geq 0, y = Mx + q\}$. The iterates are updated by moving within the kernel of a basis matrix $B^k \in \mathbb{R}^{n \times n}$ until a new boundary of the feasible region is encountered. At each iteration a "pivot" is performed that updates the basis matrix. A complete and elegant description is provided in Cottle & Dantzig (1968).

For our purposes, a more geometric view of Lemke's method is preferable. The background for this was provided by Eaves' work on piecewise linear homotopies

(Eaves 1976). To describe Lemke's method in the homotopy framework, we first reformulate LCP(M, q) as a zero-finding problem of the normal map $F_{\mathbb{B}}$, where $\mathbb{B} := \mathbb{R}^n_+$:

$$F_{\mathbb{B}}(x) = Mx_+ + q - x_-.$$

As was discussed in Section 1.3, if $F_{\mathbb{B}}(x) = 0$, then $(z, y) := (x_+, Mx_+ + q)$ is a solution of LCP(M, q), and conversely, if (z, y) is a solution to LCP(M, q), then $x := z - y$ is a zero of $F_{\mathbb{B}}$. To find a zero of $F_{\mathbb{B}}$, a homotopy mapping is constructed in which an additional variable is added to produce a piecewise-linear function $\phi : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$. This mapping is defined as follows:

$$\phi(x, \lambda) := F_{\mathbb{B}}(x) + \lambda e.$$

(Recall that e is the vector of all ones). The strategy is then to track the zero curve of ϕ starting from a trivial solution until a point $(\bar{x}, \bar{\lambda})$ is reached with $\bar{\lambda} = 0$. At this point $\bar{x}$ is a zero of $F_{\mathbb{B}}$.

In the terminology of Eaves, the domain of ϕ can be subdivided into a finite collection of $(n + 1)$-dimensional cells whose interiors are disjoint, such that the restriction of ϕ to each cell is an affine function. These cells are characterized by the signs of the components of x. For example, if $n = 2$, there are four cells given by

$$\begin{aligned}
\sigma_1 &:= \{(x, \lambda) \mid x_1 \geq 0, x_2 \geq 0, \lambda \geq 0\} \\
\sigma_2 &:= \{(x, \lambda) \mid x_1 \leq 0, x_2 \geq 0, \lambda \geq 0\} \\
\sigma_3 &:= \{(x, \lambda) \mid x_1 \geq 0, x_2 \leq 0, \lambda \geq 0\} \\
\sigma_4 &:= \{(x, \lambda) \mid x_1 \leq 0, x_2 \leq 0, \lambda \geq 0\},
\end{aligned}$$

and

$$\begin{aligned}
\phi_{|\sigma_1}(x, \lambda) &:= Mx + \lambda e + q \\
\phi_{|\sigma_2}(x, \lambda) &:= \begin{bmatrix} I_{\cdot 1} & M_{\cdot 2} \end{bmatrix} x + \lambda e + q \\
\phi_{|\sigma_3}(x, \lambda) &:= \begin{bmatrix} M_{\cdot 1} & I_{\cdot 2} \end{bmatrix} x + \lambda e + q \\
\phi_{|\sigma_4}(x, \lambda) &:= Ix + \lambda e + q
\end{aligned}$$

In this example, the matrices multiplying x correspond to the basis matrices B^k in our first description of Lemke's method.

Eaves (1976, Theorem 9.1) shows that if 0 is a regular value (Eaves 1976, Section 8), then $\phi^{-1}(0)$ is a 1-manifold whose endpoints lie on the boundary of the domain of ϕ. A 1-manifold is a disjoint union of routes and loops, where a route is a curve homeomorphic to an interval, and a loop is a curve homeomorphic to a circle.

It follows then that if 0 is a regular value, then tracking the zero curve from a given starting point can have one of three possible outcomes: either 1) the zero curve will loop back to the original point, 2) the zero curve will terminate in a ray, or 3) the zero curve will terminate at a point on the boundary of $\mathrm{dom}(\phi)$. In this last case, a solution is found since any boundary point will necessarily have $\lambda = 0$.

To eliminate the first possibility, a starting point is chosen that is the endpoint of a ray. This is accomplished by choosing the starting point (x^0, λ_0) such that $\lambda_0 \geq \max_i(-q_i)$, and $x^0 := -q - \lambda_0 e$. Observe that by defining $d^0 := (-e, 1)$, then the ray $\{(x^0, \lambda_0) + \mu d^0 \mid \mu > 0\}$ lies on the zero curve of ϕ. Note further that this ray is entirely contained within the cell characterized by x nonpositive. Thus, in the example above, this ray is contained in the cell σ_4. It is clear then that the starting point lies on a component of the zero curve that is not a loop. Thus, tracking this zero curve will either result in ray termination, or in termination at a solution.

To trace the zero curve we start by moving in the $-d^0$ direction until a boundary of the starting cell is reached (i.e., until either $\lambda = 0$ or $x_i = 0$ for some i). If this boundary point is not a solution, then it must be shared by another cell. Furthermore, under the regularity assumption, it is shared by only one other cell. In this case, a new direction d^1 is chosen that lies in the kernel of this new cell. We then move along this direction until a new boundary is encountered. The process continues until it terminates at either a ray or a solution. When 0 is not a regular value, then the above process can be enhanced by a lexicographic ordering.

One characteristic of the above process is that in moving between adjacent cells, the ϕ function differs only by a rank-1 update. Thus, the matrix can be stored in factored form, and the factors can be updated cheaply at each iteration.

The foundation laid by Eaves (1976) provided a far more powerful theory than was required for Lemke's method. Using this theory, it is possible to develop similar algorithms for more complex problems. Such an algorithm was devised for affine variational inequalities by Cao & Ferris (1995*b*). In Chapter 5 we extend this work to produce an algorithm for affine generalized equations.

1.5 Organization

So far, we have introduced several problem classes and provided an overview of some of the relevant techniques for addressing these problems. The remainder of this thesis expands upon these ideas, presenting five new algorithms for complementarity problems and generalized equations. In Chapter 2 a proximal perturbation strategy is presented in detail and a global convergence result is proven that depends only on a pseudomonotonicity assumption. This strategy is then applied to the NE/SQP algorithm of Pang & Gabriel (1993). To do this, the NE/SQP algorithm is first extended to solve problems in the MCP framework, and then modified to guarantee finite termination. Finally, the QPCOMP algorithm is presented along with a formal proof of global convergence.

In Chapter 3 the proximal perturbation strategy is applied to two additional Newton-based algorithms. The first is a very simple damped Newton scheme that, like the NE/SQP algorithm, uses a generalization of the minimum map to reformulate the MCP. The second algorithm is a semismooth equations approach that uses the second reformulation described in Section 1.3. Extensive computational results are then given comparing the performance of these algorithms to the current state-of-the-art algorithms PATH (Dirkse & Ferris 1995*b*) and SMOOTH (Chen & Mangasarian 1995*a*)

In Chapter 4, we present an infeasible interior-point algorithm for the linear mixed complementarity problem. To do this we first extend some theoretical results of two previous infeasible interior-point algorithms, proving global and local Q-subquadratic convergence from arbitrary starting points. We then use these results to prove convergence of our algorithm for the linear mixed complementarity problem.

Finally, in Chapter 5, we develop a path-following algorithm for the affine generalized equation. This algorithm is an extension of an algorithm for affine variational inequalities that was proposed by Cao & Ferris (1995*b*). We prove that under an assumption of coherent orientation, the algorithm will find a solution in a finite number of iterations.

Chapter 2

QPCOMP

In this chapter, we present QPCOMP: a Quadratic-Programming based solver for mixed COMPlementarity solvers. This algorithm is the first of three algorithms using a proximal perturbation strategy that will be presented in Section 2.1. This strategy provides a means of enhancing any algorithm that reliably solves strongly monotone MCPs so that it will solve a much broader class of problems. In particular, it will solve any problem that satisfies a pseudo-monotonocity condition at a solution. This result will be proved in Theorem 2.1.3 based upon exact solutions to a sequence of perturbed subproblems. Applying this strategy to the NE/SQP algorithm of Pang & Gabriel (1993) results in the QPCOMP algorithm.

As mentioned in the introduction, the robustness of NE/SQP is disappointing when compared to PATH and SMOOTH. In fact, we shall show in Section 2.2 that NE/SQP cannot reliably solve even one-dimensional monotone linear complementarity problems! However, NE/SQP works well on strongly monotone problems, which is all that is required for our strategy to work.

Before applying this strategy to NE/SQP, it is first necessary to extend the NE/SQP algorithm to the MCP framework. We do this in Section 2.2, where we present the algorithm and also duplicate all of the convergence results given in Pang & Gabriel (1993). In addition, we extend these convergence results to prove that the NE/SQP algorithm will solve any strongly-monotone MCP. Finally, we introduce some modifications to the algorithm to ensure that it will terminate in a finite number of iterations, even when it fails.

We will be referring to the analysis of Pang & Gabriel (1993) extensively. As much as possible our notation is identical to the notation used by Pang & Gabriel. The only difference is in the meaning of $\nabla f(x)$. If f is a function from R^n to R^m, $\nabla f(x)$ is always a $n \times m$ matrix, whereas, in Pang & Gabriel (1993), it is an $m \times n$ matrix, except when $m = 1$.

In Section 2.3, we present the QPCOMP algorithm. The main convergence result for this algorithm is given in Theorem 2.3.1, which shows global convergence under the assumption of pseudo-monotonicity at a solution, whenever f is a Lipschitz continuous, continuously differentiable function. Finally, some test results will be given in Section 2.4, which demonstrate that QPCOMP is significantly more robust than NE/SQP. Throughout this chapter we assume that f is continuously differentiable on an open set $\Omega \supset \mathrm{B}$.

2.1 The Proximal Perturbation Strategy

In this section we present a strategy for taking algorithms which work well on strongly monotone MCPs and extending them to solve MCPs for which a considerably weakened monotonicity condition is satisfied. To state this condition, we first need to define the concept of *pseudo-monotonicity*:

Definition 2.1.1 *Given a set $B \subset R^n$, the mapping $f : \Omega \to R^n$ is said to be* pseudo-monotone *at a point $x^* \in B$ if $\forall y \in B$,*

$$\langle f(x^*), y - x^* \rangle \geq 0 \quad \textit{implies} \quad \langle f(y), y - x^* \rangle \geq 0. \tag{1}$$

f is said to be pseudo-monotone on B *if it is pseudo-monotone at every point in B.*

It is known (Harker & Pang 1990) that if a function $g : \mathrm{R}^n \to \mathrm{R}$ is pseudo-convex (Mangasarian 1969, Definition 9.3.1), then ∇g is a pseudo-monotone function. However, if g is only pseudo-convex at a point x^*, it does not necessarily follow that ∇g is pseudo-monotone at x^*.

Pseudo-monotonicity is a weaker condition than monotonicity. In particular every monotone function is pseudo-monotone. But the converse is not true. For example, consider the function $f(x) := x/2 + \sin(x)$. This function is pseudo-monotone, but is not monotone. Note further that the natural merit function $\|f(x)\|^2/2$ has strict local minima that are not global minima. Thus, we see that the natural merit function of a pseudomonotone function can have local minima that are not global minima.

In order to guarantee global convergence of our algorithm we shall require that the following assumption be satisfied:

Assumption 2.1.2 *MCP(f, B) has a solution x^* such that f is pseudomonotone at x^*.*

If MCP(f, B) satisfies Assumption 2.1.2, we say that MCP(f, B) is *pseudo-monotone at a solution*. However, for simplicity, we will abuse terminology somewhat and say simply that MCP(f, B) is *pseudo-monotone*. This should not cause any confusion since all of our discussions will refer to problems which satisfy Assumption 2.1.2.

The strategy we present for pseudo-monotone MCPs is based upon extending a descent-based algorithm for strongly monotone MCPs. The idea behind a descent-based algorithm is to minimize a nonnegative merit function $\theta : \mathrm{B} \to \mathrm{R}_+$. If f is strongly monotone, it is easy to construct a merit function that has a global minimum, but no other local minima. It is then a simple task to find the global minimizer of θ, thereby giving a solution to the MCP. If however f is not monotone, then the merit function chosen will, in all likelihood, contain local minima for which $\theta \neq 0$. The algorithm may then terminate at such a local minimum, rather than at the solution.

To overcome this difficulty, we would like to find some way to "escape" from this local minimum. This can be accomplished by constructing an improved starting point $\tilde{x}$ where $\theta(\tilde{x})$ is smaller than the value of θ at the local minimum. Since the descent-based algorithm never allows the value of θ to increase, the algorithm can be restarted from $\tilde{x}$ with the guarantee that it will never return to the local minimum. Obviously, finding such an improved starting point is not a straightforward task. However, this can be achieved when the problem is pseudo-monotone. The remainder of this section describes how to construct this improved starting point.

We begin by defining a particular merit function for our algorithm: To do this, we first introduce the mapping $H : \Omega \to \mathrm{R}^n$ defined by

$$H_i(x) := \min(x_i - l_i, \max(x_i - u_i, f_i(x))). \tag{2}$$

It is easily shown that $H(x) = 0$ if and only if x solves MCP(f, B). Using this function, we define the merit function

$$\theta(x) := \frac{1}{2} H(x)^\top H(x). \tag{3}$$

Clearly, x is a solution to MCP(f, B) if and only if x is a minimizer of θ with $\theta(x) = 0$.

In Section 2.2 we will present a basic algorithm for solving strongly monotone MCPs, which is based on minimizing this particular choice of θ. However, for now, we simply assume that such an algorithm exists. Moreover we assume that the algorithm will fail in a *finite number* of iterations whenever it cannot solve the problem.

Now suppose the basic algorithm fails at a point x^0. Our strategy will be to solve a sequence of perturbed problems, generating a sequence of solutions $\{x^k\}$ that leads to an improved starting point $\tilde{x}$. The perturbed problems we solve are based on the following perturbation of f: given a *centering point* $\bar{x} \in \mathrm{B}$, and a number $\lambda > 0$, let

$$f^{\lambda,\bar{x}}(x) := f(x) + \lambda(x - \bar{x}).$$

If f is Lipschitz continuous, then for λ large enough, $f^{\lambda,\bar{x}}$ is strongly monotone. Thus, the basic algorithm will be able to solve the perturbed problem $\mathrm{MCP}(f^{\lambda,\bar{x}}, \mathrm{B})$.

With a sufficiently large λ we can then generate a sequence of iterates as follows: given a point x^0, then for $k = 0, \ldots$, choose x^{k+1} as the solution to $\mathrm{MCP}(f^{\lambda,x^k}, \mathrm{B})$. Note that every subproblem in the sequence uses the same choice of λ, but a different choice of centering point. In particular the centering point for one subproblem is the solution of the previous subproblem. This is very reminiscent of the proximal point algorithm (Rockafellar 1978) and of Tikhonov regularization (Tikhonov & Arsenin 1977).

The following lemma gives sufficient conditions for a subsequence of these iterates to converge to a solution of $\mathrm{MCP}(f, \mathrm{B})$.

Theorem 2.1.3 *Let $\lambda > 0$ and let $\{x^k\}, k = 0, 1, \ldots$ be a sequence of points in B such that for each k, x^{k+1} is a solution to MCP(f^{λ,x^k}, B). If MCP(f, B) satisfies Assumption 2.1.2, then*

1. *$\{x^k\}$ has a subsequence that converges to a solution $\bar{x}$ of MCP(f, B).*
2. *Every accumulation point of $\{x^k\}$ is a solution of MCP(f, B).*
3. *If f is pseudomonotone at any accumulation point $\bar{x}$ of $\{x^k\}$, then the iterates converge to $\bar{x}$.*

Proof Let x^* be the solution to $\mathrm{MCP}(f, \mathrm{B})$ given by Assumption 2.1.2. Since x^{k+1} is a solution to $\mathrm{MCP}(f^{\lambda,x^k}, \mathrm{B})$, then for each component i, exactly one of the following is true:

1. $x_i^{k+1} = l_i$ and $f_i(x^{k+1}) + \lambda(x_i^{k+1} - x_i^k) \geq 0$,
2. $l_i < x_i^{k+1} < u_i$ and $f_i(x^{k+1}) + \lambda(x_i^{k+1} - x_i^k) = 0$,
3. $x_i^{k+1} = u_i$ and $f_i(x^{k+1}) + \lambda(x_i^{k+1} - x_i^k) \leq 0$,

Let I_l, I_f and I_u be the sets of indices which satisfy the first, second, and third conditions respectively.

For $i \in I_l$, it follows that $0 \leq x_i^k - x_i^{k+1} \leq f_i(x^{k+1})/\lambda$. Also, $x_i^{k+1} - x_i^* = l_i - x_i^* \leq 0$, so

$$(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) \geq f_i(x^{k+1})(x_i^{k+1} - x_i^*)/\lambda. \tag{4}$$

By similar reasoning, this inequality holds for $i \in I_u$. Finally, for $i \in I_f$, $f_i^{\lambda,x^k}(x^{k+1}) = 0$, so $x_i^k - x_i^{k+1} = f_i(x^{k+1})/\lambda$, whereupon it follows that (4) is satisfied as an equality.

Thus in all cases, inequality (4) is satisfied, which gives us the following.

$$\begin{aligned}
(x_i^k - x_i^*)^2 &= (x_i^{k+1} - x_i^* + x_i^k - x_i^{k+1})^2 \\
&= (x_i^{k+1} - x_i^*)^2 + 2(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) + (x_i^k - x_i^{k+1})^2 \\
&\geq (x_i^{k+1} - x_i^*)^2 + 2f_i(x^{k+1})(x_i^{k+1} - x_i^*)/\lambda + (x_i^k - x_i^{k+1})^2 \quad \text{by (4).}
\end{aligned}$$

Summing over all components, we obtain

$$\left\|x^k - x^*\right\|^2 \geq \left\|x^{k+1} - x^*\right\|^2 + 2\left\langle f(x^{k+1}), x^{k+1} - x^*\right\rangle/\lambda + \left\|x^k - x^{k+1}\right\|^2.$$

Under Assumption 2.1.2, the inner product term above is nonnegative. Thus,

$$\left\|x^k - x^*\right\|^2 \geq \left\|x^{k+1} - x^*\right\|^2 + \left\|x^k - x^{k+1}\right\|^2,$$

so $\{\left\|x^k - x^*\right\|\}$ is a decreasing sequence, and $\left\|x^k - x^{k+1}\right\| \to 0$. It follows that $\{x^k\}$ has an accumulation point. Let $\bar{x}$ be any accumulation point of $\{x^k\}$. Then there is a subsequence $\{x^{k_j} : j = 0, 1, \ldots\}$ converging to $\bar{x}$. Since $\left\|x^k - x^{k+1}\right\| \to 0$, we also see that $x^{k_j+1} \to \bar{x}$. Finally, since x^{k_j+1} solves MCP$(f^{\lambda,x^{k_j}}, \mathrm{B})$, we have

$$\min\left(x_i^{k_j+1} - l_i, \max\left(x_i^{k_j+1} - u_i, f_i(x^{k_j+1}) + \lambda(x_i^{k_j+1} - x_i^{k_j})\right)\right) = 0, \quad \forall i.$$

By continuity, $\min(\bar{x}_i - l_i, \max(\bar{x}_i - u_i, f_i(\bar{x}))) = 0, \quad \forall i$; that is, $\bar{x}$ solves MCP(f, B). This proves items 1 and 2

To prove item 3, note that if f is pseudomonotone at an accumulation point $\bar{x}$, then by item 2, $\bar{x}$ is a solution, so the above analysis can be repeated with x^* replaced by $\bar{x}$. We can then conclude that $\{\left\|x^k - \bar{x}\right\|\}$ is a decreasing sequence. But since $\bar{x}$ is an accumulation point of $\{x^k\}$, it follows that $\left\|x^k - \bar{x}\right\| \to 0$. $\square$

Table 1: Iterates produced by solving sequence of perturbed problems, with ($\lambda = 1.1$)

k	x^k	$f(x^k)$	$\theta(x^k)$
0	0	-.01	.00005
1	.9110	-1.0021	.5021
2	1.5521	-.7052	.2487
3	1.8356	-.3118	.0486
4	1.9439	-.1191	.0071
5	1.9832	-.0433	.00094
6	1.9973	-.0155	.00012
7	2.0023	-.0055	.00002

Note that Theorem 2.1.3 did not make any assumptions on the choice of λ. Thus, even if λ is too small to ensure that $f^{\lambda,\bar{x}}$ is strongly monotone, the strategy will still work so long as each subproblem is solvable.

To illustrate the technique, it is useful to look at a simple example. Let $\mathrm{B} := \mathrm{R}_+$ and let $f : \mathrm{R} \to \mathrm{R}$ be defined by

$$f(x) = (x-1)^2 - 1.01.$$

This deceptively simple problem proved intractable for all of the descent-based methods we tested. In particular, we tried to solve this problem using PATH, MILES, NE/SQP, and SMOOTH. All four algorithms failed from a starting point of $x = 0$. But this should not be surprising since f is not monotone. However, f is pseudo-monotone on B. Thus, it is easily solved by our technique. For example, using $\lambda = 1.1$ and a starting point $x^0 = 0$, the strategy generates the sequence of iterates shown in Table 2.1.

Note that at the 7th iteration, an improved starting point is found, (i.e, $\theta(x^7) < \theta(x^0)$). At this point, a basic algorithm (e.g., Newton's method) can be used to obtain the final solution.

In this section, we have introduced a basic strategy for taking descent-based algorithms that solve strongly monotone MCPs, and extending them to solve pseudo-monotone MCPs. This is, in fact, the main idea presented in this chapter. However, to turn this strategy into a working algorithm, a number of details must be addressed:

1. We must ensure that the basic algorithm (for solving the strongly monotone MCPs) terminates in a finite number of iterations. This issue will be addressed in detail in Section 2.2.

2. Since we require finite termination of the basic algorithm, we must allow inexact solutions of the perturbed subproblems. We shall therefore need to incorporate control parameters into our strategy which govern the accuracy demanded by each subproblem. In the our actual implementation of the algorithm we demand very little accuracy for each subproblem. In fact, except in extreme circumstances, we allow only one step of the basic algorithm before updating the perturbed problem. To guarantee convergence of this approach requires more laborious analysis which we defer until Section 2.3.

3. Since we have no a priori information regarding the Lipschitz continuity of f, we shall have to incorporate some adaptive strategy for choosing λ in order to ensure that, eventually, the subproblems all become solvable.

The remainder of this chapter addresses these details.

2.2 Subproblem Solution

In this section, we present an algorithm for solving strongly monotone MCPs, which is based on the NE/SQP algorithm of Pang & Gabriel (1993). NE/SQP was originally developed as a method for solving the nonlinear complementarity problem. When it was first introduced, NE/SQP offered a significant advance in the robustness of NCP solvers because the subproblems it needs to solve at each iteration are convex quadratic programs, which are always solvable. Today, its robustness has been greatly surpassed by PATH, MILES, and SMOOTH (see Chapter 3). However, NE/SQP is still a viable technique for solving strongly monotone MCPs. Moreover, NE/SQP has the very desirable feature of evaluating the function f only on B. This is in marked contrast to the SMOOTH algorithm which requires f to be defined on all of $\mathbb{R}^n$.

In this section, we first present the NE/SQP algorithm extended to the MCP framework, and prove global and local convergence results for it, which parallel the results given in Pang & Gabriel (1993). We then extend these results to show that the algorithm solves all strongly monotone MCPs. Finally, we modify the algorithm to ensure finite termination.

Before we begin, we note that in his Ph.D. dissertation, Gabriel extended NE/SQP to address the upper bound nonlinear complementarity problem (UBNCP) (Gabriel 1992). The UBNCP is the special case of MCP where $l = 0$ and $u > 0$ is finite. Gabriel's algorithm used a slightly different merit function than the one we defined in (3). In particular, Gabriel's merit function is given by $\hat{\theta}(x) := \hat{H}(x)^\top \hat{H}(x)/2$, where $\hat{H}_i(x) := \min(x_i - l_i, f_i(x)_+) + \min(u_i - x_i, f_i(x)_-)$.

2.2.1 Extension of NE/SQP to the MCP Framework

Recall that a vector x solves MCP(f, B) if and only if $\theta(x) = 0$, where θ is defined by (2) and (3). The NE/SQP algorithm attempts to solve this problem by solving the minimization problem

$$\begin{array}{cc} \min & \theta(x), \\ \text{subject to} & x \in \mathrm{B}. \end{array}$$

We will use θ as a merit function for the MCP. To describe the algorithm in detail we need to partition the indices $\{1, \ldots, n\}$ into five sets as follows:

$$\begin{array}{rcl} I_l(x) & := & \{i : x_i - l_i < f_i(x)\} \\ I_{el}(x) & := & \{i : x_i - l_i = f_i(x)\} \\ I_f(x) & := & \{i : x_i - u_i < f_i(x) < x_i - l_i\} \\ I_{eu}(x) & := & \{i : x_i - u_i = f_i(x)\} \\ I_u(x) & := & \{i : x_i - u_i > f_i(x)\}. \end{array}$$

It will at times be convenient to refer also to the index sets $J_l(x) := I_l(x) \bigcup I_{el}(x)$ and $J_u(x) := I_u(x) \bigcup I_{eu}(x)$. As in the original description of NE/SQP, the subscripts of these sets are chosen to reflect their meaning. For example, the subscripts l, f, and u correspond to the indices where $H_i(x) = (x_i - l_i), f_i(x)$, and $(x_i - u_i)$ respectively. The subscripts el and eu correspond to the indices where $f_i(x)$ is equal to l_i and u_i, respectively.

These index sets are used to define an iteration function $\phi : \mathrm{B} \times \mathrm{R}^n \to \mathrm{R}$ as follows: $\phi(x, d) := \|h(x, d)\|^2 / 2$, where

$$(5) \quad h_i(x, d) := \begin{cases} (x_i - l_i + d_i) & i \in I_l(x) \bigcup I_{el}(x) \\ (x_i - u_i + d_i) & i \in I_u(x) \bigcup I_{eu}(x) \\ (f_i(x) + \nabla f_i(x)^\top d) & i \in I_f(x) \end{cases} \qquad i = 1, \ldots, n.$$

To be consistent with the notation used by Pang and Gabriel, we also define $\phi_i(x, d) := (h_i(x, d))^2/2$. Given a point $x \in \mathrm{B}$, the algorithm chooses a descent direction d by solving the convex quadratic programming problem (QP$_x$) given by

$$\begin{array}{cc} \text{minimize} & \phi(x, d) \\ \text{subject to} & x + d \in \mathrm{B}. \end{array} \qquad (\mathrm{QP}_k)$$

When discussing particular iterates of the algorithm, we shall also use the abbreviation $(\mathrm{QP}_k) := (\mathrm{QP}_{x^k})$.

We note that in the original NE/SQP algorithm, an additional constraint was added to this quadratic program, namely,

$$d_i = 0 \ \text{ if } \ f_i(x^k) = 0 \ \text{ and } \ x_i^k = l_i \ \text{ or } \ x_i^k = u_i.$$

However, this constraint is unnecessary for the convergence results, so we omit it from our algorithm.

To ensure descent of the merit function θ, we will need to perform a linesearch along the direction d^k. To describe this linesearch, we introduce a forcing function $z : \mathrm{B} \times \mathrm{R}^n \to \mathrm{R}_+$, defined by $z(x,d) := \|b(x,d)\|^2/2$, where

$$b_i(x,d) := \begin{cases} d_i & i \notin I_f(x) \\ \nabla f_i(x)^\top d & i \in I_f(x) \end{cases} \qquad i = 1, \ldots, n. \tag{6}$$

This forcing function will be used to guarantee sufficient decrease in the merit function at each iteration. Again to be consistent with the notation used by Pang and Gabriel, we define $z_i(x,d) := (b_i(x,d))^2/2$. Note that $h(x,d) = H(x) + b(x,d)$.

The NE/SQP algorithm for mixed complementarity problems is stated in Figure 1.

Figure 1: Algorithm NE/SQP

Step 1 [Initialization] Select $\rho, \sigma \in (0,1)$, and a starting vector $x^0 \in \mathrm{B}$. Set $k = 0$.

Step 2 [Direction generation] Solve (QP_k), giving the direction d^k.
If $\phi(x^k, d^k) = \theta(x^k)$, terminate the algorithm; otherwise, continue.

Step 3 [Steplength determination] Let m_k be the smallest nonnegative integer m such that

$$\theta(x^k + \rho^m d^k) - \theta(x^k) \leq -\sigma \rho^m z(x^k, d^k); \tag{7}$$

set $x^{k+1} = x^k + \rho^{m_k} d^k$.

Step 4 [Termination check] If x^{k+1} satisfies a prescribed stopping rule, stop.
Otherwise, return to Step 2, with k replaced by $k+1$.

To present the convergence results for this algorithm, it is convenient to further partition the index sets as follows:

$$
\begin{aligned}
I_{el}^{+}(x) &:= \{i \in I_{el} : x_i - l_i > 0\} \\
I_{el}^{0}(x) &:= \{i \in I_{el} : x_i - l_i = 0\} \\
I_{f}^{l}(x) &:= \{i \in I_{f} : x_i - l_i = 0\} \\
I_{f}^{n}(x) &:= \{i \in I_{f} : x_i - u_i < 0 < x_i - l_i\} \\
I_{f}^{u}(x) &:= \{i \in I_{f} : x_i - u_i = 0\} \\
I_{eu}^{0}(x) &= \{i \in I_{eu} : x_i - u_i = 0\} \\
I_{eu}^{-}(x) &:= \{i \in I_{eu} : x_i - u_i < 0\}.
\end{aligned}
$$

Note that for $x \in \mathbf{B}$, the sets $I_l(x)$, $I_{el}^{+}(x)$, $I_{el}^{0}(x)$, $I_f^l(x)$, $I_f^n(x)$, $I_f^u(x)$, I_{eu}^0, $I_{eu}^-(x)$, and $I_u(x)$ form a partition of the indices $\{1, \ldots, n\}$.

The convergence results of this algorithm are based on two regularity conditions: *b-regularity* and *s-regularity*.

Definition 2.2.1 *A nonnegative vector x is said to be b-regular if for every index set α satisfying*

$$I_f^n(x) \subset \alpha \subset I_f(x) \bigcup I_{el}(x) \bigcup I_{eu}(x),$$

the principal submatrix

$$\nabla_\alpha f_\alpha(x)$$

is nonsingular.

The term "b-regular" is chosen because the condition is used to prove boundedness of the search directions computed by the NE/SQP algorithm.

Definition 2.2.2 *A nonnegative vector x is said to be s-regular if the following* *linear inequality system has a solution in* y*:*

$$
\begin{array}{rl}
x_i - l_i + y_i = 0 & i \in I_l(x) \\
x_i - u_i + y_i = 0 & i \in I_u(x) \\
f_i(x) + \nabla f_i(x)^\top y = 0 & i \in I_f^n(x) \\
x_i - l_i + y_i \geq 0 & i \in I_f^l(x) \\
f_i(x) + \nabla f_i(x)^\top y \geq 0 & i \in I_f^l(x) \\
x_i - u_i + y_i \leq 0 & i \in I_f^u(x) \\
f_i(x) + \nabla f_i(x)^\top y \leq 0 & i \in I_f^u(x) \\
x_i - l_i + y_i \leq 0 & i \in I_{el}^+(x) \\
f_i(x) + \nabla f_i(x)^\top y \leq 0 & i \in I_{el}^+(x) \\
x_i - u_i + y_i \geq 0 & i \in I_{eu}^-(x) \\
f_i(x) + \nabla f_i(x)^\top y \geq 0 & i \in I_{eu}^-(x) \\
y_i = 0 & i \in I_{el}^0(x) \bigcup I_{eu}^0(x).
\end{array}
\tag{8}
$$

Note that when $l = 0, u = \infty$ the above definitions are identical to the concepts of b-regularity and s-regularity defined by Pang and Gabriel. The term "s-regular" is chosen to emphasize the relationship given by Pang & Gabriel (1993, Proposition 3), which relates s-regularity to the notion of an S-matrix (Cottle et al. 1992, Definition 3.1.4).

The analysis that follows is nearly identical to the analysis in Pang & Gabriel (1993) with the exception that we have to handle a more extensive collection of index sets. Whenever possible, we refer to the original paper. However, occasionally we are forced to recreate the proofs to be assured that the results hold for the MCP.

Lemma 2.2.3 *Suppose that* $\{x^k\} \subset B$ *converges to* $\bar{x}$*. Then for all* k *sufficiently large, the following relations hold:*

1. $I_f^n(\bar{x}) \subset I_f^n(x^k)$,
2. $I_l(\bar{x}) \subset I_l(x^k)$,
3. $I_u(\bar{x}) \subset I_u(x^k)$,
4. $I_f^l(\bar{x}) \subset I_f^n(x^k) \bigcup I_f^l(x^k)$,
5. $I_f^u(\bar{x}) \subset I_f^n(x^k) \bigcup I_f^u(x^k)$,
6. $I_{el}^+(\bar{x}) \subset I_f^n(x^k) \bigcup I_l(x^k) \bigcup I_{el}^+(x^k)$,

7. $I_{eu}^{-}(\bar{x}) \subset I_f^n(x^k) \bigcup I_u(x^k) \bigcup I_{eu}^{-}(x^k)$,

8. $I_{el}^0(x^k) \subset I_{el}^0(\bar{x})$,

9. $I_{eu}^0(x^k) \subset I_{eu}^0(\bar{x})$,

Proof Parts 1–7 are obvious consequences of the continuity of the function f. To prove part 8, let $T_l(x^k) := I_f(x^k) \bigcup I_{el}(x^k) \bigcup I_l(x^k)$, and observe that $I_{el}^0(x^k)$ is the complement in $T_l(x^k)$ of $I_f^n(x^k) \bigcup I_l(x^k) \bigcup I_f^l(x^k) \bigcup I_{el}^{+}(x^k)$. Part 8 then follows easily from parts 1,2,4, and 6. Similarly, part 9 follows from parts 1,3,5, and 7. $\square$

Proposition 2.2.4 *Suppose that x is a stationary point of the problem*

$$\begin{array}{ll} \min & \theta(x) \\ \textit{subject to} & x \in \boldsymbol{B}. \end{array}$$

Then x solves MCP(f, B) if and only if x is s-regular.

Proof If x is a solution, then $I_f^l(x)$, $I_f^u(x)$, $I_{el}^{+}(x)$, and $I_{eu}^{-}(x)$ are all empty. It is then easily seen that $y = 0$ satisfies the s-regularity conditions (8). This establishes the necessity.

To prove the sufficiency, let y be a solution of (8). It is easily seen that for $\epsilon > 0$ small enough, $x + \epsilon y \in$ B. Thus, since x is a stationary point of θ over B,

$$0 \le \theta'(x; \epsilon y) \quad \text{which implies} \quad 0 \le \theta'(x; y). \tag{9}$$

But $\theta'(x; y) = \sum_{i=1}^n H_i(x) H_i'(x; y)$. By examining each term of this sum, we can show that $\theta'(x; y) \le -2\theta(x)$. For example if $i \in I_l(x)$, we see that $H_i(x) = x_i - l_i$, while by (8), $H_i'(x; y) = y_i = -(x_i - l_i)$. Thus, $H_i(x) H_i'(x; y) = -(x_i - l_i)^2 = -H_i(x)^2 = -2\phi_i(x, 0)$. In similar fashion, we can easily show that $H_i(x) H_i'(x; y) \le -2\phi_i(x, 0)$ for all i. Thus, $\theta'(x; y) \le -2\theta(x) \le 0$. But by (9), $0 \le \theta'(x; y)$, so $\theta(x) = 0$. $\square$

Lemma 2.2.5 *The following properties hold:*

1. *$\phi(x, 0) = \theta(x)$ for all $x \in \boldsymbol{B}$.*

2. *$\phi(x, d) - \phi(x, 0) - z(x, d) \ge \theta'(x; d)$ for all $(x, d) \in \boldsymbol{B} \times \boldsymbol{R}^n$, and*

3. *for any sequence $\{(x^k, d^k)\} \subset \boldsymbol{B} \times \boldsymbol{R}^n$ converging to $(\bar{x}, 0)$ for some $\bar{x} \in \boldsymbol{B}$, $\lim_{k \to \infty} \phi(x^k, d^k) = \phi(\bar{x}, 0)$.*

Proof The proof of the first two parts comes simply by writing out the definition for all the quantities involved as the sum of the componentwise terms, and then comparing each of the summands individually. The proof of the last part requires the use of Lemma 2.2.3. Let the sequence $\{(x^k, d^k)\}$ be as given. In similar fashion to the proof of Pang & Gabriel (1993, Lemma 2), we see that for $i \in I_f(\bar{x})$, then for all k sufficiently large, by Lemma 2.2.3, $i \in I_f(x^k)$, so

$$\phi_i(\bar{x}, 0) = \frac{1}{2} f_i(\bar{x})^2 = \lim_{k\to\infty} \frac{1}{2}(f_i(x^k) + \nabla f_i(x^k)^\top d^k)^2 = \lim_{k\to\infty} \phi_i(x^k, d^k).$$

Similarly, for $i \in I_l(\bar{x})$,

$$\phi_i(\bar{x}, 0) = \frac{1}{2}(\bar{x}_i - l_i)^2 = \lim_{k\to\infty} \frac{1}{2}(x_i^k - l_i + d_i^k)^2 = \lim_{k\to\infty} \phi_i(x^k, d^k),$$

and for $i \in I_u(\bar{x})$,

$$\phi_i(\bar{x}, 0) = \lim_{k\to\infty} \phi_i(x^k, d^k),$$

For $i \in I_{el}(\bar{x})$, $\lim_{k\to\infty}(x_i^k - l_i + d^k) = \bar{x}_i - l_i = f_i(\bar{x}) = \lim_{k\to\infty}(f_i(x^k) + \nabla f_i(x^k)^\top d^k)$. Thus, by a simple combinatorial argument, $\lim_{k\to\infty} \phi_i(x^k, d^k) = \phi_i(\bar{x}, 0)$. A similar argument gives the result for $i \in I_{eu}(\bar{x})$. □

Proposition 2.2.6 *Let $x \in \boldsymbol{B}$ be arbitrary. Then the problem (QP_x) has at least one optimal solution. Let $\tilde{d}_x$ denote an arbitrary optimal solution. The following statements hold:*

1. *$z(x, \tilde{d}_x) \le \theta(x)$,*
2. *$\phi(x, \tilde{d}_x) \le \phi(x, 0)$, with equality holding if and only if $z(x, \tilde{d}_x) = 0$,*
3. *if $\phi(x, \tilde{d}_x) < \phi(x, 0)$, then for any $\sigma \in (0, 1)$, there exists a scalar $\bar{\tau} > 0$ such that for all $\tau \in [0, \bar{\tau}]$,*

$$\theta(x + \tau\tilde{d}_x) - \theta(x) \le -\sigma\tau z(x, \tilde{d}_x),$$

4. *if $\phi(x, \tilde{d}_x) = \phi(x, 0)$ and if x is s-regular, then x solves the MCP(f, B).*

Proof For simplicity, we drop the subscript x from the optimal solution $\tilde{d}_x$. In similar fashion to the proof of Pang & Gabriel (1993, Proposition 2), we can establish the inequality

$$(10) \qquad \sum_{i\in J_l(x)} \tilde{d}_i^2 + \sum_{i\in J_u(x)} \tilde{d}_i^2 + \sum_{i\in I_f(x)} (\nabla f_i(x)^\top \tilde{d})^2$$

$$
\begin{aligned}
&\leq -\sum_{i\in J_l(x)}(x_i-l_i)\tilde{d}_i-\sum_{i\in J_u(x)}(x_i-u_i)\tilde{d}_i-\sum_{i\in I_f(x)}f_i(x)(\nabla f_i(x)^\top\tilde{d})\\
&\leq \frac{1}{2}\sum_{i\in J_l(x)}((x_i-l_i)^2+\tilde{d}_i^{\,2})+\frac{1}{2}\sum_{i\in J_u(x)}((x_i-u_i)^2+\tilde{d}_i^{\,2})\\
&\quad+\frac{1}{2}\sum_{i\in I_f(x)}((f_i(x)^2+(\nabla f_i(x)^\top\tilde{d})^2).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
2z(x,\tilde{d}) &= \sum_{i=1}^{n} b_i(x,\tilde{d})^2\\
&\leq -\sum_{i=1}^{n} H_i(x)b_i(x,\tilde{d})\\
&\leq \frac{1}{2}\sum_{i=1}^{n}\left(H_i(x)^2+b_i(x,\tilde{d})^2\right)\\
&= \theta(x)+z(x,\tilde{d}).
\end{aligned}
\tag{11}
$$

Part 1 easily follows from this inequality. For part 2, we need only establish the necessary and sufficient conditions for equality to hold. Suppose $\phi(x,\tilde{d})=\phi(x,0)$. Then we have

$$0=\phi(x,\tilde{d})-\phi(x,0)=H(x)^\top b(x,\tilde{d})+z(x,\tilde{d}),$$

which, by the second line of (11), implies that $z(x,\tilde{d})=0$. Conversely, if $z(x,\tilde{d})=0$, then $b(x,\tilde{d})=0$, which clearly yields $\phi(x,\tilde{d})=\phi(x,0)$.

The proof to part 3 is identical to the proof of Pang & Gabriel (1993, Proposition 2, part(c)). The proof to part 4 is also identical except that we have to examine a different collection of index sets to establish the following inequality: let y be any vector satisfying the s-regularity condition (8), then for $\epsilon>0$ sufficiently small,

$$\phi_i(x,\epsilon y)\leq(1-\epsilon)^2\phi_i(x,0),\quad i=1,\ldots,n. \tag{12}$$

For $i\in I_l(x)\bigcup I_f^n(x)\bigcup I_{el}^0(x)\bigcup I_u(x)\bigcup I_{eu}^0(x)$, this inequality is satisfied as an equality. For $i\in I_f^l(x)\bigcup I_{eu}^-(x)$, we have $f_i(x)<0$, so for small ϵ,

$$0>f_i(x)+\epsilon\nabla f_i(x)^\top y\geq(1-\epsilon)f_i(x),\quad \text{by (8)}.$$

(12) follows by squaring this inequality. The argument for $i\in I_f^u(x)\bigcup I_{el}^+(x)$ is similar. The remainder of the proof is then identical to the proof of Pang & Gabriel (1993, Proposition 2, part(d)). $\square$

The third part of this proposition guarantees that the integer m_k in Step 3 of the algorithm can always be determined in a finite number of trials. Moreover, since the direction finding subproblems are always solvable, the algorithm will generate a well-defined sequence of points $\{x^k\}$, along with a sequence of optimal solutions $\{d^k\}$ to the direction finding subproblems (QP_k). If we remove the stopping rule in Step 4 of the algorithm, the sequence generated by the algorithm will be infinite unless the algorithm terminates at a point $\bar{x}$ in step 2. In that case, by part 4 of Proposition 2.2.6, $\bar{x}$ solves $\mathrm{MCP}(f, \mathrm{B})$ if $\bar{x}$ is s-regular. For the remainder of our analysis, we assume that the algorithm does not terminate in Step 2, so that we may assume that the algorithm generates an infinite sequence of iterates $\{x^k\}$.

Lemma 2.2.7 *Suppose that x^* is the limit of the subsequence $\{x^k : k \in \kappa\}$. If x^* is b-regular, then $\{d^k : k \in \kappa\}$ is bounded.*

Proof The proof is identical to the proof of Pang & Gabriel (1993, Lemma 3), with the following substitutions:

$$\begin{array}{lcl} I_f^+(x^*) & \Rightarrow & I_f^n(x^*) \\ I_f(x^*) \bigcup I_e(x^*) & \Rightarrow & I_f(x^*) \bigcup I_{el}(x^*) \bigcup I_{eu}(x^*) \\ J_f(x^k) & \Rightarrow & J_l(x^k) \bigcup J_u(x^k). \end{array}$$

$\square$

Lemma 2.2.8 *Suppose that x^* is the limit of the subsequence $\{x^k : k \in \kappa\}$ and that the sequence $\{z(x^k, d^k) : k \in \kappa\}$ converges. Suppose also that x^* is b-regular. Then, for every sequence of positive scalars $\{\lambda_k : k \in \kappa\}$ converging to zero,*

$$\limsup_{k \to \infty, k \in \kappa} \frac{\theta(x^k + \lambda_k d^k) - \theta(x^k)}{\lambda_k} \leq - \lim_{k \to \infty, k \in \kappa} z(x^k, d^k).$$

Proof By Lemma 2.2.7, the sequence $\{d^k : k \in \kappa\}$ is bounded. Thus, the sequence $\{x^k + \lambda_k d^k : k \in \kappa\}$ also converges to x^*. Let $y^k := x^k + \lambda_k d^k$. In similar fashion to the proof of Pang & Gabriel (1993, Lemma 4), we can show that for k sufficiently large, the following holds: for $i \in I_f(x^*) \bigcup I_l(x^*) \bigcup I_u(x^*)$,

$$\theta_i(y^k) - \theta_i(x^k) = \begin{cases} \lambda_k f_i(x^k) \nabla f_i(x^k)^\top d^k + o(\lambda_k) & i \in I_f(x^k) \\ \lambda_k (x_i^k - l_i) d_i^k + o(\lambda_k) & i \in I_l(x^k) \\ \lambda_k (x_i^k - u_i) d_i^k + o(\lambda_k) & i \in I_u(x^k); \end{cases}$$

and for $i \in I_{el}^{+}(x^*) \bigcup I_{eu}^{-}(x^*)$,

$$\theta_i(y^k) - \theta_i(x^k) \leq \begin{cases} \lambda_k f_i(x^k) \nabla f_i(x^k)^\top d^k + o(\lambda_k) & i \in I_f^n(x^k) \\ \lambda_k (x_i^k - l_i) d_i^k + o(\lambda_k) & i \in I_l(x^k) \bigcup I_{el}^{+}(x^k) \\ \lambda_k (x_i^k - u_i) d_i^k + o(\lambda_k) & i \in I_u(x^k) \bigcup I_{eu}^{-}(x^k). \end{cases}$$

Summing these expression, we get

$$\begin{aligned} (13) \qquad & \sum_{i \notin I_{el}^0(x^*) \bigcup I_{eu}^0(x^*)} \frac{\theta(x^k + \lambda_k d^k) - \theta(x^k)}{\lambda_k} \\ & \leq \sum_{i \in T_1} f_i(x^k) \nabla f_i(x^k)^\top d^k + \sum_{i \in T_2} (x_i^k - l_i) d_i^k + \sum_{i \in T_3} (x_i^k - u_i) d_i^k + o(\lambda_k), \end{aligned}$$

where

$$\begin{aligned} T_1 &:= \left(I_f(x^*) \bigcap I_f(x^k)\right) \bigcup \left((I_{el}^{+}(x^*) \bigcup I_{eu}^{-}(x^*)) \bigcap I_f^n(x^k)\right) \\ T_2 &:= \left(I_l(x^*) \bigcap I_l(x^k)\right) \bigcup \left((I_{el}^{+}(x^*) \bigcup I_{eu}^{-}(x^*)) \bigcap \left(I_l(x^k) \bigcup I_{el}^{+}(x^k)\right)\right) \\ T_3 &:= \left(I_u(x^*) \bigcap I_u(x^k)\right) \bigcup \left((I_{el}^{+}(x^*) \bigcup I_{eu}^{-}(x^*)) \bigcap \left(I_u(x^k) \bigcup I_{eu}^{-}(x^k)\right)\right). \end{aligned}$$

Note that $T_1 \bigcup T_2 \bigcup T_3$ is the complement of $I_{el}^0(x^*) \bigcup I_{eu}^0(x^*)$.

It is also easily shown that for $i \in I_{el}^0(x^*) \bigcup I_{eu}^0(x^*)$,

$$\lim_{k \to \infty, k \in \kappa} \frac{\theta_i(y^k) - \theta_i(x^k)}{\lambda_k} = 0.$$

Notice that $I_f(x^k) \backslash T_1 \subset I_{el}^0(x^*) \bigcup I_{eu}^0(x^*)$, $J_l(x^k) \backslash T_2 \subset I_{el}^0(x^*)$, and, $J_u(x^k) \backslash T_3 \subset I_{eu}^0(x^*)$. Thus,

$$\begin{aligned} \limsup_{k \in \kappa, k \to \infty} \sum_{i \in T_1} f_i(x^k) \nabla f_i(x^k)^\top d^k &= \limsup_{k \in \kappa, k \to \infty} \sum_{i \in I_f(x^k)} f_i(x^k) \nabla f_i(x^k)^\top d^k, \\ \limsup_{k \in \kappa, k \to \infty} \sum_{i \in T_2} (x_i^k - l_i) d_i^k &= \limsup_{k \in \kappa, k \to \infty} \sum_{i \in J_l(x^k)} (x_i^k - l_i) d_i^k, \quad \text{and} \\ \limsup_{k \in \kappa, k \to \infty} \sum_{i \in T_3} (x_i^k - u_i) d_i^k &= \limsup_{k \in \kappa, k \to \infty} \sum_{i \in J_u(x^k)} (x_i^k - u_i) d_i^k. \end{aligned}$$

Thus, by (13)

$$\limsup_{k \in \kappa, k \to \infty} \frac{\theta(x^k + \lambda_k d^k) - \theta(x^k)}{\lambda_k} \leq \limsup_{k \in \kappa, k \to \infty} \left(\sum_{i \in I_f(x^k)} f_i(x^k) \nabla f_i(x^k)^\top d^k \right.$$

$$
\begin{aligned}
& + \sum_{i \in J_l(x^k)} (x_i^k - l_i) d_i^k + \sum_{i \in J_u(x^k)} (x_i^k - u_i) d_i^k \Bigg) \\
\leq & \; -2 \lim_{k \in \kappa, k \to \infty} z(x^k, d^k) \quad \text{by (11)} \\
\leq & \; - \lim_{k \in \kappa, k \to \infty} z(x^k, d^k),
\end{aligned}
$$

since $z(x^k, d^k)$ is nonnegative. □

Lemma 2.2.9 *Suppose that x^* is the limit of the subsequence $\{x^k : k \in \kappa\}$. If x^* is b-regular, then*

$$\lim_{k \in \kappa, k \to \infty} z(x^k, d^k) = 0;$$

hence, $\{d^k : k \in \kappa\}$ converges to zero.

Proof The proof is identical to the proof of Pang & Gabriel (1993, Lemma 5) except that Lemma 2.2.7 is used in place of Pang & Gabriel (1993, Lemma 3). □

Lemma 2.2.10 *Suppose that x^* is the limit of the subsequence $\{x^k : k \in \kappa\}$ and that x^* is s-regular. Let y be a solution of the system (8) with $x = x^*$. Then there exists an $\bar{\epsilon} \in (0, 1]$ such that for all $k \in \kappa$ large enough, ϵy is feasible for the problem (QP_k) for any $\epsilon \in [0, \bar{\epsilon}]$.*

Proof We look at each component i to show that $l_i \leq x_i^k + \epsilon y \leq u_i$ for all $k \in \kappa$ sufficiently large, and for all ϵ sufficiently small. This is certainly true if $l_i < x_i^* < u_i$, since for k large enough, x_i^k will be bounded away from l_i and u_i. If $x_i^* = l_i$, then for k large enough, x_i^k is bounded away from u_i. Thus, we need only show that $x_i^k + \epsilon y_i^k \geq l_i$. But $x_i^* = l_i \Rightarrow i \in I_f^l(x^*) \bigcup I_l(x^*) \bigcup I_{el}^0(x^*)$. Thus by the s-regularity system (8), $y_i \geq 0$, so $x_i^k + \epsilon y_i^k \geq l_i$ for any $\epsilon \geq 0$. The proof is completed by a similar argument showing that if $x_i^* = u_i$, then $y_i \leq 0$. □

Lemma 2.2.11 *Under the assumptions of Lemma 2.2.10, it holds that for all $\epsilon \geq 0$ sufficiently small,*

$$\limsup_{k \in \kappa, k \to \infty} \phi(x^k, \epsilon y) \leq (1 - \epsilon)^2 \phi(x^*, 0) + \mathcal{O}(\epsilon^2). \tag{14}$$

Proof Suppose $k \in \kappa$ is large enough that all of the inclusions of Lemma 2.2.3 are satisfied. If $i \in J_l(x^k)$ then by Lemma 2.2.3, $i \in I_l(x^*) \bigcup I_{el}^0(x^*) \bigcup I_{el}^+(x^*)$. If $i \in I_l(x^*) \bigcup I_{el}^0(x^*)$, then by (8), $y_i = l_i - x_i^*$, so

$$
\begin{aligned}
\phi(x^k, \epsilon y) &= \frac{1}{2}(x_i^k - l_i - \epsilon(x_i^* - l_i))^2 \\
&= \frac{1}{2}\left((1 - \epsilon)^2 (x_i^* - l_i) + x_i^k - x_i^*\right)^2 \\
&= (1 - \epsilon)^2 \phi(x^*, 0) + \mathcal{O}\left(x_i^k - x_i^*\right).
\end{aligned}
$$

If $i \in I_{el}^{+}(x^*)$, then for k large enough, x_i^k is bounded away from l_i, so for ϵ small enough,

$$0 \leq x_i^k - l_i + \epsilon y_i \leq x_i^k - l_i - \epsilon(x_i^* - l_i).$$

This implies that

$$\phi_i(x^k, \epsilon y) \leq \frac{1}{2}(x_i^k - l_i - \epsilon(x_i^* - l_i))^2 = (1-\epsilon)^2 \phi(x^*, 0) + \mathcal{O}\left(x_i^k - x_i^*\right).$$

For $i \in I_u(x^*) \bigcup I_{eu}^0(x^*) \bigcup I_{eu}^-(x^*)$, a similar argument gives

$$\phi_i(x^k, \epsilon y) \leq \frac{1}{2}(x_i^k - u_i - \epsilon(x_i^* - u_i))^2 = (1-\epsilon)^2 \phi(x^*, 0) + \mathcal{O}\left(x_i^k - x_i^*\right).$$

Summing over all $i \in J_l(x^k) \bigcup J_u(x^k)$ gives

$$\sum_{i \notin I_f(x^k)} \phi_i(x^k, \epsilon y) \leq (1-\epsilon)^2 \sum_{i \notin I_f(x^k)} \phi_i(x^*, 0) + \mathcal{O}\left(\left\|x^k - x^*\right\|\right). \tag{15}$$

Consider now an index $i \in I_f(x^k)$. By Lemma 2.2.3,

$$i \in I_f^n(x^*) \bigcup I_f^l(x^*) \bigcup I_f^u(x^*) \bigcup I_{el}^+(x^*) \bigcup I_{el}^0(x^*) \bigcup I_{eu}^-(x^*) \bigcup I_{eu}^0(x^*).$$

If $i \in I_f^n(x^*)$, then by (8), $\nabla f_i(x^*)^\top y = -f_i(x^*)$, so

$$\begin{aligned} f_i(x^k) + \epsilon \nabla f_i(x^k)^\top y &= f_i(x^k) + \epsilon \nabla f_i(x^*)^\top y + \epsilon(\nabla f_i(x^k)^\top y - \nabla f_i(x^*)^\top y) \\ &= f_i(x^k) - \epsilon f_i(x^*) + \epsilon(\nabla f_i(x^k)^\top y - \nabla f_i(x^*)^\top y). \end{aligned}$$

Thus,

$$\begin{aligned} \phi_i(x^k, \epsilon y) &= \tfrac{1}{2}\left(f_i(x^k) - \epsilon f_i(x^*) + \epsilon(\nabla f_i(x^k) - \nabla f_i(x^*))^\top y\right)^2 \\ &= (1-\epsilon)^2 \phi_i(x^*, 0) + \mathcal{O}\left(f_i(x^k) - f_i(x^*)\right) \\ &\quad + \mathcal{O}\left(\left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|\right). \end{aligned}$$

If $i \in I_f^l(x^*)$, then $f_i(x^k)$ is negative and bounded away from 0 for k large enough. Thus, for $\epsilon \geq 0$ small enough,

$$0 \geq f_i(x^k) + \epsilon \nabla f_i(x^k)^\top y \geq f_i(x^k) - \epsilon f_i(x^*) + \epsilon(\nabla f_i(x^k) - \nabla f_i(x^*))^\top y,$$

which implies

$$\begin{aligned} \phi_i(x^k, \epsilon y) &\leq \tfrac{1}{2}\left(f_i(x^k) - \epsilon f_i(x^*) + \epsilon(\nabla f_i(x^k) - \nabla f_i(x^*))^\top y\right)^2 \\ &= (1-\epsilon)^2 \phi_i(x^*, 0) + \mathcal{O}\left(f_i(x^k) - f_i(x^*)\right) \\ &\quad + \mathcal{O}\left(\left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|\right). \end{aligned}$$

Similar arguments duplicate this inequality for $i \in I_f^u(x^*)$, $i \in I_{el}^+(x^*)$ and $i \in I_{eu}^-(x^*)$. Finally, for $i \in I_{el}^0(x^*) \bigcup I_{eu}^0(x^*)$, we have $f_i(x^*) = 0$, so

$$\begin{aligned}\phi_i(x^k, \epsilon y) &= \frac{1}{2}\left(f_i(x^k) - \epsilon f_i(x^*) + \epsilon \nabla f_i(x^k)^\top y\right)^2 \\ &= (1-\epsilon)^2 \phi_i(x^*, 0) + \mathcal{O}\left(f_i(x^k) - f_i(x^*)\right) \\ &\quad + \mathcal{O}\left(\left\|\nabla f_i(x^k) - \nabla f_i(x^*)\right\|\right) + \mathcal{O}(\epsilon^2),\end{aligned}$$

where $\mathcal{O}(\epsilon^2)$ comes from squaring the term $\epsilon \nabla f_i(x^k)^\top y$. Summing up the above inequalities, for all $i \in I_f(x^k)$ we get

$$\begin{aligned}&\sum_{i \in I_f(x^k)} \phi_i(x^k, \epsilon y) \\ &\quad \leq (1-\epsilon)^2 \sum_{i \in I_f(x^k)} \phi_i(x^*, 0) + \mathcal{O}\left(\left\|f(x^k) - f(x^*)\right\|\right) \\ &\qquad + \mathcal{O}\left(\left\|\nabla f(x^k) - \nabla f(x^*)\right\|\right) + \mathcal{O}(\epsilon^2).\end{aligned} \tag{16}$$

Adding (15) and (16) and passing to the limit $k \to \infty$, we obtain inequality (14). □

Theorem 2.2.12 *Let $f : \Omega \supset \boldsymbol{B} \to \boldsymbol{R}^n$ be a once continuously differentiable function. Let $x^0 \in \boldsymbol{B}$ be arbitrary. The following two statements hold:*

1. *NE/SQP generates a well defined sequence of iterates $\{x^k\} \subset \boldsymbol{B}$ along with a sequence of optimal solutions $\{d^k\}$ for the subproblems (QP_k);*

2. *if x^* is the limit of an infinite subsequence $\{x^k : k \in \kappa\}$, and if x^* is both b-regular and s-regular, then x^* is a solution of MCP($f, \boldsymbol{B}$).*

Proof The proof is identical to the proof of Pang & Gabriel (1993, Theorem 1), except that Lemmas 2.2.5, 2.2.9, 2.2.10, and 2.2.11 are used in place of Pang & Gabriel (1993, Lemmas 2,5,6, and 7). □

Lemma 2.2.13 *Let $\bar{x}$ be an arbitrary solution of MCP($f, \boldsymbol{B}$). Suppose that $\bar{x}$ is b-regular. Then for every $\epsilon > 0$, there exists a $\delta > 0$ such that whenever $\|z - \bar{x}\| \leq \delta$, and $z \in \boldsymbol{B}$,*

$$\left\|z + \tilde{d}_z - \bar{x}\right\| \leq \epsilon \left\|z - \bar{x}\right\|, \tag{17}$$

where $\tilde{d}_z$ is any optimal solution of the quadratic program (QP_z).

Proof For convenience, we drop the subscript z from $\tilde{d}_z$.

Since $z \in \mathrm{B}$, it follows that the vector $d := \bar{x} - z$ is feasible for the quadratic program (QP_z). By letting $v := d - \tilde{d}$, we obtain from the minimum principle for this program that

$$(18)\qquad \begin{array}{rcl} 0 & \leq & \nabla \phi_x(\tilde{d})^\top v \\ & = & \sum_{i \in J_l(z)} (z_i - l_i + \tilde{d}_i) v_i + \sum_{i \in J_u(z)} (z_i - u_i + \tilde{d}_i) v_i \\ & & + \sum_{i \in I_f(z)} \left(f_i(z) + \nabla f_i(z)^\top \tilde{d} \right) (\nabla f_i(z)^\top v). \end{array}$$

Furthermore, since z is close to $\bar{x}$, we have $i \in J_l(z) \Rightarrow \bar{x}_i = l_i$, $i \in J_u(z) \Rightarrow \bar{x}_i = u_i$, and $i \in I_f(z) \Rightarrow f_i(\bar{x}) = 0$. Hence, by subtracting these quantities from (18) and rearranging terms, we get

$$(19)\qquad \begin{array}{l} \displaystyle\sum_{i \in J_l(z) \bigcup J_u(z)} v_i^2 + \sum_{i \in I_f(z)} (\nabla f_i(z)^\top v)^2 \\ \displaystyle\leq \; - \sum_{i \in I_f(z)} \left(f_i(\bar{x}) - f_i(z) - \nabla f_i(z)^\top (\bar{x} - z) \right) (\nabla f_i(z)^\top v). \end{array}$$

Let

$$A_z := \begin{bmatrix} I_{zz} & 0 \\ \nabla_{J_z} f_{I_z}(z)^\top & \nabla_{I_z} f_{I_z}(z)^\top \end{bmatrix}$$

where $J_z := J_l(z) \bigcup J_u(z)$, $I_z = I_f(z)$, and I_{zz} denotes the identity matrix of order $|J_z|$. Using this matrix, the left-hand side of (19) becomes $\|A_z v\|^2$. Applying the Cauchy-Schwarz inequality to the right-hand side of this same inequality produces

$$\|A_z v\|^2 \leq \left\| f(\bar{x}) - f(z) - \nabla f(z)^\top (\bar{x} - z) \right\| \|A_z v\|.$$

The remainder of the proof is identical to the proof of Pang & Gabriel (1993, Lemma 8). □

Lemma 2.2.14 *In the setting of Lemma 2.2.13, there exists a constant $c > 0$ such that for all vectors z close enough to $\bar{x}$,*

$$\left\| \tilde{d}_z \right\| \leq c \|H(z)\|.$$

Moreover, if $L > 0$ is the Lipschitzian modulus of H at $\bar{x}$, then, for $\epsilon \in (0, 1)$,

$$\theta(z + \tilde{d}_z) \leq \left(\frac{\epsilon c L}{1 - \epsilon} \right)^2 \theta(z).$$

Proof The proof is identical to the proof of Pang & Gabriel (1993, Lemma 9). □

Theorem 2.2.15 *Let $f : \Omega \supset \boldsymbol{B} \to \boldsymbol{R}^n$ be a once continuously differentiable function, and $x^0 \in \boldsymbol{B}$ be arbitrary. Suppose that x^* is an accumulation point of an infinite sequence of iterates $\{x^k\}$ generated by the NE/SQP method, and x^* is both b-regular and s-regular. Then x^* solves MCP(f, B). Moreover, the following statements hold:*

1. *there exists an integer $K > 0$ such that for all $k \geq K$, the stepsize $\tau_k = \rho^{m_k} = 1$, hence, $x^{k+1} = x^k + d^k$;*
2. *the sequence $\{x^k\}$ converges to x^* Q-superlinearly.*
3. *if ∇f is Lipschitzian in a neighborhood of x^*, then the sequence $\{x^k\}$ converges to x^* Q-quadratically.*

Proof The proof is identical to the proof of Pang & Gabriel (1993, Theorem 2) except that Lemma 2.2.13 is used in place of Pang & Gabriel (1993, Lemma 8), and Proposition 2.2.6 is used in place of Pang & Gabriel (1993, Proposition 2). $\square$

The above convergence results establish the fact that the NE/SQP algorithm has very good local convergence behavior. But the global convergence results are not very useful from a practical standpoint. The problem is that the s-regularity and b-regularity conditions are dependent not only on the problem, but also on the algorithm. In particular, they depend on the particular choice of merit function used.

A global convergence result can be established by showing that these regularity assumptions are satisfied whenever f is strongly monotone. This result is more useful for our purposes and is proved in Theorem 2.2.18; the next two lemmas are instrumental in proving this theorem.

Lemma 2.2.16 *Let $A \in \boldsymbol{R}^{m\times n}, B \in \boldsymbol{R}^{l\times n}$, where $l + m = n$. If the matrix*

$$M := \begin{bmatrix} A \\ B \end{bmatrix}$$

is positive definite, then for any vectors $p \in \boldsymbol{R}^m$, $q \in \boldsymbol{R}^l$, and $r \in \boldsymbol{R}^l$, the system

$$A \begin{bmatrix} x \\ y \end{bmatrix} = p, \qquad B \begin{bmatrix} x \\ y \end{bmatrix} \geq q, \qquad y \geq r \tag{20}$$

has a solution $(x, y) \in \boldsymbol{R}^m \times \boldsymbol{R}^l$.

Proof We first establish that the system

$$A\begin{bmatrix}\xi\\ \eta\end{bmatrix}=0,\qquad B\begin{bmatrix}\xi\\ \eta\end{bmatrix}>0,\qquad \eta>0 \tag{21}$$

has a solution $(\xi,\eta)\in \mathrm{R}^m\times\mathrm{R}^l$. To do this, assume that no solution exists. Then by Motzkin's theorem of the alternative (Mangasarian 1969, Theorem 2.4.2), there are vectors $u\in\mathrm{R}^m$, $v\in\mathrm{R}^l_+$, and $w\in\mathrm{R}^l_+$, with $(v,w)\neq 0$ such that

$$\begin{bmatrix}A^\top B^\top\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix}=\begin{bmatrix}0\\ -w\end{bmatrix}. \tag{22}$$

Premultiplying both sides of this equation by $(u^\top,v^\top)$, we see that

$$(u^\top,v^\top)M^\top\begin{bmatrix}u\\ v\end{bmatrix}=-v^\top w\le 0,\quad (\text{since } v\ge 0 \text{ and } w\ge 0).$$

Since M is positive definite, this can happen only if $(u,v)=0$. But then (22) implies that $w=0$, which contradicts the fact that $(v,w)\neq 0$. Thus, by contradiction, (21) has a solution (ξ,η).

We return now to (20). Since M is positive definite, A has full row rank, so there exists a vector $(\tilde{x},\tilde{y})$ such that $A(\tilde{x},\tilde{y})=p$. Then, for λ sufficiently large, the point $(x,y):=(\tilde{x},\tilde{y})+\lambda(\xi,\eta)$ will satisfy (20). □

Lemma 2.2.17 *If f is strongly monotone, then all points $x\in B$ are both b-regular and s-regular.*

Proof Since f is strongly monotone, then by Ortega & Rheinboldt (1970, 5.4.3), $\nabla f(x)$ is positive definite at every point x. Thus, $\nabla_\alpha f_\alpha(x)$ is nonsingular for any index set α and any point x. It then follows that every point x is b-regular.

To prove s-regularity, we show that the point y in Definition 2.2.2 can be found. Note that for $i\in I_l\bigcup I_u\bigcup I^0_{el}\bigcup I^0_{eu}$, y_i is determined a priori by the equations in the definition. We can therefore eliminate these variables from the inequality system (8), which leaves a system of the following form:

$$\begin{bmatrix}\nabla_\alpha f_\alpha(x)^\top & \nabla_\beta f_\alpha(x)^\top & -\nabla_\gamma f_\alpha(x)^\top\\ \nabla_\alpha f_\beta(x)^\top & \nabla_\beta f_\beta(x)^\top & -\nabla_\gamma f_\beta(x)^\top\\ -\nabla_\alpha f_\gamma(x)^\top & -\nabla_\beta f_\gamma(x)^\top & \nabla_\gamma f_\gamma(x)^\top\end{bmatrix}\begin{bmatrix}y_\alpha\\ y_\beta\\ -y_\gamma\end{bmatrix}\begin{matrix}=\\ \ge\\ \ge\end{matrix}\begin{bmatrix}p\\ q_1\\ q_2\end{bmatrix},$$

and

$$\begin{bmatrix} y_\beta \\ -y_\gamma \end{bmatrix} \geq \begin{bmatrix} r_1 \\ r_2 \end{bmatrix},$$

where $\alpha := I_f^n(x)$, $\beta := I_f^l \bigcup I_{eu}^-$, and $\gamma := I_f^u \bigcup I_{el}^+$, and the vectors p, q_1, q_2, r_1, and r_2 are determined by the elimination of the variables. Note that the matrix in the system above is positive definite; thus, this system is of the form given by (20). A solution is therefore guaranteed by Lemma 2.2.16. □

It should be noted that the strongness in the monotonicity assumption above is essential. For example, consider the monotone function $f : \mathrm{R}_+ \to \mathrm{R}$ given by $f(x) = 1$, and let $\mathrm{B} := \mathrm{R}_+$. For this choice of f and B, it is easily verified that for all $x > 1$, x is neither b-regular or s-regular. As a consequence, even though MCP(f, B) has the trivial solution $x = 0$, NE/SQP fails to find it with any starting point $x > 1$. Thus, we see that NE/SQP cannot be relied upon to solve monotone linear complementarity problems.

We now state our main convergence result of the NE/SQP algorithm.

Theorem 2.2.18 *Suppose f is strongly monotone. If x^* is an accumulation point of the iterates $\{x^k\}$ produced by the NE/SQP algorithm, then x^* is a solution of MCP(f, B) and the sequence $\{x^k\}$ converges to x^* with the local convergence rates specified in Theorem 2.2.15.*

Proof By Lemma 2.2.17, x^* is both b-regular and s-regular. Therefore, by Theorem 2.2.12, x^* is a solution of MCP(f, B) and the iterates $\{x^k\}$ generated by the NE/SQP algorithm converge to x^* with convergence rates specified in Theorem 2.2.15. □

2.2.2 Modification of NE/SQP to Guarantee Finite Termination

The NE/SQP algorithm has the drawback that it does not necessarily terminate in a finite number of iterations unless it converges to a solution. In particular, while the algorithm guarantees descent of θ at every iteration, the sequence $\{\theta(x^k)\}$ may not converge to 0. This can happen either by generating an unbounded sequence of points, or by converging slowly to an irregular point. This will clearly be unacceptable if we are to use the algorithm to solve a sequence of perturbed subproblems. We therefore present a modified NE/SQP algorithm in Figure 2 which has the same local convergence properties as the original NE/SQP algorithm, but which also guarantees finite termination, even when it fails.

Figure 2: Modified NE/SQP Algorithm

Step 1 [Initialization] Given a starting vector $x^0 \in \mathrm{B}$, a convergence tolerance *tol*, and termination parameters $\gamma \in (0,1)$, and $\Delta \geq 11$, select $\rho, \sigma \in (0,1)$, and set $k = 0$.

Step 2 [Direction generation] Solve (QP_k), giving the direction d^k.
If $\phi(x^k, d^k) \geq (1-\gamma)\theta(x^k)$, or if $\left\|d^k\right\|^2 > \Delta\theta(x^0)$, then terminate the algorithm, returning the point x^k along with a failure message; otherwise, continue.

Step 3 [Steplength determination] Let m_k be the smallest nonnegative integer m such that

$$\theta(x^k + \rho^m d^k) - \theta(x^k) \leq -\sigma\rho^m z(x^k, d^k). \tag{23}$$

Set $x^{k+1} = x^k + \rho^{m_k} d^k$ and continue.

Step 4 [Termination check] If $\theta(x^{k+1}) \leq tol$ terminate the algorithm, returning the solution x^{k+1}. Otherwise, return to Step 2, with k replaced by $k+1$.

Note that by setting $\gamma = 0$ and $\Delta = \infty$, the modified algorithm is identical to NE/SQP, with the addition of a particular stopping criteria in Step 4. However, by choosing $\gamma \in (0,1)$ and $\Delta < \infty$, we can ensure that the algorithm will terminate in a finite number of iterations, which we will prove in Theorem 2.2.25. This has the drawback that the modified algorithm may fail when the original algorithm would have succeeded. However, we shall overcome this drawback in the QPCOMP algorithm by carefully controlling the parameter γ. Moreover, the modified algorithm also has the same local convergence properties as the original algorithm. To prove this fact, we shall show that if x^k is near a b-regular solution of $\mathrm{MCP}(f, \mathrm{B})$, then the tests in Step 2 can never cause failure.

Observe, that when x^k is close enough to a b-regular solution, $\left\|H(x^k)\right\| \leq \left\|H(x^0)\right\|/c$, so $\left\|d^k\right\| \leq \left\|H(x^0)\right\|$, and therefore, $\left\|d^k\right\|^2 \leq \Delta\theta(x^0)$. Thus, when x^k is close to a b-regular solution, the second test in Step 2 of the Modified NE/SQP algorithm cannot cause failure. We now show that the first test in Step 2 cannot cause failure either.

Lemma 2.2.19 *Let $\bar{x}$ be a solution of MCP(f, B). If $\bar{x}$ is b-regular, then for any $\epsilon \in (0, 1/2)$, there is a neighborhood N of $\bar{x}$ such that if $x^k \in N \bigcap B$, then*

$$\phi(x^k, d^k) \leq \epsilon\theta(x^k),$$

where d^k is an optimal solution of (QP_k).

Proof By Lemma 2.2.14, for x^k close enough to $\bar{x}$,

$$\left\|x^k - \bar{x}\right\| \leq \left\|x^k + d^k - \bar{x}\right\| + \left\|d^k\right\| \leq \left\|x^k + d^k - \bar{x}\right\| + c\left\|H(x^k)\right\|.$$

By Lemma 2.2.13, for any $\bar{\epsilon} > 0$, it follows that

$$\left\|x^k - \bar{x}\right\| \leq \bar{\epsilon}\left\|x^k - \bar{x}\right\| + c\left\|H(x^k)\right\|,$$

whenever x^k is close enough to $\bar{x}$. Rearranging terms,

$$\left\|x^k - \bar{x}\right\| \leq \frac{c}{1-\bar{\epsilon}}\left\|H(x^k)\right\| \leq 2c\left\|H(x^k)\right\|. \tag{24}$$

By the continuity of f, there is a neighborhood N of $\bar{x}$ such that if $x^k \in N \bigcap \mathrm{B}$, then the following hold:

$$\begin{aligned}
&I_f(\bar{x}) \subset I_f(x^k)\\
&I_l(\bar{x}) \subset I_l(x^k)\\
&I_u(\bar{x}) \subset I_u(x^k)\\
&I_{el}(\bar{x}) \subset I_{el}(x^k) \bigcup I_f(x^k) \bigcup I_l(x^k)\\
&I_{eu}(\bar{x}) \subset I_{eu}(x^k) \bigcup I_f(x^k) \bigcup I_u(x^k).
\end{aligned}$$

Without loss of generality, we can assume that N is small enough that (24) holds whenever $x^k \in N$.

Now, define $G^k(x) : \Omega \to \mathrm{R}^n$ by

$$G_i^k(x) := \begin{cases} x_i - l_i & i \in I_l(x^k) \bigcup I_{el}(x^k) \\ x_i - u_i & i \in I_u(x^k) \bigcup I_{eu}(x^k) \\ f_i(x) & i \in I_f(x^k). \end{cases}$$

It is easily seen that for $x^k \in N \bigcap \mathrm{B}$, $G^k(\bar{x}) = 0$. Note further that G^k is continuously differentiable, and that $\left\|G^k(x^k)\right\|^2/2 = \phi(x^k, 0)$. Further, for $i \notin I_f(x^k)$, we see that

$$\frac{1}{2}G_i^k(x)^2 = \phi_i(x^k, x - x^k).$$

Conversely, if $i \in I_f(x^k)$, then

$$G_i^k(x) = f_i(x) = f_i(x^k) + \nabla f_i(x^k)^\top (x - x^k) + o\left(\left\|x - x^k\right\|\right).$$

Thus,

$$\frac{1}{2} G_i^k(x)^2 = \frac{1}{2}\left(f_i(x^k) + \nabla f_i(x^k)^\top (x - x^k)\right)^2 + f_i(x^k) o\left(\left\|x - x^k\right\|\right) + o\left(\left\|x - x^k\right\|^2\right).$$

Summing over all i, we get

$$\frac{1}{2}\left\|G^k(x)\right\|^2 = \phi(x^k, x - x^k) + \left\|H(x^k)\right\| o\left(\left\|x - x^k\right\|\right) + o\left(\left\|x - x^k\right\|^2\right).$$

Choosing $x := \bar{x}$, we see that, since $G^k(\bar{x}) = 0$,

$$\phi(x^k, \bar{x} - x^k) = \left\|H(x^k)\right\| o\left(\left\|\bar{x} - x^k\right\|\right) + o\left(\left\|\bar{x} - x^k\right\|^2\right).$$

Thus, for any $\hat{\epsilon} > 0$, then for x^k close enough to $\bar{x}$,

$$\begin{aligned} \phi(x^k, \bar{x} - x^k) &\leq \hat{\epsilon}\left(\left\|H(x^k)\right\| \left\|\bar{x} - x^k\right\| + \left\|\bar{x} - x^k\right\|^2\right) \\ &\leq \hat{\epsilon}(2c + 4c^2)\left\|H(x^k)\right\|^2 \quad \text{by (24).} \end{aligned}$$

Finally, since $\bar{x} - x^k$ is feasible for (QP$_k$), we see that by choosing $\hat{\epsilon} \leq \epsilon/(4(c+2c^2))$,

$$\phi(x^k, d^k) \leq \phi(x^k, \bar{x} - x^k) \leq \epsilon\theta(x^k).$$

$\square$

The above lemmas show that for x^k close enough to $\bar{x}$, the modified algorithm will not terminate in Step 2, as long as $\bar{x}$ is b-regular. Thus, the modified algorithm has the same local convergence properties as the original algorithm. We have therefore proved the following theorem:

Theorem 2.2.20 *Under the conditions of Theorem 2.2.12, the Modified NE/SQP algorithm generates a well defined sequence of iterates $\{x^k\} \subset \boldsymbol{B}$, along with a sequence of optimal solutions $\{d^k\}$ for the subproblems (QP$_k$). Furthermore, if x^* is an accumulation point of $\{x^k\}$, and if either f is strongly monotone, or x^* is both b-regular and s-regular, then x^* is a solution of MCP(f, B) and the iterates converge to x^* at the rates specified in Theorem 2.2.15.*

The remainder of this section is aimed at proving that the Modified NE/SQP algorithm terminates. This is accomplished by considering what happens if the algorithm does not terminate. In this case, we shall show that the iterates $\{x^k\}$ converge to a point x^*. Using this fact, we will place bounds on certain quantities, which will then be used to establish a minimum rate of decrease for the merit function θ. This will then force the merit function to zero, which means that the algorithm will terminate after all, by the test in Step 4.

For ease of discussion, we define the function

$$\phi_x(d) := \phi(x, d).$$

The following lemma is a technical result needed in several ensuing proofs.

Lemma 2.2.21 *If $\phi_x(d) \le (1-\gamma)\theta(x)$ then*

$$z(x, d) \ge \frac{1}{2}\gamma^2\theta(x).$$

Proof Recall that $z(x,d) = \|b(x,d)\|^2/2$, where b is defined by (6). For simplicity, let $b := b(x,d)$. Note that

$$\phi_x(d) = \|H(x) + b\|^2/2 \ge \|H(x)\|^2/2 + \langle H(x), b\rangle = \theta(x) + \langle H(x), b\rangle.$$

Thus, by assumption,

$$(1-\gamma)\theta(x) \ge \phi_x(d) \ge \theta(x) + \langle H(x), b\rangle.$$

Subtracting $\theta(x)$ from both sides, we obtain

$$\langle H(x), b\rangle \le -\gamma\theta(x) = -\frac{\gamma}{2}\|H(x)\|^2.$$

But this implies that

$$\|b\| \ge \frac{\gamma}{2}\|H(x)\|,$$

so

$$z(x,d) = \|b\|^2/2 \ge \frac{\gamma^2}{4}\|H(x)\|^2 = \frac{\gamma^2}{2}\theta(x),$$

which establishes the lemma. □

We now prove that the iterates converge.

Lemma 2.2.22 *Suppose f is continuously differentiable. If the Modified NE/SQP algorithm, with $\gamma \in (0,1)$ and $\Delta < \infty$, fails to terminate, then the iterates $\{x^k\}$ produced by the algorithm will converge to a point $x^* \in$ **B** with $\theta(x^*) > 0$.*

Proof Let $\phi_k(d) := \phi(x^k, d)$ and let $z_k(d) := z(x^k, d)$. By the test in Step 2 of the algorithm, $\phi_k(d) \leq (1-\gamma)\theta(x^k)$. Thus, by Lemma 2.2.21, $z_k(d) \geq \frac{1}{2}\gamma^2\theta(x^k)$.

Let $\{\tau_k\}$ be the sequence of steplengths generated in step 3 of the algorithm, i.e., $\tau_k := \rho^{m_k}$. Then,

$$\begin{aligned} \theta(x^{k+1}) &= \theta(x^k + \tau_k d^k) \\ &\leq \theta(x^k) - \sigma\tau_k z_k(d^k) \quad \text{by the linesearch test (23)} \\ &\leq \theta(x^k) - \sigma\tau_k\gamma^2\theta(x^k)/2 \quad \text{by Lemma 2.2.21} \\ &= \left(1 - \frac{\sigma\tau_k\gamma^2}{2}\right)\theta(x^k). \end{aligned}$$

Let $\hat{\beta}_k := \sigma\tau_k\gamma^2/2$. Then

$$\theta(x^{k+1}) \leq \theta(x^0)\prod_{j=0}^{k}(1-\hat{\beta}_j).$$

Since $\theta(x^k)$ is bounded away from 0, it follows that

$$\prod_{k=0}^{\infty}(1-\hat{\beta}_k) > 0.$$

By Ahlfors (1966, Theorem 5 of Chapter 5), this implies that $\sum_{i=0}^{\infty}\hat{\beta}_k$ is finite, which means that $\sum_{i=0}^{\infty}\tau_k$ is finite.

Now, by the test in Step 2 of the algorithm, $\left\|d^k\right\|^2 \leq \Delta\theta(x^0)$. Thus, $\left\|d^k\right\|$ is bounded, so

$$\sum_{k=0}^{\infty}\tau_k\left\|d^k\right\| < \infty.$$

From this it follows that the sequence of iterates $\{x^k\}$ converges to some point x^*. Clearly, $\theta(x^*) > 0$, or the algorithm would terminate in Step 4. □

Using the fact that the iterates converge, we are now able to place bounds on several quantities, which we will use to prove Lemma 2.2.24.

Lemma 2.2.23 *Under the hypotheses of Lemma 2.2.22, there exist constants M_1, M_2, and L, depending on the starting point x^0, such that for all $\tau \in [0,1]$, the following inequalities hold:*

$$|f_i(x^k + \tau d^k)| \leq M_1, \qquad |\nabla f_i(x^k + \tau d^k)| \leq M_2 \tag{25}$$

and

$$f_i(x^k) - \tau L\left\|d^k\right\| \leq f_i(x^k + \tau d^k) \leq f_i(x^k) + \tau L\left\|d^k\right\|. \tag{26}$$

Furthermore, for any $\delta > 0$, we can choose $\hat{\tau}(\delta) > 0$ such that for k sufficiently large, the following holds for all $\tau \in [0, \hat{\tau}(\delta)]$:

$$|f_i(x^k + \tau d^k)| \leq \left| f_i(x^k) + \tau \nabla f_i(x^k)^\top d^k \right| + \tau\delta \left\| d^k \right\|. \tag{27}$$

Proof By Lemma 2.2.22, $\{x^k\}$ converges. Also, $\left\| d^k \right\|$ is bounded by the constant $\sqrt{\Delta\theta(x^0)}$. Thus, there is a compact set $S \subset \mathrm{B}$ such that for all $\tau \in [0, 1]$, $x^k + \tau d^k \in S, \forall k$. Since f is continuously differentiable, f and ∇f are bounded on S, from which we get (25). Furthermore, f is Lipschitz continuous on S, from which (26) follows.

Since f is continuously differentiable, then for any $\delta > 0$, there exists an $\epsilon > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq \delta \quad \forall x, y \in N_\epsilon(x^*), \tag{28}$$

where the neighborhood N_ϵ is defined by $N_\epsilon(x^*) := \{x : \|x - x^*\| < \epsilon\}$.

Since x^k converges to x^*, then for k large enough, $x^k \in N_{\epsilon/2}(x^*)$. Moreover, by Step 2 of the algorithm, $\left\| d^k \right\| \leq \sqrt{\Delta\theta x^0}$. Thus, if we let $\hat{\tau}(\delta) := \epsilon/(2\sqrt{\Delta\theta(x^0)})$, it follows that $\left\| \hat{\tau}(\delta) d^k \right\| \leq \epsilon/2$. Thus, $x + \tau d^k \in N_\epsilon(x^*), \forall \tau \in [0, \hat{\tau}(\delta)]$.

Thus, for any $\delta > 0$, we can choose $\hat{\tau}(\delta) > 0$ such that for k sufficiently large, the following holds for all $\tau \in [0, \hat{\tau}(\delta)]$:

$$\begin{aligned} |f_i(x^k + \tau d^k)| &= \left| f_i(x^k) + \tau \nabla f_i(x^k + \tilde{\tau} d^k)^\top d^k \right| \quad \text{for some } \tilde{\tau} \in [0, \tau] \\ &= \left| f_i(x^k) + \tau \nabla f_i(x^k)^\top d^k + \tau \left(\nabla f_i(x^k + \tilde{\tau} d^k)^\top - \nabla f_i(x^k)^\top \right) d^k \right| \\ &\leq \left| f_i(x^k) + \tau \nabla f_i(x^k)^\top d^k \right| + \tau\delta \left\| d^k \right\| \quad \text{by (28)}. \end{aligned}$$

$\square$

We are now able to establish a minimum rate of decrease for the merit function.

Lemma 2.2.24 *Under the hypotheses of Lemma 2.2.22, there exists a constant $\hat{\rho} \in (0, 1)$ such that*

$$\theta(x^{k+1}) \leq \hat{\rho}\theta(x^k), \quad \forall\ k \textit{ sufficiently large.}$$

Proof Suppose $\delta \in (0, 1)$, and let $\tau \in [0, \hat{\tau}(\delta)]$ where $\hat{\tau}(\delta)$ is chosen according to Lemma 2.2.23. Suppose that k is large enough that (27) holds. We shall examine the terms $H_i(x^k + \tau d^k)^2$ in order to establish an upper bound on $\theta(x^{k+1}) = \sum_i H_i(x^k + \tau d^k)^2/2$.

To simplify notation, we drop the superscripts k. Thus, we let $x := x^k$ and $d := d^k$, etc. We shall also find it convenient to define the scalar function $\hat{\phi}_i : \mathrm{R}_+ \to \mathrm{R}_+$, as follows:

$$\hat{\phi}_i(\tau) \;\; := \;\; \phi_i(x, \tau d).$$

Observe that $\hat{\phi}_i''(0) = z_i(x, d)$, so

$$\sum_{i=1}^{n} \hat{\phi}_i''(0) = z(x, d). \tag{29}$$

To bound $H_i(x + \tau d)^2$, we have to look at two different cases:

Case 1: $i \in I_f(x)$. Note that $|H_i(x + \tau d)| \leq |f_i(x + \tau d)|$. Thus, by (27),

$$H_i(x+\tau d)^2 \leq (f_i(x)+\tau\nabla f_i(x)^\top d)^2+2\tau\delta \left|f_i(x) + \tau\nabla f_i(x)^\top d\right| \|d\|+\tau^2\delta^2 \|d\|^2 .$$

But, $(f_i(x) + \tau\nabla f_i(x)^\top d)^2 = 2\hat{\phi}_i(\tau) = 2\hat{\phi}_i(0) + 2\tau\hat{\phi}_i'(0) + \tau^2\hat{\phi}_i''(0)$, so

$$\begin{aligned} H_i(x + \tau d)^2 \;\; \leq \;\; & 2\hat{\phi}_i(0) + 2\tau\hat{\phi}_i'(0) + \tau^2\hat{\phi}_i''(0) \\ & +2\tau\delta \left|f_i(x) + \tau\nabla f_i(x)^\top d\right| \|d\| + \tau^2\delta^2 \|d\|^2 . \end{aligned} \tag{30}$$

Case 2: $i \notin I_f(x)$. We look only at the case $i \in I_l(x) \bigcup I_{el}(x)$; the argument for $i \in I_u(x) \bigcup I_{eu}(x)$ is similar.

If $H_i(x + \tau d)$ is negative, then

$$\begin{aligned} H_i(x + \tau d) \;\; &= \;\; f_i(x + \tau d) \\ &\geq \;\; f_i(x) - \tau L \|d\| \quad \text{by (26)} \\ &\geq \;\; x_i - l_i + \tau d_i - \tau(d_i + L \|d\|) \quad \text{since } f_i(x) \geq x_i - l_i \\ &\geq \;\; x_i - l_i + \tau d_i - \tau(L + 1) \|d\| . \end{aligned}$$

Thus,

$$\begin{aligned} H_i(x + \tau d)^2 \;\; &\leq \;\; (x_i - l_i + \tau d_i)^2 - 2\tau(x_i - l_i + \tau d_i)(L + 1) \|d\| + \\ & \qquad \tau^2(L + 1)^2 \|d\|^2 \\ &\leq \;\; (x_i - l_i + \tau d_i)^2 + \tau^2(L + 1)^2 \|d\|^2 . \end{aligned}$$

If $H_i(x+\tau d)$ is nonnegative, this inequality holds trivially since $H_i(x+\tau d_i) \leq x_i - l_i + \tau d_i$. Finally, $(x_i - l_i + \tau d_i)^2 = 2\hat{\phi}_i(\tau) = 2\hat{\phi}_i(0) + 2\tau\hat{\phi}_i'(0) + \tau^2\hat{\phi}_i''(0)$, so

$$H_i(x + \tau d)^2 \leq 2\hat{\phi}_i(0) + 2\tau\hat{\phi}_i'(0) + \tau^2\hat{\phi}_i''(0) + \tau^2(L + 1)^2 \|d\|^2 . \tag{31}$$

Summing over all components, we get

$$\theta(x+\tau d) = \frac{1}{2}\sum H_i(x+\tau d)^2 \le \phi_x(0) + \tau\phi_x'(0;d) + \tau\delta\eta + \tau^2\zeta, \tag{32}$$

where

$$\eta := \sum_{i\in I_f(x)} \left| f_i(x) + \tau\nabla f_i(x)^\top d \right| \|d\|,$$

and

$$\zeta := \sum_{i=1}^{n} \hat{\phi}_i''(0) + \sum_{i\notin I_f(x)} (L+1)^2 \|d\|^2 + \sum_{i\in I_f(x)} \delta^2 \|d\|^2.$$

We now establish bounds for η and ζ. By (25),

$$\sum_{i\in I_f(x)} \left| f_i(x) + \tau\nabla f_i(x)^\top d \right| \le \|f(x)\| + \tau M_2 \|d\| \le M_1 + \tau M_2\sqrt{\Delta\theta(x^0)} =: C_1.$$

Thus, $\eta \le C_1 \|d\| \le C_1\sqrt{\Delta\theta(x^0)} =: K_1$.

For ζ, we deduce from (29) that

$$\sum_{i=1}^{n} \hat{\phi}_i''(0) = z(x,d) \le \theta(x), \qquad \text{by item 4 of Proposition 2.2.4.}$$

Thus,

$$\begin{aligned} \zeta &\le \theta(x) + \|d\|^2 (n(L+1)^2 + n\delta^2) \\ &\le (1 + n\Delta((L+1)^2+\delta^2))\,\theta(x^0), \quad \text{since } \|d\|^2 \le \Delta\theta(x^0) \\ &\le K_2, \end{aligned}$$

where $K_2 := (1+n\Delta((L+1)^2+1))\theta(x^0)$. This last inequality holds since $\delta \le 1$. Returning to (32), we find that

$$\begin{aligned} \theta(x+\tau d) &\le \phi_x(0) + \tau\phi_x'(0;d) + \tau\delta K_1 + \tau^2 K_2 \\ &= \theta(x) + \tau\theta'(x;d) + \tau\delta K_1 + \tau^2 K_2. \end{aligned}$$

By Item 2 of Proposition 2.2.4,

$$\begin{aligned} \theta'(x;d) &\le \phi_x(d) - \phi_x(0) - z(x,d) \\ &\le (1-\gamma)\theta(x) - \theta(x) - z(x,d), \quad \text{by the test in Step 2} \\ &= -\gamma\theta(x) - z(x,d). \end{aligned}$$

Thus,

$$\theta(x+\tau d)-\theta(x) \leq \tau\left(-\gamma\theta(x)-z(x,d)\right)+\tau\delta K_1+\tau^2 K_2.$$

Note that the definitions of K_1 and K_2 are independent of δ. We can therefore consider a particular choice of δ: let $\delta := \min(1, \gamma\theta(x^*)/(2K_1))$ and let $\bar{\tau} := \min(\hat{\tau}(\delta), \gamma\theta(x^*)/(2K_2))$. Note that $\delta > 0$ and $\bar{\tau} > 0$, since $\theta(x^*) > 0$. Further, since $\theta(x^*) \leq \theta(x)$, it follows that for all $\tau \leq \bar{\tau}$, and for k sufficiently large,

$$\begin{aligned}\theta(x+\tau d)-\theta(x) &\leq -\tau z(x,d)-\tau\gamma\theta(x)+\tau\gamma\theta(x^*)/2+\tau\bar{\tau}K_2\\ &\leq -\tau z(x,d)-\tau\gamma\theta(x)+\tau\gamma\theta(x)/2+\tau\gamma\theta(x)/2\\ &= -\tau z(x,d)\\ &\leq -\sigma\tau z(x,d), \quad \forall\, \sigma \leq 1.\end{aligned} \tag{33}$$

Observe that the steplength ρ^m generated by Step 3 of the algorithm is chosen such that m is the *smallest* integer satisfying (23). Thus, $\tau := \rho^{m-1}$ cannot satisfy (33). But this means that

$$\rho^{m-1} \geq \bar{\tau}, \text{ which implies } \rho^m \geq \rho\bar{\tau}.$$

From this, it follows by the linesearch test (23) that

$$\begin{aligned}\theta(x+\rho^m d) &\leq \theta(x)-\sigma\rho\bar{\tau}z(x,d)\\ &\leq (1-\frac{\sigma\rho\bar{\tau}\gamma^2}{2})\theta(x), \quad \text{by Lemma 2.2.21.}\end{aligned}$$

By setting $\hat{\rho} := 1-\sigma\rho\bar{\tau}\gamma^2/2$, we complete the proof. □

Theorem 2.2.25 *If $\gamma \in (0,1)$ and $\Delta < \infty$, then the modified NE/SQP algorithm will terminate in a finite number of iterations provided that f is continuously differentiable on $\boldsymbol{B}$.*

Proof Let $tol > 0$ be the stopping tolerance used in the algorithm. If the algorithm does not terminate, then by Lemma 2.2.24, there exists $\hat{\rho} \in (0,1)$ such that for k sufficiently large,

$$\theta(x^{k+1}) \leq \hat{\rho}\theta(x^k).$$

Thus, after sufficiently many iterations, $\theta(x^k) < tol$, and the algorithm will terminate in Step 4. □

2.3 The QPCOMP Algorithm

The basic idea behind QPCOMP is simple. The algorithm first tries to solve the problem using the modified NE/SQP algorithm. If this fails, QPCOMP then solves a sequence of perturbed problems in order to find a point with an improved value of the merit function. Once this point is found, QPCOMP returns to running the modified NE/SQP algorithm on the original problem, starting from this improved point.

One complication of the algorithm is that the subproblems must be solved inexactly in order to guarantee that they are each completed in a finite amount of time. To handle this we have introduced a sequence of tolerances $\{\eta_j\}$ which control the accuracy demanded by each subproblem.

Another complication is that the best choices of the parameters λ and γ cannot be known in advance. Thus, the algorithm must choose these parameters adaptively. The algorithm is given is Figure 3.

Observe, that the QPCOMP algorithm has the same local convergence properties as NE/SQP. In particular, by Theorem 2.2.20, for any b-regular solution x^*, there is a neighborhood such that the modified NE/SQP algorithm is identical to NE/SQP within this neighborhood. Thus, in Step 2 of the QPCOMP algorithm, if x^k is sufficiently close to x^*, then the modified NE/SQP algorithm will converge to x^* at the rates specified by Theorem 2.2.15.

We now establish global convergence properties for the algorithm:

Theorem 2.3.1 *If f is Lipschitz continuous and continuously differentiable on B, and if MCP(f,B) satisfies Assumption 2.1.2, then for any $\epsilon > 0$ the QPCOMP algorithm generates an iterate x^k satisfying $\theta(x^k) < \epsilon$ in a finite number of iterations.*

The remainder of this section is devoted to proving this theorem. As an introduction to the proof, note that if Step 4 is always successful at generating an improved starting point, then even if the Modified NE/SQP always fails in Step 2, the merit function values $\{\theta(x^k)\}$ will converge to 0 at least linearly, since $\theta(x^{k+1}) \leq \mu\theta(x^k)$ for all k. Thus, our convergence analysis is reduced to proving that Step 4 always generates an improved starting point.

In the analysis that follows, it will be convenient to define perturbed index

Figure 3: Algorithm QPCOMP

Step 1 [Initialization] Given a starting vector $x^0 \in \mathbf{B}$ and a convergence tolerance $\epsilon > 0$, choose $\delta > 0$, $\mu \in (0, 1)$, $\gamma \in (0, 1)$, $\nu \in (0, 1)$, and set $k = 0$.

Step 2 [Attempt NE/SQP] Run the Modified NE/SQP algorithm with starting point x^k, with $tol = \epsilon$. This generates a point $\tilde{x}$.

Step 3 [Termination check] If $\tilde{x}$ solves MCP(f, B), stop; otherwise continue with step 4.

Step 4 [Generate better starting point] Set $\theta_{best} := \theta(\tilde{x})$, set $y^0 = \tilde{x}$, set $j = 0$, and choose $\lambda > 0$, and choose a positive sequence $\{\eta_j\} \downarrow 0$.

Step 4a Run the Modified NE/SQP algorithm to solve the perturbed problem MCP(f^{λ, y^j}, B) from starting point y^j, with $tol = \eta_j/(1 + \|y^j\|)$. This generates a point $\tilde{y}$.

Step 4b If $\tilde{y}$ fails to solve the perturbed problem to the requested accuracy, set $\lambda \geq \lambda + \delta$ and $\gamma \leq \nu\gamma$, and goto step 4a; otherwise, continue.

Step 4c [Check point] If $\theta(\tilde{y}) \leq \mu\theta_{best}$, set $x^{k+1} = \tilde{y}$ and return to step 2, with k replaced by $k + 1$. Otherwise, set $y^{j+1} := \tilde{y}$ and return to step 4a, with j replaced by $j + 1$.

sets by

$$\begin{array}{rcl} I_l^{\lambda,\bar{x}}(x) & := & \{i : x_i - l_i < f_i^{\lambda,\bar{x}}(x)\} \\ I_{el}^{\lambda,\bar{x}}(x) & := & \{i : x_i - l_i = f_i^{\lambda,\bar{x}}(x)\} \\ I_f^{\lambda,\bar{x}}(x) & := & \{i : x_i - u_i < f_i^{\lambda,\bar{x}}(x) < x_i - l_i\} \\ I_{eu}^{\lambda,\bar{x}}(x) & := & \{i : x_i - u_i = f_i^{\lambda,\bar{x}}(x)\} \\ I_u^{\lambda,\bar{x}}(x) & := & \{i : x_i - u_i > f_i^{\lambda,\bar{x}}(x)\}. \end{array}$$

We shall also use the following obvious perturbations of the functions H, θ, ϕ, and z:

$$\begin{array}{l} H^{\lambda,\bar{x}}(x) := \min(x_i - l_i, \max(x_i - u_i, f_i^{\lambda,\bar{x}}(x))), \\ \theta^{\lambda,\bar{x}}(x) := \frac{1}{2}\left\| H^{\lambda,\bar{x}}(x)\right\|^2, \\ \phi_x^{\lambda,\bar{x}}(d) := \phi^{\lambda,\bar{x}}(x,d) := \sum \phi_i^{\lambda,\bar{x}}(x,d), \text{ where for } i = 1,\ldots,n, \\ \qquad \phi_i^{\lambda,\bar{x}}(x,d) := \begin{cases} \frac{1}{2}(x_i - l_i + d_i)^2 & i \in I_l^{\lambda,\bar{x}}(x) \bigcup I_{el}^{\lambda,\bar{x}}(x) \\ \frac{1}{2}(x_i - u_i + d_i)^2 & i \in I_u^{\lambda,\bar{x}}(x) \bigcup I_{eu}^{\lambda,\bar{x}}(x) \\ \frac{1}{2}(f_i^{\lambda,\bar{x}}(x) + \nabla f_i^{\lambda,\bar{x}}(x)^\top d)^2 & i \in I_f^{\lambda,\bar{x}}(x). \end{cases} \\ z_x^{\lambda,\bar{x}}(d) := z^{\lambda,\bar{x}}(x,d) := \sum z^{\lambda,\bar{x}}(x,d), \text{ where for } i = 1,\ldots,n, \\ \qquad z_i^{\lambda,\bar{x}}(x,d) := \begin{cases} \frac{1}{2}d_i^2 & i \notin I_f^{\lambda,\bar{x}}(x) \\ \frac{1}{2}(\nabla f_i^{\lambda,\bar{x}}(x)^\top d)^2 & i \in I_f^{\lambda,\bar{x}}(x). \end{cases} \end{array}$$

To show that Step 4 is always successful at generating an improved starting point, we begin by assuming that the Modified NE/SQP algorithm in Step 4a of QPCOMP fails at most a finite number of times. Later will shall remove this assumption. It follows that after a finite number of iterations, $\tilde{y}$ always solves the perturbed problem to the desired accuracy, so the algorithm always continues past Step 4b to Step 4c. Thus, either an improved point will eventually be found, or the algorithm will generate a sequence of iterates $\{y^j\}$ such that

$$\left\| H^{\lambda,y^j}(y^{j+1})\right\| \leq \frac{\eta_j}{1 + \|y^j\|}.$$

We then use the fact that $\{\eta_j\}$ converges to 0 to show that $\theta(y^j) \to 0$. This result is proved in the following lemma:

Lemma 2.3.2 *Let f be a Lipschitz continuous function and let $\{\eta_k\}$ be a sequence of positive numbers that converges to* 0. *Let $\lambda > 0$ and let $\{x^k\}$ be a sequence of points in B such that*

$$\left\| H^{\lambda,x^k}(x^{k+1})\right\| \leq \frac{\eta_k}{1 + \|x^k\|}, \qquad \forall k. \tag{34}$$

Suppose MCP(f, B) satisfies Assumption 2.1.2, then for any $\epsilon > 0$, there exists an iterate $x^j \in \{x^k\}$ such that $\theta(x^j) \leq \epsilon$.

Proof Let x^* be the solution to MCP(f, B) guaranteed by Assumption 2.1.2 which satisfies (1), and let $y^k := H^{\lambda,x^k}(x^{k+1})$. In the same spirit as the proof of Theorem 2.1.3, we establish a lower bound on the term $(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1})$.

Case 1: $y_i^k = x_i^{k+1} - l_i$ **and** $x_i^* < x_i^{k+1}$**.** Observe that

$$(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) = (x_i^{k+1} - x_i^*)\frac{f_i(x^{k+1}) - y_i}{\lambda} + w_i^k, \tag{35}$$

where

$$w_i^k := (x_i^{k+1} - x_i^*)\left(x_i^k - x_i^{k+1} - \frac{f_i(x^{k+1}) - y_i}{\lambda}\right).$$

Now, $0 < (x_i^{k+1} - x_i^*) \leq x_i^{k+1} - l_i = y_i^k$. Also, $x_i^k - x_i^{k+1} \geq l_i - x_i^{k+1} = -y_i^k$. Thus,

$$\begin{aligned} w_i^k &= (x_i^{k+1} - x_i^*)\left(x_i^k - x_i^{k+1} + y_i/\lambda\right) - (x_i^{k+1} - x_i^*)(f_i(x^{k+1})/\lambda) \\ &\geq y_i^k\left(-y_i^k + y_i^k/\lambda\right) - |y_i^k|\left|f_i(x^{k+1})\right|/\lambda \\ &\geq -(y_i^k)^2 - |y_i^k|\left|f_i(x^{k+1})\right|/\lambda. \end{aligned}$$

Returning to (35), we get

$$(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) \geq \tag{36}$$

$$(x_i^{k+1} - x_i^*)\left(f_i(x^{k+1}) - y_i\right)/\lambda - (y_i^k)^2 - \frac{|y_i^k|}{\lambda}\left|f_i(x^{k+1})\right|. \tag{37}$$

Case 2: $y_i^k = x_i^{k+1} - l_i$, **and** $x_i^* \geq x_i^{k+1}$**.** In this case, $f_i^{\lambda,x^k}(x^{k+1}) \geq x_i^{k+1} - l_i = y_i^k$. Thus,
$f_i(x^{k+1}) + \lambda(x_i^{k+1} - x_i^k) \geq y_i^k$, so $x_i^k - x_i^{k+1} \leq (f_i(x^{k+1}) - y_i^k)/\lambda$. Since $x_i^{k+1} - x_i^* \leq 0$, we get

$$(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) \geq (x_i^{k+1} - x_i^*)\left(f_i(x^{k+1}) - y_i\right)/\lambda. \tag{38}$$

Case 3: $y_i^k = f_i^{\lambda,x^k}(x^{k+1})$**.** In this case, $y_i^k = f_i(x^{k+1}) + \lambda(x_i^{k+1} - x_i^k)$, so $x_i^k - x_i^{k+1} = (f_i(x^{k+1}) - y_i^k)/\lambda$. Thus,

$$(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) = (x_i^{k+1} - x_i^*)\left(f_i(x^{k+1}) - y_i^k\right)/\lambda.$$

Case 4: $y_i^k = x_i^{k+1} - u_i, x_i^{k+1} \geq x_i^*$. By similar arguments to Case 2, inequality (38) is satisfied.

Case 5: $y_i^k = x_i^{k+1} - u_i, x_i^{k+1} < x_i^*$. By similar arguments to Case 1, inequality (36) is satisfied.

In every case above, inequality (36) holds: Thus,

$$\begin{aligned}
(x_i^k - x_i^*)^2 &= (x_i^{k+1} - x_i^* + x_i^k - x_i^{k+1})^2 \\
&= (x_i^{k+1} - x_i^*)^2 + 2(x_i^{k+1} - x_i^*)(x_i^k - x_i^{k+1}) + (x_i^k - x_i^{k+1})^2 \\
&\geq (x_i^{k+1} - x_i^*)^2 + 2(x_i^{k+1} - x_i^*)\left(f_i(x^{k+1}) - y_i^k\right)/\lambda - 2(y_i^k)^2 \\
&\quad -\tfrac{2}{\lambda}|y_i^k|\left|f_i(x^{k+1})\right| + (x_i^k - x_i^{k+1})^2, \qquad \text{by (36)}.
\end{aligned}$$

Summing over all components, we get

$$\begin{aligned}
\left\|x^k - x^*\right\|^2 &\geq \left\|x^{k+1} - x^*\right\|^2 + 2\left\langle f(x^{k+1}), x^{k+1} - x^*\right\rangle/\lambda - 2\left\langle y^k, x^{k+1} - x^*\right\rangle/\lambda \\
&\quad -2\left\|y^k\right\|^2 - 2n\left\|y^k\right\|\left\|f(x^{k+1})\right\|/\lambda + \left\|x^k - x^{k+1}\right\|^2.
\end{aligned}$$

Now, let L be the Lipschitz constant for f. Then

$$\left\|f(x^{k+1})\right\| \leq \left\|f(x^{k+1}) - f(x^*)\right\| + \|f(x^*)\| \leq L\left\|x^{k+1} - x^*\right\| + \|f(x^*)\|.$$

Further, by Assumption 2.1.2, $\left\langle f(x^{k+1}), x^{k+1} - x^*\right\rangle \geq 0$. Thus,

$$\begin{aligned}
\left\|x^k - x^*\right\|^2 &\geq \left\|x^{k+1} - x^*\right\|^2 - 2\left\|y^k\right\|\left\|x^{k+1} - x^*\right\|/\lambda - 2\left\|y^k\right\|^2 \\
&\quad -2n\left\|y^k\right\|\left(L\left\|x^{k+1} - x^*\right\| + \|f(x^*)\|\right)/\lambda + \left\|x^k - x^{k+1}\right\|^2 \\
&\geq \left\|x^{k+1} - x^*\right\|^2 - 2\frac{\eta_k}{\lambda}\left\|x^{k+1} - x^*\right\|/\left(1 + \left\|x^k\right\|\right) - 2\eta_k^2/\left(1 + \left\|x^k\right\|\right)^2 \\
&\quad -2n\frac{\eta_k}{\lambda}\left(L\left\|x^{k+1} - x^*\right\| + \|f(x^*)\|\right)/\left(1 + \left\|x^k\right\|\right) \\
&\quad +\left\|x^k - x^{k+1}\right\|^2 \quad \text{by (34)} \\
&= \left\|x^{k+1} - x^*\right\|^2 + \left\|x^k - x^{k+1}\right\|^2 - 2\eta_k\beta_k, \qquad (39)
\end{aligned}$$

where

$$\beta_k := \frac{\left\|x^{k+1} - x^*\right\|}{\lambda(1 + \|x^k\|)} + \frac{\eta_k}{(1 + \|x^k\|)^2} + \frac{n\left(L\left\|x^{k+1} - x^*\right\| + \|f(x^*)\|\right)}{\lambda(1 + \|x^k\|)}.$$

Note that

$$\beta_k \leq \left\|x^{k+1} - x^*\right\|(nL + 1)/\lambda + \eta_0 + n\|f(x^*)\|/\lambda. \qquad (40)$$

Let $C := (nL+1)/\lambda + \eta_0 + n\left\|f(x^*)\right\|/\lambda$. Then $\beta_k \geq C$ implies that $\left\|x^{k+1} - x^*\right\| \geq 1$. Now, let $\{\beta_k : k \in \kappa\}$ be the subsequence of $\{\beta_k\}$ for which $\beta_k \geq C, \forall k \in \kappa$. It follows then that $\left\|x^{k+1} - x^*\right\| \geq 1, \forall k \in \kappa$. If we divide each side of (40) by $\left\|x^{k+1} - x^*\right\|^2$, it is easily seen that $\{\beta_k / \left\|x^{k+1} - x^*\right\|^2 : k \in \kappa\}$ is bounded.

However, dividing (39) by $\left\|x^{k+1} - x^*\right\|^2$ gives

$$\frac{\left\|x^k - x^*\right\|^2}{\left\|x^{k+1} - x^*\right\|^2} \geq 1 + \frac{\left\|x^k - x^{k+1}\right\|^2}{\left\|x^{k+1} - x^*\right\|^2} - \frac{2\eta_k\beta_k}{\left\|x^{k+1} - x^*\right\|^2}.$$

Since $\eta_k \downarrow 0$, the last term above converges to 0 on κ. Thus, for $k \in \kappa$ large enough,

$$\frac{\left\|x^k - x^*\right\|}{\left\|x^{k+1} - x^*\right\|} > \frac{1}{2},$$

and

$$\beta_k \leq \frac{2\left\|x^k - x^*\right\|}{\lambda(1 + \left\|x^k\right\|)} + \frac{\eta_k}{(1 + \left\|x^k\right\|)^2} + \frac{n\left(2L\left\|x^k - x^*\right\| + \left\|f(x^*)\right\|\right)}{\lambda(1 + \left\|x^k\right\|)}.$$

Observe that

$$\frac{\left\|x^k - x^*\right\|}{(1 + \left\|x^k\right\|)} \leq \frac{\left\|x^*\right\| + \left\|x^k\right\|}{(1 + \left\|x^k\right\|)} \leq \max(1, \left\|x^*\right\|).$$

Thus, the subsequence $\{\beta_k : k \in \kappa\}$ is bounded, from which it follows that $\{\beta_k\}$ is bounded.

Now, assume the lemma is false. Then there exists an $\epsilon > 0$ such that for all k, $\theta(x^k) > \epsilon^2/2$, which implies $\left\|H(x^k)\right\| > \epsilon$. Furthermore, for k large enough, $\eta_k < \epsilon^2$. Without loss of generality, we can assume that this inequality holds for all k.

Since f is Lipschitz continuous, H^{λ,x^k} is also Lipschitz continuous with some Lipschitz constant K. But then,

$$\begin{aligned}
\epsilon - \epsilon^2 &< \left\|H(x^k)\right\| - \eta_k \\
&\leq \left\|H^{\lambda,x^k}(x^k)\right\| - \left\|H^{\lambda,x^k}(x^{k+1})\right\|\left(1 + \left\|x^k\right\|\right) \\
&\leq \left\|H^{\lambda,x^k}(x^k) - H^{\lambda,x^k}(x^{k+1})\right\| \\
&\leq K\left\|x^{k+1} - x^k\right\|
\end{aligned}$$

Thus, for ϵ small enough,

$$\epsilon/(2K) < (\epsilon - \epsilon^2)/K < \left\|x^{k+1} - x^k\right\|.$$

Finally, since the sequence $\{\eta_k\beta_k\}$ converges to 0, then for all k sufficiently large, $\eta_k\beta_k < \epsilon/(8K)$. Thus, from (39)

$$\begin{aligned}\left\|x^k - x^*\right\|^2 &\geq \left\|x^{k+1} - x^*\right\|^2 + \left\|x^{k+1} - x^k\right\|^2 - 2\eta_k\beta_k \\ &\geq \left\|x^{k+1} - x^*\right\|^2 + \epsilon/(2K) - \epsilon/(4K) \\ &= \left\|x^{k+1} - x^*\right\|^2 + \epsilon/(4K).\end{aligned}$$

But, then

$$\left\|x^k - x^*\right\|^2 \geq \sum_{k+1}^{\infty} \epsilon/(4K) = \infty > \left\|x^k - x^*\right\|^2.$$

But this contradicts the assumption that $\epsilon > 0$. The lemma is thus proved by contradiction. $\square$

Note that Lemma 2.3.2 did not make any assumption on the choice of λ other than that it is greater than 0. Thus, even if λ is smaller than the Lipschitz constant, we can guarantee convergence.

The next stage in our analysis is to prove that the Modified NE/SQP algorithm can fail at most a finite number of times in Step 4a of QPCOMP. This is accomplished by observing that after each failure, the value of λ is increased, while the value of γ is decreased. Thus, the result will be proved if we can show that for λ large enough, and γ small enough, the Modified NE/SQP algorithm will always solve the perturbed problem MCP(f^{λ,y^j}, B). This is accomplished in the following two lemmas.

Lemma 2.3.3 *Suppose f is Lipschitz continuous with Lipschitz constant L, and let x and $\bar{x}$ be arbitrary points in $\mathbb{B}$. If $\lambda > 2L + 2$, and if $\bar{d}$ satisfies $\phi_x^{\lambda,\bar{x}}(\bar{d}) \leq \phi_x^{\lambda,\bar{x}}(0)$, then*

$$\left\|\bar{d}\right\|^2 < 11\,\theta^{\lambda,\bar{x}}(x).$$

Proof For simplicity of notation, we shall drop the superscripts $\lambda, \bar{x}$; that is, let $\phi := \phi^{\lambda,\bar{x}}$, etc. Also, let $I_f := I_f(x) := I_f^{\lambda,\bar{x}}(x)$. Observe that $\phi_x(d)$ is a quadratic function with Hessian B given by

$$B := (M + D + I)^\top (M + D + I)$$

where M is the $n \times n$ matrix whose ith row is given by

$$M_{i,\cdot} := \begin{cases} \nabla f_i^\top(x) & \text{if } i \in I_f \\ 0 & \text{if } i \notin I_f, \end{cases} \tag{41}$$

and D is the diagonal matrix with entries

$$D_{ii} := \begin{cases} \lambda - 1 & \text{if } i \in I_f \\ 0 & \text{if } i \notin I_f. \end{cases}$$

We then see that

$$\begin{aligned} d^\top B d &= \|Md\|^2 + 2d^\top DMd + \|Dd\|^2 + 2d^\top Md + 2d^\top Dd + \|d\|^2 \\ &= \|Md\|^2 + 2(\lambda-1)d_{I_f}^\top (Md)_{I_f} + (\lambda-1)^2 \left\|d_{I_f}\right\|^2 + 2d_{I_f}^\top (Md)_{I_f} + \\ &\quad 2(\lambda-1)\left\|d_{I_f}\right\|^2 + \|d\|^2 \\ &\geq \|Md\|^2 - 2(\lambda-1)\left\|d_{I_f}\right\| \left\|(Md)_{I_f}\right\| + (\lambda-1)^2 \left\|d_{I_f}\right\|^2 \\ &\quad -2\left\|d_{I_f}\right\| \left\|(Md)_{I_f}\right\| + 2(\lambda-1)\left\|d_{I_f}\right\|^2 + \|d\|^2 \\ &= \|d\|^2 \left(a^2 - 2(\lambda-1)ab + (\lambda-1)^2 b^2 - 2ab + 2(\lambda-1)b^2 + 1\right), \end{aligned}$$

where

$$a := \frac{\|Md\|}{\|d\|} = \frac{\left\|(Md)_{I_f}\right\|}{\|d\|} \quad \text{and} \quad b := \frac{\left\|d_{I_f}\right\|}{\|d\|}.$$

Note that $a \leq L$ and $b \leq 1$. Simplifying the inequality above, we obtain

$$\begin{aligned} d^\top Bd &\geq \|d\|^2 \left(a^2 - 2\lambda ab + \lambda^2 b^2 - b^2 + 1\right) \\ &= \|d\|^2 \left((a - \lambda b)^2 + (1 - b^2)\right). \end{aligned}$$

If $b \geq 1/2$, then $\lambda b \geq 2(L+1)b \geq L+1$. Since $a \leq L$, we get $(a - \lambda b)^2 \geq 1$. Conversely, if $b \leq 1/2$, then $1 - b^2 \geq 3/4$. Thus, in either case,

$$d^\top Bd \geq \frac{3}{4}\|d\|^2, \tag{42}$$

for any d.

Now, let $\hat{d}$ be an *unconstrained* minimizer of $\phi_x(d)$. Clearly, $\nabla\phi_x(\hat{d}) = 0$. Furthermore, since ϕ_x is a nonnegative function, it follows that for any d,

$$\begin{aligned} \phi_x(d) &= \phi_x(\hat{d}) + \nabla\phi_x(\hat{d})^\top (d - \hat{d}) + \tfrac{1}{2}(d - \hat{d})^\top B(d - \hat{d}) \\ &\geq \tfrac{1}{2}(d - \hat{d})^\top B(d - \hat{d}) \\ &\geq \tfrac{3}{8}\left\|d - \hat{d}\right\|^2 \quad \text{by (42)}. \end{aligned}$$

Since $\phi_x(0) = \theta(x)$, it follows that $\left\|\hat{d}\right\|^2 \leq 8\theta(x)/3$. By hypothesis and the above, $\left\|\bar{d} - \hat{d}\right\|^2 \leq 8\theta(x)/3$. Thus,

$$\begin{aligned} \left\|\bar{d}\right\| &= \left\|\bar{d} - \hat{d} + \hat{d}\right\| \\ &\leq \left\|\bar{d} - \hat{d}\right\| + \left\|\hat{d}\right\| \\ &\leq 2\sqrt{8\theta(x)/3}. \end{aligned}$$

Thus, $\left\|\bar{d}\right\|^2 \leq 32\,\theta(x)/3 < 11\,\theta(x)$. $\square$

Lemma 2.3.4 *Suppose f is Lipschitz continuous. There exist constants $\bar{\gamma} > 0$, and $\bar{\lambda} \geq 0$, such that for any $\lambda \geq \bar{\lambda}$, the modified NE/SQP algorithm applied to MCP($f^{\lambda,\bar{x}}$) will not terminate in Step 2 for any $\gamma \leq \bar{\gamma}$ and $\bar{x} \in$ B.*

Proof Suppose the lemma is false. Then there must exist a sequence $\{\lambda_j, \gamma_j\}$, with $\lambda \to \infty$ and $\gamma \downarrow 0$ such that for each j there exists a perturbed problem MCP($f^{\lambda_j,\bar{x}^j}$, B) where the modified NE/SQP algorithm with $\gamma := \gamma_j$ fails in Step 2 when run on MCP($f^{\lambda_j,\bar{x}^j}$, B)}.

Define $f^j(x)$, $H^j(x)$, $\theta_j(x)$, and $\phi_j(x,d)$, to be the f, H, θ, and ϕ functions corresponding to the jth perturbed problem. For example $f^j(x) := f^{\lambda_j,\bar{x}^j}(x)$, etc. Then for the jth problem to fail in Step 2, there must exist a point x^j and a direction d^j such that d^j is an optimal solution to the quadratic program (QP$_j$) defined by

$$\begin{array}{ll} \min & \phi_j(x^j, d) \\ \text{subject to} & x^j + d \in \mathrm{B}, \end{array}$$

and also d^j fails one of the two tests in Step 2 of the algorithm. Without loss of generality, we can assume $\lambda_j \geq 2L + 2, \forall j$. By Lemma 2.3.3, $\|d^j\|^2 < 11\,\theta_j(x^j) \leq \Delta\theta_j(x^j)$. Thus, the failure must occur because of the first test in Step 2. In other words,

$$\phi_j(x^j, d^j) \geq (1 - \gamma_j)\theta_j(x^j), \quad \forall j. \tag{43}$$

Since $\phi_j(x^j, d^j) \leq \phi_j(x^j, 0) = \theta_j(x^j)$, and also, $\gamma_j \downarrow 0$, we see that

$$\lim \frac{\phi_j(x^j, d^j)}{\theta_j(x^j)} = 1. \tag{44}$$

Let $I_j := I_f^{\lambda_j,\bar{x}^j}(x^j)$, J_j be the set of indices not in I_j,

$$A_j := \frac{\left\|H^j_{I_j}(x^j)\right\|}{\|H^j(x^j)\|} \quad \text{and} \quad B_j := \frac{\left\|H^j_{J_j}(x^j)\right\|}{\|H^j(x^j)\|}.$$

We first show that $\lim_{j\to\infty} A_j = 0$. To do this, we examine a particular choice of j. Let $H^j := H^j(x^j)$. We can then rewrite $\phi_j(x^j, d)$, as follows:

$$\phi_j(x^j, d) := \frac{1}{2}\left\|(M^j + D^j)d + H^j\right\|^2$$

where

$$M^j_{i,\cdot} := \begin{cases} \nabla f_i^j(x^j)^\top & \text{if } i \in I_j \\ 0 & \text{if } i \in J_j. \end{cases} \qquad D^j_{ii} := \begin{cases} \lambda & \text{if } i \in I_j \\ 1 & \text{if } i \in J_j. \end{cases}$$

Observe that $x_i^j - u_i \leq H_i^j \leq x_i^j - l_i$. Note that for $\tilde{d}$ defined by

$$\tilde{d}_i := \begin{cases} -H_i^j/\lambda & \text{if } i \in I_j \\ 0 & \text{if } i \in J_j \end{cases}$$

it follows that $x^j + \tilde{d} \in \mathrm{B}$, since $\lambda \geq 1$. Furthermore

$$(M^j + D^j)\tilde{d} + H^j = \begin{bmatrix} (M^j\tilde{d})_{I_j} \\ H^j_{J_j} \end{bmatrix}.$$

Now, since d^j is an optimal solution to (QP_j),

$$\begin{aligned} \phi_j(x^j, d^j) \leq \phi_j(x^j, \tilde{d}) &= \tfrac{1}{2}\left\|(M^j + D^j)\tilde{d} + H^j\right\|^2 \\ &= \tfrac{1}{2}\left(\left\|(M^j\tilde{d})_{I_j}\right\|^2 + \left\|H^j_{J_j}\right\|^2\right) \\ &\leq \tfrac{1}{2}\left(L^2\left\|\tilde{d}\right\|^2 + \left\|H^j_{J_j}\right\|^2\right) \\ &\leq \tfrac{1}{2}\left\|H^j\right\|^2\left(L^2 {A_j}^2/{\lambda_j}^2 + {B_j}^2\right). \end{aligned}$$

Thus, by (44),

$$1 = \lim \frac{\phi_j(x^j, d^j)}{\theta_j(x^j)} \leq \liminf \left(\frac{L^2 A_j^2}{\lambda_j^2} + B_j^2\right).$$

But, since $\{A_j\}$ is bounded, and $\lambda_j \to \infty$, we see that $1 \leq \liminf B_j^2$. Furthermore, $B_j \leq 1$, so $\lim B_j = 1$, which implies that $A_j \to 0$.

Let us now examine the direction finding subproblem (QP_j) for large j. For some $\alpha \in [0, 1]$, define $\tilde{d}$ by

$$\tilde{d}_i := \begin{cases} 0 & \text{if } i \in I_j \\ -\alpha H_i^j & \text{if } i \in J_j. \end{cases}$$

Here we see that

$$(M^j + D^j)\tilde{d} + H^j = \begin{bmatrix} H^j_{I_j} + (M^j\tilde{d})_{I_j} \\ (1-\alpha)H^j_{J_j} \end{bmatrix}.$$

Thus,

$$\begin{aligned} \phi_j(x^j, d^j) &\leq \frac{1}{2}\left\|(M^j + D^j)\tilde{d} + H^j\right\|^2 \\ &= \frac{1}{2}\left(\left\|H^j_{I_j} + (M^j\tilde{d})_{I_j}\right\|^2 + (1-\alpha)^2\left\|H^j_{J_j}\right\|^2\right) \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{2}\left(\left\|H^j_{I_j}\right\|^2 + 2\left\|H^j_{I_j}\right\|\left\|M^j\tilde{d}\right\| + \left\|M^j\tilde{d}\right\|^2 + (1-\alpha)^2\left\|H^j_{J_j}\right\|^2\right) \\
&\leq \frac{1}{2}\left((A_j\left\|H^j\right\|)^2 + 2A_j\left\|H^j\right\|L\left\|\tilde{d}\right\| + L^2\left\|\tilde{d}\right\|^2 + \left((1-\alpha)B_j\left\|H^j\right\|\right)^2\right) \\
&\leq \frac{1}{2}\left\|H^j\right\|^2\left(A_j^2 + 2A_j\alpha L + \alpha^2L^2 + (1-\alpha)^2B_j^2\right) \\
&\leq \theta(x^j)\left(A_j^2 + 2A_j\alpha L + (1 - 2\alpha + (L^2+1)\alpha^2)\right), \quad \text{since } B_j \leq 1.
\end{aligned}
$$

Choosing $\alpha = 1/(1+L^2)$, we get

$$\phi_j(x^j, d^j) \leq \theta(x^j)\left(A_j(2L/(1+L^2) + A_j) + 1 - 1/(1+L^2)\right).$$

But since $\lim A_j = 0$, we see that

$$\limsup \frac{\phi_j(x^j, d^j)}{\theta(x^j)} \leq 1 - \frac{1}{1+L^2} < 1,$$

contradicting (44). Thus, the lemma is proved by contradiction. □

We can now combine the results of the previous three lemmas to prove that Step 4 always generates an improved starting point.

Lemma 2.3.5 *Suppose that f is Lipschitz continuous and continuously differentiable on $\boldsymbol{B}$ and that MCP(f, $\boldsymbol{B}$) satisfies Assumption 2.1.2. If the QPCOMP algorithm fails to terminate, it will execute Step 2 an infinite number of times.*

Proof Assume the lemma is false. It then follows that after a finite number of statements are executed, the algorithm never returns to Step 2. But this means that, thereafter, the test in Step 4c of the algorithm is never satisfied.

By Theorem 2.2.25, the modified NE/SQP algorithm will always terminate in a finite number of steps. Thus, Step 4b of the QPCOMP algorithm will be executed an infinite number of times. But the test in Step 4b can fail only a finite number of times. After that, λ will be large enough and γ will be small enough that by Lemma 2.3.4 the Modified NE/SQP algorithm will always find a solution to the perturbed problems. Thus we see that Step 4c is visited an infinite number of times, and moreover, after a finite number of iterations, the value of λ is fixed. But then Lemma 2.3.2 guarantees that the test in Step 4c will be satisfied after a finite number of iterations. But this contradicts our original assumption, so the lemma is true. □

We are now ready to prove Theorem 2.3.1

Proof (of Theorem 2.3.1)
By Lemma 2.3.5 either the algorithm will terminate with a solution in Step 3, or Step 2 will be executed an infinite number of times. But if Step 2 is executed an infinite number of times, then we have

$$\theta(x^{k+1}) < \mu\theta(x^k)$$

, thus, $\theta(x^k) < \mu^k\theta(x^0)$, so $\theta(x^k)$ converges to zero. □

2.4 Implementation and Testing

We implemented the QPCOMP algorithm in ANSI C, using double precision arithmetic. The Fortran package MINOS (Murtagh & Saunders 1983) was used to solve the quadratic subproblems. An interface with the GAMS modeling language (Brooke, Kendrick & Meeraus 1988, Dirkse, Ferris, Preckel & Rutherford 1994) was incorporated so that the solver can be called from GAMS. This allows problems to be easily specified, and also allowed the algorithm to be tested using MCPLIB (Dirkse & Ferris 1995*a*) and GAMSLIB (Brooke et al. 1988). The algorithm allows for a great deal of flexibility in the choice of parameters, which can be specified in an options file. For testing purposes, we used the following choices of parameters in the QPCOMP and Modified NE/SQP algorithms: $\mu = .9$, $\Delta = 1.0e4$, $\rho = .5$, $\sigma = .5$. The sequence $\{\eta_j\}$ used in Step 4 of the QPCOMP algorithm was given by $\eta_{j+1} = 0.999 * \eta_j$, with η_0 set to 1000. This effectively caused the Modified NE/SQP algorithm to perform only one iteration before returning control back to QPCOMP. The parameter λ was updated as follows:

1. In Step 4, λ is set to θ_{best}.

2. In Step 4b, if $\tilde{y}$ fails to solve the perturbed problem, λ is set to $\max(.1, 10\lambda)$; otherwise, it is multiplied by .9.

Finally, the parameter γ is initially chosen to be .01. Thereafter, in Step 4b, it is set to $\min(1/\lambda, \gamma)$. For practical considerations, we also placed a limit on the number of allowable iterations of the linesearch in Step 3 of modified NE/SQP algorithm. This limit is set to 10 when the Modified NE/SQP algorithm is called from Step 2 of QPCOMP, and is increased by 4 whenever the Modified NE/SQP algorithm fails, up to a maximum of 30.

QPCOMP was tested using problems from MCPLIB and GAMSLIB, as well as the example problem given in Section 2.1. Specifically, we tested QPCOMP on every problem with fewer than 110 variables in MCPLIB and GAMSLIB. Larger problems were excluded because our implementation of QPCOMP uses a dense solver for the QP subproblems. Table 2 summarizes the features of the problems in MCPLIB and GAMSLIB. Even though only the small problems are used here, we include descriptions of all the problems since they will be used in Chapter 3

We also tested NE/SQP, on the problems in Table 2. To run NE/SQP, we simply used the QPCOMP algorithm with $\Delta = \infty$ and $\gamma = 0$. A comparison of the performance of the algorithms is given in Table 3. Many of the problems in the library are specified with more than one starting point. The particular starting point used is shown in the second column of the table. For each problem we report the execution time (in seconds) and the number of function and Jacobian evaluations, f and J. To save space, we have omitted from this table any problems that both algorithms solved in less than a second. All of the problems were solved to an accuracy of 10^{-6}. Specifically, for QPCOMP the stopping criteria was $\|H(x)\| \leq 10^{-6}$.

The results of the testing demonstrate the high degree of robustness of the QPCOMP algorithm. We note that although it did not solve the Von Thünen problems, QPCOMP was able to solve these problems to an accuracy of 10^{-4}. Experimentation with the Von Thünen problems suggests that the Jacobian matrix is singular at the solution. Thus, near the solution, the Jacobian matrix is poorly conditioned. This ill-conditioning is exacerbated in QPCOMP by the fact that the QP subproblems are formulated using the square of the Jacobian matrix, resulting in extremely ill-conditioned QP subproblems. The inability of QPCOMP to achieve higher-accuracy on these problems appears to be a symptom of this difficulty.

Table 2: Models

GAMS file	Model origin	Type	Size	Nonzeros
bert_oc.gms	Optimal control	MCP	5000	21991
bertsekas.gms	Traffic assignment	NCP	15	74
bratu.gms	MCPLIB	NLP	5625	33750
billups.gms	Section 2	NCP	1	1
cafemge.gms	GAMSLIB (139)	MCP	101	900
cammcp.gms	"	MCP	242	1622
cammge.gms	"	MPSGE	128	1228
choi.gms	Nash equil.	NCP	13	169
cirimge.gms	GAMSLIB	MCP	9	34
co2mge.gms	"	MCP	208	1464
colvncp.gms	Colville #2	NLP	15	99
colvdual.gms	Colville #2 (Dual)	NLP	20	168
dmcmge.gms	GAMSLIB	MCP	170	1595
ehl_k60.gms	Lubrication	MCP	61	3721
ehl_k80.gms	"	MCP	81	6561
ehl_kost.gms	"	MCP	101	10201
ers82mcp.gms	GAMSLIB	MCP	232	1553
etamge.gms	"	MCP	114	849
finmge.gms	"	MCP	153	1916
freebert.gms	Traffic assignment	MCP	15	74
gafni.gms	"	MCP	5	25
gemmcp.gms	GAMSLIB	MCP	262	2794
gemmge.gms	"	MCP	178	3442
hanskoop.gms	Capital stock	NCP	14	129
hansmcp.gms	GAMSLIB (135)	MCP	43	398
hansmge.gms	"(147)	MCP	43	503
harkmcp.gms	"(128)	MCP	32	131
harmge.gms	"(148)	MCP	11	60
hydroc06.gms	Distillation	NE	29	222
hydroc20.gms	"	NE	99	838
josephy.gms	MCPLIB	NCP	4	16
kehomge.gms	GAMSLIB (149)	MCP	9	75
kojshin.gms	MCPLIB	NCP	4	16
kormcp.gms	GAMSLIB (130)	MCP	78	423

Table 2: Models (cont.)

GAMS file	Model origin	Type	Size	Nonzeros
mathi*.gms	Walrasian	NCP	4	14
methan08.gms	Distillation	NE	31	225
mr5mcp.gms	GAMSLIB	MCP	350	1688
nash.gms	Nash equil.	NCP	10	100
nsmge.gms	GAMSLIB	MCP	212	1409
obstacle.gms	MCPLIB	NLP	2500	15000
oligomcp.gms	GAMSLIB (133)	MCP	6	16
opt_cont.gms	MCPLIB	MCP	288	4928
pgvon105.gms	Von Thünen	NCP	105	796
pgvon106.gms	"	NCP	106	898
pies.gms	PIES model	MCP	42	183
powell.gms	Powell	NLP	16	203
powell_mcp.gms	"	NCP	8	54
sammge.gms	GAMSLIB (151)	MCP	23	117
scarfanum.gms	Walrasian	NCP	13	98
scarfasum.gms	"	NCP	14	109
scarfbnum.gms	"	NCP	39	361
scarfbsum.gms	"	NCP	40	614
scarfmge.gms	"	NCP	18	181
shovmge.gms	GAMSLIB (153)	MCP	51	375
sppe.gms	Spatial price	MCP	27	110
tobin.gms	"	MCP	42	243
transmcp.gms	GAMSLIB (126)	MCP	11	34
two3mcp.gms	"(131)	MCP	6	29
unstmge.gms	"(155)	MCP	5	25
vonthmcp.gms	GAMSLIB	MCP	125	761
vonthmge.gms	Von Thünen	MCP	80	594
wallmcp.gms	GAMSLIB (127)	MCP	6	25

Table 3: Performance Results

Problem Name	st. pt.	NE/SQP sec.	NE/SQP f(J)	QPCOMP sec.	QPCOMP f(J)
bertsekas	1	fail	fail	2.83	151(44)
bertsekas	2	fail	fail	2.41	126(40)
billups	1	fail	fail	0.11	23(22)
cafemge	1	18.16	16(10)	20.11	16(10)
cafemge	2	16.57	15(8)	14.19	15(8)
choi	1	2.00	5(4)	2.28	5(4)
colvdual	1	fail	fail	5.76	252(78)
colvdual	2	fail	fail	5.39	184(59)
colvnlp	1	fail	fail	2.13	178(54)
colvnlp	2	fail	fail	1.62	137(30)
ehl_k60	1	16.11	11(8)	16.91	11(8)
ehl_k60	2	fail	fail	147.22	186(84)
ehl_k60	3	fail	fail	492.33	1030(98)
ehl_k80	1	fail	fail	313.15	98(95)
ehl_k80	2	fail	fail	129.02	72(33)
ehl_k80	3	435.77	442	729.89	556(135)
ehl_kost	1	fail	fail	611.41	108(105)
ehl_kost	2	248.79	97(30)	250.28	97(30)
ehl_kost	3	fail	fail	866.08	409(79)
freebert	1	fail	fail	2.72	151(44)
freebert	3	fail	fail	2.86	173(45)
freebert	4	fail	fail	2.47	151(44)
freebert	5	fail	fail	1.38	116(23)
freebert	6	fail	fail	3.02	173(45)
hanskoop	5	fail	fail	0.70	27(11)
hanskoop	7	fail	fail	0.86	45(13)
hansmcp	1	fail	fail	fail	fail
hansmge	1	3.14	11(8)	2.86	11(8)
harkmcp	1	1.27	34(11)	1.06	23(10)
harkmcp	4	6.96	29(13)	9.31	27(14)
harmge	1	fail	fail	1.86	132(57)
harmge	2	fail	fail	0.14	5(4)
harmge	3	fail	fail	0.13	5(4)
harmge	4	fail	fail	0.15	5(4)

Table 3: Performance Results (cont.)

Problem Name	st. pt.	NE/SQP sec.	NE/SQP f(J)	QPCOMP sec.	QPCOMP f(J)
harmge	5	fail	fail	0.16	8(5)
harmge	6	fail	fail	3.24	379(78)
hydroc20	1	16.11	10(8)	13.31	10(8)
josephy	1	fail	fail	0.08	13(7)
josephy	2	fail	fail	0.07	15(7)
josephy	4	fail	fail	0.04	5(4)
kojshin	1	fail	fail	0.07	16(7)
kojshin	3	fail	fail	0.12	35(10)
kormcp	1	2.82	4(3)	2.82	4(3)
pgvon105	1	fail	fail	fail	fail
pgvon105	2	41.51	199(39)	50.91	213(30)
pgvon105	3	33.47	153(32)	58.80	322(40)
pgvon105	4	fail	fail	fail	fail
pgvon106	1	fail	fail	fail	fail
pgvon106	2	fail	fail	fail	fail
pgvon106	3	fail	fail	fail	fail
pgvon106	4	fail	fail	fail	fail
pgvon106	5	fail	fail	fail	fail
pgvon106	6	fail	fail	fail	fail
pies	1	fail	fail	7.26	54(49)
sammge	1	fail	fail	fail	fail
sammge	10	fail	fail	fail	fail
sammge	17	0.62	24(7)	1.05	43(7)
scarfasum	2	fail	fail	1.51	73(26)
scarfbnum	1	6.27	70(20)	6.42	76(21)
scarfbnum	2	6.01	97(22)	6.09	58(19)
scarfbsum	1	fail	fail	8.77	26(22)
scarfbsum	2	fail	fail	31.11	157(83)
shovmge	4	1.19	10(4)	1.96	20(4)
tobin	1	1.33	15(10)	1.49	15(10)
tobin	2	1.83	18(11)	1.78	18(11)
transmcp	1	fail	fail	1.22	69(67)
transmcp	2	fail	fail	fail	fail
vonthmge	1	fail	fail	fail	fail

Chapter 3

Computational Experience with the Proximal Perturbation Strategy

In the preceding chapter, we presented a proximal perturbation strategy and used it to improve the robustness of the NE/SQP algorithm. Unfortunately, the resulting QPCOMP algorithm is relatively slow compared to SMOOTH and PATH. This speed differential is largely explained by the fact that the Gauss-Newton-based method of finding search directions, which is used by NE/SQP and QPCOMP, is more expensive than the Newton-based method used by SMOOTH and PATH. The additional cost of the Gauss-Newton-based method was well-justified in the NE/SQP algorithm since it resulted in a significant theoretical improvement in robustness. However, in the context of the proximal perturbation strategy, the advantages of the Gauss-Newton method are not clear. Note that by using the proximal perturbation strategy, it is no longer fatal if a Newton-based direction finding subproblem is unsolvable; we simply perturb the problem and try again. It is therefore reasonable to expect that using the proximal perturbation strategy on a Newton-based method will be as robust as using it on a Gauss-Newton-based method.

In this chapter we present two new Newton-based algorithms that use the proximal perturbation strategy. The first algorithm, called PROXI, is very similar to QPCOMP in that it searches for a zero of the generalized minimum map given by

$$H_i(x) := \min(x_i - l_i, \max(x_i - u_i, f_i(x))); \tag{1}$$

however, instead of computing the Gauss-Newton direction at each iteration, it computes a Newton direction by solving a single linear system.

The second algorithm, called SEMICOMP, is based upon the semismooth equations approach of De Luca et al. (1995). Here the Φ function defined by (8) in the introduction is generalized to the MCP framework, and solved using a variant of Newton's method, which is based upon the theory of semismooth equations. The resulting algorithm is shown to be Q-quadratically convergent under standard assumptions. We then enhance the robustness of this algorithm by applying the proximal perturbation strategy.

The emphasis in this chapter is on computational experimentation; we provide extensive computational results comparing the performance of PROXI and SEMICOMP with QPCOMP, PATH, and SMOOTH. Further theoretical development of the new algorithms remains as a subject for future research.

3.1 PROXI

The algorithm PROXI is very closely related to QPCOMP. The major difference is that instead of solving a quadratic program to determine the search direction, PROXI solves a linear system to compute the Newton point and then performs a projected linesearch to determine the next iterate. In what follows, we first describe the basic (i.e., unperturbed) algorithm, which we call NE/NEWT. We then apply the proximal perturbation strategy to the NE/NEWT algorithm to produce PROXI.

3.1.1 NE/NEWT Algorithm

To describe the basic algorithm, we shall use several functions that were defined in Chapter 2. The first is the generalization of the minimum map given by (1). As we showed in Chapter 2, finding a zero of H is equivalent to finding a solution to MCP(f, B). Thus, using the H function, the MCP is reformulated as a zero finding problem. The natural merit function for this reformulation is given by

$$\theta(x) := \frac{1}{2}\left\|H(x)\right\|^2. \tag{2}$$

We shall also need to use the index sets

$$\begin{aligned}
I_l(x) &:= \{i : x_i - l_i < f_i(x)\} \\
I_{el}(x) &:= \{i : x_i - l_i = f_i(x)\} \\
I_f(x) &:= \{i : x_i - u_i < f_i(x) < x_i - l_i\} \\
I_{eu}(x) &:= \{i : x_i - u_i = f_i(x)\} \\
I_u(x) &:= \{i : x_i - u_i > f_i(x)\} \\
J_l(x) &:= I_l(x) \bigcup I_{el}(x) \\
J_u(x) &:= I_u(x) \bigcup I_{eu}(x).
\end{aligned}$$

Using these index sets, a linearization of the H function is given by the function $h : \mathrm{B} \times \mathrm{R}^n \to \mathrm{R}^n$ defined by

$$(3) \quad h_i(x,d) := \begin{cases} (x_i - l_i + d_i) & i \in I_l(x) \bigcup I_{el}(x) \\ (x_i - u_i + d_i) & i \in I_u(x) \bigcup I_{eu}(x) \\ (f_i(x) + \nabla f_i(x)^\top d) & i \in I_f(x) \end{cases} \qquad i = 1, \ldots, n.$$

Recall that in NE/SQP and QPCOMP, the search direction was calculated by computing the constrained minimum of the function $\phi_x : \mathrm{R}^n \to \mathrm{R}$ defined by

$$\phi_x(d) := \|h(x,d)\|^2 / 2.$$

In NE/NEWT and PROXI, we shall instead compute the direction d^k simply by solving the linear system $h^k(d) := h(x^k, d) = 0$. One potential difficulty of this approach is that the Newton point $x^k + d^k$ may not be in B. Thus, $f(x^k + d^k)$ may not be defined. To circumvent this difficulty, a projected linesearch is used, based on the projected gradient method of Calamai & Moré (1987).

The projected gradient method finds a stationary point of a continuously differentiable function $g : \mathrm{R}^n \to \mathrm{R}$ on a box $\mathrm{B} \subset \mathrm{R}^n$. At each iteration, the method calculates a direction $d^k := -\nabla g(x^k)$. Then a projected linesearch is used to determine a new point x^{k+1} that produces "sufficient" descent of g. The idea behind the projected linesearch is to evaluate g at a sequence of projected points $\{x(\alpha_i) := \pi_{\mathbb{B}}(x^k + \alpha_i d^k)\}$, where $\alpha_i := \rho^i$ for some $\rho \in (0,1)$. The linesearch procedure terminates when the following condition is satisfied for a given $\sigma \in (0,1)$:

$$g(x(\alpha_i)) \leq g(x^k) + \sigma \nabla g(x^k)^\top (x(\alpha_i) - x^k).$$

The next iterate is then defined by $x^{k+1} := x(\alpha_i)$.

Calamai & Moré (1987) showed that if g is bounded below on B, g is continuously differentiable, and ∇g is uniformly continuous on B, then the projected gradient method converges to a stationary point of g on B.

In our context, we can use the projected linesearch technique to force descent of the merit function θ. However, two potential difficulties must be addressed. First, the function θ is not continuously differentiable. To deal with this problem, instead of using the gradient of θ at x^k, we use the gradient of $\phi_k(d) := \phi_{x^k}(d)$ evaluated at $d = 0$. This gives us the following condition for terminating the linesearch:

$$\theta(x(\alpha_i)) \leq \theta(x^k) + \sigma \nabla \phi_k(0)^\top (x(\alpha_i) - x^k).$$

Note that if $I_{el}(x^k)$ and $I_{eu}(x^k)$ are empty, then $\nabla \theta(x^k)$ is equal to $\nabla \phi_k(d)$.

A simple calculation reveals that $\nabla\phi_x(0)$ is given by

$$\begin{aligned}\nabla\phi_x(0) &:= \nabla h(x) h(x) \\ &= \sum_{i\in I_f} \nabla f_i(x) f_i(x) + \sum_{i\in J_l} (x_i - l_i) e^{i^\top} + \sum_{i\in J_u} (x_i - u_i) e^{i^\top}.\end{aligned}$$

The second difficulty that must be addressed is that the solution d^k to the problem $h^k(d) = 0$ is not in general the negative gradient of θ at x^k. However, part 2 of Lemma 2.2.5 shows that if $\phi_x(d) < \phi_x(0)$, then d is a descent direction for θ at the point x^k. Since d^k satisfies $\phi_k(d^k) = 0$, it follows that d^k is a descent direction for θ at x^k. If the parameter σ is made suitably small, we can therefore expect that the process will yield a stationary point of θ. A formal proof of this hypothesis is left as future research.

As a final observation, note that if x^k is on or very near the boundary of B, then the projected point $\pi_{\mathbb{B}}(x^k + d^k)$ may be very close to x^k. Thus, even if a full step is taken, the decrease in θ may be very small, or none at all. This situation is analogous to the NE/SQP algorithm converging to an s-irregular point. We therefore add an additional termination test into the NE/NEWT algorithm in order to detect this situation. This test is given by

$$\nabla\phi_k(0)^\top \left(\pi_{\mathbb{B}}(x^k + d^k) - x^k\right) \geq -\gamma\theta(x^k).$$

In words, ϕ represents a quadratic model of θ. If the projection of $x^k + d^k$ does not represent a "sufficient" decrease of this quadratic model, we cannot reasonably hope to get descent of θ. Therefore, we abandon the algorithm.

The complete *unperturbed* algorithm is given in Figure 4.

For our implementation, the stopping rule in Step 4 is given by $\theta(x^{k+1}) < tol$ for some small tolerance $tol > 0$.

3.1.2 PROXI Algorithm

We now apply the proximal perturbation strategy to the NE/NEWT algorithm. The resulting PROXI algorithm is given in Figure 5.

The PROXI algorithm was coded in ANSI C, using double precision arithmetic. As with QPCOMP, an interface with the GAMS modeling language was incorporated. For testing purposes, we used the following choices of parameters: $\mu = .9$, $\rho = .5$, $\sigma = .01$. The sequence $\{\eta_j\}$ used in Step 4 was given by $\eta_{j+1} = 0.999 * \eta_j$, with η_0 set to 1000. This effectively caused the NE/NEWT algorithm to perform only one iteration before returning control back to PROXI. The parameter λ was updated as follows:

Figure 4: Algorithm NE/NEWT

Step 1 [Initialization] Select $\rho, \sigma, \gamma \in (0,1)$, and a starting vector $x^0 \in$ B. Set $k = 0$.

Step 2 [Direction generation] Solve the system

$$h(x^k, d) = 0$$

for d, giving the direction d^k. If this system is unsolvable, or if

$$\nabla \phi_k(0)^\top \left(\pi_{\mathbb{B}}(x^k + d^k) - x^k \right) \geq -\gamma \theta(x^k)$$

terminate the algorithm, returning the point x^k along with a failure message; otherwise, continue.

Step 3 [Steplength determination] Let m_k be the smallest nonnegative integer m such that

$$\theta(\pi_{\mathbb{B}}(x^k + \rho^m d^k)) - \theta(x^k) \leq +\sigma \nabla \phi_k(0)^\top (\pi_{\mathbb{B}}(x^k + \rho^m d^k) - x^k);$$

set $x^{k+1} = x^k + \rho^{m_k} d^k$.

Step 4 [Termination check] If x^{k+1} satisfies a prescribed stopping rule, stop, returning the point x^{k+1}. Otherwise, return to Step 2, with k replaced by $k + 1$.

Figure 5: Algorithm PROXI

Step 1 [Initialization] Given a starting vector $x^0 \in \mathbf{B}$ and a convergence tolerance $\epsilon > 0$, choose $\delta > 0$, $\mu \in (0,1)$, and set $k = 0$.

Step 2 [Attempt NE/SQP] Run the NE/NEWT algorithm with starting point x^k and with $tol = \epsilon$. This generates a point $\tilde{x}$.

Step 3 [Termination check] If $\tilde{x}$ solves MCP$(f, \mathbf{B})$, stop; otherwise continue with step 4.

Step 4 [Generate better starting point] Set $\theta_{best} := \theta(\tilde{x})$, set $y^0 = \tilde{x}$, set $j = 0$, and choose $\lambda > 0$, and choose a positive sequence $\{\eta_j\} \downarrow 0$.

Step 4a Run the NE/NEWT algorithm to solve the perturbed problem MCP$(f^{\lambda,y^j}, \mathbf{B})$ from starting point y^j, with $tol = \eta_j/(1 + \|y^j\|)$. This generates a point $\tilde{y}$.

Step 4b If $\tilde{y}$ fails to solve the perturbed problem to the requested accuracy, set $\lambda \geq \lambda + \delta$ and goto step 4a; otherwise, continue.

Step 4c [Check point] If $\theta(\tilde{y}) \leq \mu\theta_{best}$, set $x^{k+1} = \tilde{y}$ and return to step 2, with k replaced by $k+1$. Otherwise, set $y^{j+1} := \tilde{y}$ and return to step 4a, with j replaced by $j+1$.

1. In Step 4, λ is set to θ_{best}.

2. In Step 4b, if $\tilde{y}$ fails to solve the perturbed problem, λ is set to $\max(.1, 10\lambda)$; otherwise, it is multiplied by .9.

Finally, the parameter γ is initially chosen to be .01. Thereafter, in Step 4b, it is set to $\min(1/\lambda, \gamma)$. For practical considerations, we also placed a limit on the number of allowable iterations of the linesearch in Step 3 of modified NE/SQP algorithm. This limit is set to 10 when the NE/NEWT algorithm is called from Step 2 of PROXI, and is increased by 4 whenever the NE/NEWT algorithm fails, up to a maximum of 30.

PROXI was tested on all of the problems from MCPLIB and GAMSLIB, as well as the example problem given in Section 2.1. Table 4 shows the results of this testing in comparison to the unperturbed algorithm NE/NEWT. To run

NE/NEWT, we simply used the PROXI algorithm with $\gamma = 0$ and no limit on the line search. To save space, we have omitted from the table any problems that both algorithms solved in less than one second.

3.2 SEMICOMP

The second algorithm presented in this chapter is based upon an algorithm developed by De Luca et al. (1995), which uses the function

$$\phi(a, b) = \sqrt{a^2 + b^2} - (a + b). \tag{4}$$

Recall from the introduction that this function has the property that

$$\phi(a, b) = 0 \Longleftrightarrow a \geq 0, b \geq 0, ab = 0. \tag{5}$$

Using this fact, De Luca et al. (1995) reformulated the NCP as the zero finding problem $\Phi(x) = 0$, where $\Phi : \mathrm{R}^n \to \mathrm{R}^n$ is defined by

$$\Phi_i(x) := \phi(x_i, f_i(x)),$$

where, by assumption, f is continuously differentiable on all of R^n.

To find a zero of Φ, a generalization of Newton's method was used, which is based upon the theory of semismooth equations. To describe this method, some background definitions are needed.

Definition 3.2.1 (Qi (1993)) *Let $f : \boldsymbol{R}^n \to \boldsymbol{R}^n$ be locally Lipschitzian, and let D be the set where f is differentiable, the* B-subdifferential *of f at x is defined by*

$$\partial_B f(x) := \left\{ H \in \boldsymbol{R}^{n\times n} \;\middle|\; H = \lim_{k\to\infty} f'(x^k) \textit{ for some } \{x^k\} \subset D, \textit{ converging to } x \right\}.$$

The Clarke subdifferential *(Clarke 1983) of f at x is defined by*

$$\partial f(x) := co \partial_B f(x).$$

Definition 3.2.2 *Let $f : \boldsymbol{R}^n \to \boldsymbol{R}^n$ be locally Lipschitzian at $x \in \boldsymbol{R}^n$. We say that f is semismooth at x if*

$$\lim_{\substack{H \in \partial f(x+tv') \\ v' \to v, t \downarrow 0}} Hv' \tag{6}$$

exists for all $v \in \boldsymbol{R}^n$.

Table 4: NE/NEWT vs. PROXI

Problem	st.	NE/NEWT		PROXI	
Name	pt.	sec	f(J)	sec	f(J)
bert_oc	1	3.37	4(3)	2.61	4(3)
bert_oc	2	2.64	4(3)	3.24	4(3)
bert_oc	3	2.72	4(3)	2.78	4(3)
bert_oc	4	2.29	4(3)	2.67	4(3)
bertsekas	1	fail	fail	0.39	138(37)
bertsekas	2	fail	fail	0.27	83(31)
bertsekas	3	fail	fail	0.12	21(20)
billups	1	fail	fail	0.02	23(22)
bratu	1	145.96	48(25)	149.37	48(25)
cammcp	1	fail	fail	2.89	77(23)
choi	1	2.36	5(4)	2.03	5(4)
co2mge	1	fail	fail	0.27	8(2)
co2mge	6	fail	fail	fail	fail
colvdual	1	fail	fail	0.25	201(36)
colvdual	2	fail	fail	0.50	250(55)
colvnlp	1	fail	fail	0.09	77(16)
colvnlp	2	fail	fail	0.05	29(12)
dmcmge	1	fail	fail	fail	fail
dmcmge	2	fail	fail	fail	fail
ehl_k60	1	fail	fail	9.47	60(14)
ehl_k60	2	fail	fail	10.72	60(42)
ehl_k60	3	fail	fail	142.28	1221(144)
ehl_k80	1	fail	fail	8.20	20(13)
ehl_k80	2	fail	fail	29.26	101(45)
ehl_k80	3	fail	fail	51.59	190(74)
ehl_kost	1	fail	fail	18.50	25(14)
ehl_kost	2	fail	fail	37.67	95(28)
ehl_kost	3	fail	fail	64.88	144(44)
finmge	2	fail	fail	11.34	151(25)
finmge	3	1.82	30(7)	0.98	10(4)
finmge	4	fail	fail	12.34	135(28)
finmge	5	fail	fail	2.01	20(6)
freebert	1	fail	fail	0.39	138(37)
freebert	2	fail	fail	0.07	26(8)

Table 4: NE/NEWT vs. PROXI (cont.)

Problem Name	st. pt.	NE/NEWT sec	NE/NEWT f(J)	PROXI sec	PROXI f(J)
freebert	3	fail	fail	0.25	106(35)
freebert	4	fail	fail	0.31	138(37)
freebert	5	fail	fail	0.12	53(14)
freebert	6	fail	fail	0.33	106(35)
gemmge	2	3.75	30(7)	3.31	22(7)
gemmge	3	2.25	6(5)	1.89	6(5)
gemmge	4	2.65	7(6)	2.37	7(6)
gemmge	5	4.95	25(11)	5.00	25(11)
hanskoop	1	fail	fail	0.10	42(16)
hanskoop	2	fail	fail	0.01	2(1)
hanskoop	3	fail	fail	0.09	44(13)
hanskoop	4	fail	fail	0.01	2(1)
hanskoop	5	fail	fail	0.10	68(15)
hanskoop	6	fail	fail	0.01	2(1)
hanskoop	7	fail	fail	0.09	37(15)
hanskoop	8	fail	fail	0.00	2(1)
hanskoop	9	fail	fail	0.24	187(41)
hanskoop	10	fail	fail	0.01	2(1)
hansmcp	1	fail	fail	0.14	18(9)
hansmge	1	fail	fail	0.70	37(15)
harkmcp	1	fail	fail	0.08	39(12)
harkmcp	2	fail	fail	0.07	24(12)
harkmcp	3	fail	fail	0.02	5(4)
harkmcp	4	fail	fail	0.21	23(14)
harmge	1	fail	fail	0.44	222(38)
josephy	1	fail	fail	0.02	37(14)
josephy	3	fail	fail	0.06	155(32)
kehomge	2	fail	fail	0.66	125(26)
kehomge	3	fail	fail	0.13	30(11)
kojshin	1	fail	fail	0.01	18(9)
kojshin	3	fail	fail	0.05	97(22)
mathinum	3	fail	fail	0.01	10(8)
mr5mcp	1	fail	fail	2.17	64(15)
nsmge	1	fail	fail	1.64	35(14)

Table 4: NE/NEWT vs. PROXI (cont)

Problem	st.	NE/NEWT		PROXI	
Name	pt.	sec	f(J)	sec	f(J)
obstacle	1	3.00	11(10)	3.40	11(10)
obstacle	2	5.98	12(11)	7.33	12(11)
obstacle	3	7.23	21(13)	8.85	21(13)
obstacle	4	10.86	23(16)	9.29	23(16)
obstacle	5	5.57	8(6)	4.52	8(6)
obstacle	6	10.51	16(9)	9.92	16(9)
obstacle	7	9.89	17(9)	7.57	17(9)
obstacle	8	8.33	9(6)	7.54	9(6)
opt_cont127	1	8.02	6(5)	9.91	6(5)
opt_cont255	1	19.97	6(5)	18.71	6(5)
opt_cont31	1	1.36	5(4)	1.51	5(4)
opt_cont511	1	42.43	6(5)	43.19	6(5)
pgvon105	1	fail	fail	7.99	403(75)
pgvon105	2	fail	fail	2.18	135(23)
pgvon105	3	fail	fail	52.13	3353(338)
pgvon105	4	fail	fail	fail	fail
pgvon106	1	fail	fail	13.21	739(88)
pgvon106	2	fail	fail	fail	fail
pgvon106	3	fail	fail	fail	fail
pgvon106	4	fail	fail	2.46	86(28)
pgvon106	5	fail	fail	fail	fail
pgvon106	6	fail	fail	fail	fail
pies	1	fail	fail	0.29	73(23)
powell	1	fail	fail	0.10	11(8)
powell	2	fail	fail	0.06	6(5)
powell	3	fail	fail	0.08	8(7)
powell	4	fail	fail	0.25	46(13)
sammge	6	fail	fail	0.27	52(19)
sammge	9	fail	fail	0.45	139(26)
sammge	15	fail	fail	0.48	83(23)
sammge	17	fail	fail	0.57	75(22)

Table 4: NE/NEWT vs. PROXI (cont.)

Problem Name	st. pt.	NE/NEWT sec	NE/NEWT f(J)	PROXI sec	PROXI f(J)
scarfanum	1	fail	fail	0.12	21(8)
scarfanum	2	fail	fail	0.14	25(9)
scarfanum	3	fail	fail	0.16	40(13)
scarfasum	2	fail	fail	0.15	25(9)
scarfasum	3	fail	fail	0.15	34(13)
scarfbnum	1	fail	fail	0.57	164(46)
scarfbnum	2	fail	fail	0.43	165(35)
scarfbsum	1	fail	fail	0.49	60(25)
scarfbsum	2	fail	fail	5.16	1062(117)
scarfmcp	1	fail	fail	0.16	26(11)
scarfmge	1	fail	fail	0.19	28(12)
scarfmge	2	fail	fail	0.13	14(7)
scarfmge	3	fail	fail	0.28	34(12)
scarfmge	4	fail	fail	0.32	39(12)
sppe	1	fail	fail	0.07	25(14)
sppe	2	fail	fail	0.02	9(6)
threemge	7	fail	fail	fail	fail
threemge	8	fail	fail	fail	fail
threemge	11	fail	fail	fail	fail
threemge	12	fail	fail	0.10	26(9)
tobin	1	fail	fail	0.13	27(13)
tobin	2	fail	fail	0.14	43(17)
transmcp	1	fail	fail	0.09	92(26)
transmcp	2	fail	fail	0.00	1(1)
transmcp	3	fail	fail	0.01	3(2)
transmcp	4	fail	fail	0.01	3(2)
vonthmcp	1	fail	fail	fail	fail
vonthmge	1	fail	fail	fail	fail

It is known (Mifflin 1977, Qi & Sun 1993) that convex functions and continuously differentiable functions are semismooth. Semismooth functions also have the following useful properties:

- Sums, products, and composites of semismooth functions are semismooth.
- If a function f is semismooth at x, then f is directionally differentiable at x, and the directional derivative $f'(x;v)$ is equal to the limit (6).

Definition 3.2.3 *A function $f : \boldsymbol{R}^n \to \boldsymbol{R}$ is said to be SC^1 at x if f is continuously differentiable at x and if the gradient of f is semismooth at x. f is said to be SC^1 if for all $x \in \boldsymbol{R}^n$, f is SC^1 at x.*

Qi & Sun (1993) show that C^2 functions are SC^1.

Definition 3.2.4 *Suppose that f is semismooth at x. We say that f is* strongly semismooth *at x if for any sequence $\{d^k\} \subset \boldsymbol{R}^n$ converging to 0, and for $H^k \in \partial f(x+d^k)$,*

$$H^k d^k - f'(x;d^k) = \mathcal{O}(\left\|d^k\right\|^2). \tag{7}$$

Definition 3.2.5 *We say that a semismooth function $f : \boldsymbol{R}^n \to \boldsymbol{R}^n$ is* BD-regular *at x if all the elements of $\partial_B f(x)$ are nonsingular. f is said to be* BD-regular *if for all $x \in \boldsymbol{R}^n$, f is BD-regular at x.*

A generalized Newton method can now be defined as follows:

$$x^{k+1} = x^k + d^k, \quad \text{where} \quad d^k = -(H^k)^{-1} f(x^k), \tag{8}$$

where H^k is a nonsingular element of $\partial_B f(x^k)$.

The utility of this generalized method is established by the following theorem from Qi (1993):

Theorem 3.2.6 *Suppose that x^* is a solution of the system $f(x) = 0$ and that f is semismooth and BD-regular at x^*. Then the iteration method (8) is well-defined and convergent to x^* Q-superlinearly in a neighborhood of x^*. If, in addition, f is directionally differentiable in a neighborhood of x^* and strongly semismooth at x^*, then the convergence rate is Q-quadratic.*

Facchinei & Soares (1994) showed that if f is continuously differentiable on R^n, then Φ is semismooth everywhere. Thus, the iterative method (8) can be applied to find a zero of Φ. In addition, they proved that if f_i is twice continuously differentiable with Lipschitz continuous Hessian, then Φ is strongly semismooth everywhere. Thus, under reasonable assumptions, the method converges Q-quadratically.

A nice feature of this particular reformulation of the NCP is that the natural merit function Ψ associated with Φ, which is defined by

$$\Psi(x) := \frac{1}{2} \|\Phi(x)\|^2, \tag{9}$$

is continuously differentiable (Facchinei & Soares 1994). Thus the problem of finding a solution to NCP(f) is reduced to finding a global minimum of the smooth function Ψ. However, because Φ itself is not smooth, it is still possible to achieve local Q-quadratic convergence to degenerate solutions, which, as we discussed in the introduction, is not possible for algorithms involving smooth reformulations.

3.2.1 Generalization to MCP Framework

To generalize this method to the MCP framework, we propose the function $\Phi : \mathrm{R}^n \to \mathrm{R}^n$ given by

$$\Phi_i(x) := \phi(x_i - l_i, \phi(u_i - x_i, -f_i(x)), \tag{10}$$

where obvious limits are used to define the function when either bound is infinite; thus, if $l_i = -\infty$, then $\Phi_i(x) := -\phi(u_i - x_i, -f_i(x))$, if $u_i = \infty$, then $\Phi_i(x) := \phi(x_i - l_i, f_i(x))$, and if $l_i = -\infty$ and $u_i = \infty$, then $\Phi_i(x) := -f_i(x)$. Observe that if $l = 0$ and $u = \infty$, this function is identical to the Φ function used by De Luca et al. (1995).

It can again easily be shown that finding a zero of this function is equivalent to solving MCP(f, B):

Proposition 3.2.7 *x is a solution to MCP(f, B) if and only if $\Phi(x) = 0$.*

Proof If x solves MCP(f, B), then for each i, one of three cases can occur:

1. $x_i = u_i$ with $f_i(x) \leq 0$;
2. $l_i < x_i < u_i$ with $f_i(x) = 0$; or
3. $x_i = l_i$ with $f_i(x) \geq 0$.

In the first two cases, by (5), $\phi(u_i - x_i, -f_i(x)) = 0$, so again by (5), $\Phi_i(x) = 0$. In the third case, $\phi(u_i - x_i, -f_i(x)) \geq 0$. Thus, since $x_i - l_i = 0$, it follows by (5) that $\Phi_i(x) = 0$.

To prove the converse, observe that if $\Phi_i(x) = 0$, then either $x_i - l_i = 0$ and $\phi(u_i - x_i, -f_i(x)) \geq 0$, or $x_i - l_i > 0$ and $\phi(u_i - x_i, -f_i(x)) = 0$. In the first case, $u_i - x_i = u_i - l_i > 0$. Thus, since $\phi(a, b)$ is negative for $(a, b) > 0$, we see that $f_i(x) \geq 0$. Thus, the complementarity condition $x_i - l_i \geq 0$, $f_i(x) \geq 0$, $(x_i - l_i)f_i(x) = 0$ is satisfied.

In the second case, by (5), the complementarity conditions $u_i - x_i \geq 0$, $f_i(x) \leq 0$, $f_i(x)(u_i - x_i) = 0$ are satisfied. It follows then that x is a solution of MCP(f, B).
□

To use the generalized Newton method (8) to find a zero of Φ, we need to establish that Φ is semismooth.

Theorem 3.2.8 *If f is continuously differentiable on $\mathbf{R}^n$, then the following hold:*

1. *The function Φ defined by (10) is semismooth on $\mathbf{R}^n$.*
2. *If f is twice continuously differentiable with Lipschitz continuous Hessian, then Φ is strongly semismooth everywhere.*
3. *The function Ψ defined by (9) is continuously differentiable, with gradient given by $\nabla\Psi(x) = H^\top \Phi(x)$, where H is any element of $\partial\Phi(x)$.*

Proof To prove Part 1, let $\eta_i : \mathrm{R}^n \to \mathrm{R}^2$ be defined by $\eta_i(x) := (u_i - x_i, -f_i(x))$, and let $\zeta_i : \mathrm{R}^n \to \mathrm{R}$ be defined by $\zeta_i(x) := \phi(\eta_i(x))$. Since ζ_i is the composition of the semismooth functions ϕ and η_i, then ζ_i is semismooth. In similar fashion, if we define $\xi_i : \mathrm{R}^n \to \mathrm{R}^2$ by $\xi_i(x) := (x_i - l_i, \zeta_i(x))$, then we see that $\Phi_i(x) := \phi(\xi_i(x))$, so Φ_i is the composition of semismooth functions and is therefore semismooth. Finally, by Qi & Sun (1993), Φ is semismooth since each of its components is semismooth.

To prove Part 2, we observe that if $\eta_i(x) \neq (0, 0)$, then $\zeta_i(x)$ is twice continuously differentiable with Lipschitz continuous Hessian in a neighborhood of x. Thus, by Qi & Jiang (1994, Lemma 3.1), Φ_i is strongly semismooth at x.

On the other hand, if $\eta_i(x) = (0, 0)$, then $x_i - l_i = u_i - l_i > 0$, so $\xi_i(x) \neq (0, 0)$, and thus, ϕ is continuously differentiable in a neighborhood of $\xi_i(x)$. We then get that

$$\Phi_i'(x + d^k; d^k) = \nabla\phi(\xi_i(x + d^k))^\top \xi_i'(x + d^k; d^k).$$

Now, by Qi & Jiang (1994, Lemma 3.1), η_i is strongly semismooth, so ξ_i is also strongly semismooth. Thus, by (7)

$$\xi_i'(x+d^k;d^k) = \xi_i'(x;d^k) + \mathcal{O}(\left\|d^k\right\|^2).$$

Finally, since $\nabla\phi$ is Lipschitz continuous in a neighborhood of $\xi_i(x)$, and ξ_i is Lipschitz continuous in a neighborhood of x, we see that

$$\nabla\phi(\xi_i(x+d^k)) = \nabla\phi(\xi_i(x)) + \mathcal{O}(\left\|d^k\right\|).$$

Combining the last three equations, we get

$$\begin{aligned}
\Phi_i'(x+d^k;d^k) &= \left(\nabla\phi(\xi_i(x)) + \mathcal{O}(\left\|d^k\right\|)\right)^\top \left(\xi'(x;d^k) + \mathcal{O}(\left\|d^k\right\|^2)\right) \\
&= \nabla\phi(\xi_i(x))^\top \xi'(x;d^k) + \mathcal{O}(\left\|d^k\right\|^2) \\
&= \Phi_i'(x;d^k) + \mathcal{O}(\left\|d^k\right\|^2).
\end{aligned}$$

It then follows by Qi (1993, Lemma 2.3) that Φ_i is strongly semismooth at x. Finally, Φ is strongly semismooth since each component of Φ is strongly semismooth.

The proof to Part 3 is identical to the proof of Facchinei & Soares (1994, Proposition 3.4).

□

The SEMISMOOTH algorithm is given in Figure 6.

In our implementation, SEMISMOOTH terminates in Step 2 if one of the following two conditions is satisfied:

1. $\left\|\Psi(x^k)\right\| \leq tol$ for some tolerance *tol*.
2. $\left\|\nabla\Psi(x^k)\right\| \leq \gamma$ for some tolerance γ.

In the first case, the algorithm is considered to be successful, whereas in the second case, the algorithm is considered to have failed, since the iterates have converged to a stationary point of Ψ, which is not a solution. These stopping rules are similar to the rules used by De Luca et al. (1995).

The main convergence result for this algorithm is given by the following theorem, whose proof is identical to the proof of De Luca et al. (1995, Theorem 3.1), except that we use Theorem 3.2.8 in place of De Luca et al. (1995, Theorem 2.3).

Theorem 3.2.9 *It holds that*

Figure 6: Algorithm SEMISMOOTH

Step 1 [Initialization] Choose $x^0 \in \mathrm{R}^n$, $\rho > 0$, $p > 2$, $\sigma \in (0, 1/2)$, and set $k = 0$.

Step 2 [Termination Check] If x^k satisfies a prescribed stopping rule, stop. Otherwise continue.

Step 3 [Direction generation] Select an element $H^k \in \partial_B \Phi(x^k)$. Find the solution d^k of the system

$$H^k d = -\Phi(x^k). \tag{11}$$

If (11) is not solvable, or if the condition

$$\nabla \Psi(x^k)^\top d^k \le -\rho \left\| d^k \right\|^p \tag{12}$$

is not satisfied, set $d^k = -\nabla \Psi(x^k)$.

Step 4 [Linesearch] Find the smallest $m_k \in \{0, 1, 2, \ldots\}$ such that

$$\Psi(x^k + 2^{-m_k} d^k) \le \Psi(x^k) + \sigma 2^{-m_k} \nabla \Psi(x^k)^\top d^k. \tag{13}$$

Set $x^{k+1} = x^k + 2^{-m_k} d^k$, and go to Step 1, with k replaced by $k + 1$.

1. *Each accumulation point of the sequence $\{x^k\}$ generated by the semismooth algorithm is a stationary point of Ψ.*

2. *If one of the limit points of the sequence $\{x^k\}$, say x^*, is an isolated solution of MCP(f,B), then $\{x^k\} \to x^*$.*

3. *If one of the limit points of the sequence $\{x^k\}$, say x^*, is a BD-regular solution of the system $\Phi(x) = 0$, and if each f_i is twice continuously differentiable with Lipschitz continuous Hessian, then $\{x^k\} \to x^*$, and*

   (a) *Eventually d^k is always given by the solution of (11) (i.e., the negative gradient is never used for k large enough).*

   (b) *Eventually the stepsize of one is always accepted so that $x^{k+1} = x^k + d^k$.*

   (c) *The local convergence rate is Q-quadratic.*

Observe that Step 2 of the algorithm requires choosing an element of $\partial_B \Phi(x^k)$. We now address the question of how to calculate such an element. To do this, we shall need the following lemma, which generalizes Facchinei & Soares (1994, Proposition 3.1).

Lemma 3.2.10

$$\partial \Phi(x)^\top \subset \{D_a(x) + \nabla f(x) D_b(x)\}.$$

Here $D_a(x)$ and $D_b(x)$ are $n \times n$ diagonal matrices whose ith diagonal elements are given by

$$(D_a)_{ii}(x) := a_i(x) + b_i(x) c_i(x), \quad (D_b)_{ii}(x) := b_i(x) d_i(x),$$

where

$$\begin{aligned} a_i(x) &= \frac{x_i - l_i}{\|(x_i - l_i, \phi(u_i - x_i, -f_i(x)))\|} - 1, \\ b_i(x)) &= \frac{\phi(u_i - x_i, -f_i(x))}{\|(x_i - l_i, \phi(u_i - x_i, -f_i(x)))\|} - 1, \end{aligned} \tag{14}$$

if $(x_i - l_i, f_i(x)) \neq (0,0)$, or

$$(a_i(x), b_i(x)) \in \left\{(\xi - 1, \rho - 1) \in \boldsymbol{R}^2 \mid \|(\xi, \rho)\| \leq 1\right\} \tag{15}$$

if $(x_i - l_i, f_i(x)) = 0$; and

$$\begin{aligned} c_i(x) &= \frac{x_i - u_i}{\|(x_i - u_i, f_i(x))\|} + 1, \\ d_i(x)) &= \frac{f_i(x)}{\|(x_i - u_i, f_i(x))\|} + 1 \end{aligned} \tag{16}$$

if $(x_i - u_i, f_i(x)) \neq 0$, *or*

$$(c_i(x), b_i(x)) \in \left\{(\xi+1, \rho+1) \in \boldsymbol{R}^2 \mid \|(\xi,\rho)\| \leq 1\right\} \tag{17}$$

if $(x_i - u_i, f_i(x)) = 0$.

Note that in (14) and (16), if either l_i or u_i is infinite, then the obvious limits are used to define the fractions. Thus, if $l_i = -\infty$, then $(a_i(x), b_i(x)) = (0, -1)$, and if $u_i = \infty$, then $(c_i(x), d_i(x)) = (0, 1)$.

Proof By Clarke (1983, Proposition 2.6.2(e)),

$$\partial\Phi(x)^\top \subset (\partial\Phi_1(x) \times \cdots \times \partial\Phi_n(x)).$$

Thus, it suffices to prove that for each i,

$$\partial\Phi_i(x) \subset \{(a_i(x) + b_i(x)c_i(x))e^{i^\top} + b_i(x)d_i(x)\nabla f_i(x)^\top\}, \tag{18}$$

where $a_i(x), b_i(x), c_i(x), d_i(x)$ satisfy (14)–(17).

To prove this result, let $g_i : \mathrm{R}^n \to \mathrm{R}$ be defined by $g_i(x) := \phi(u_i - x_i, -f_i(x))$, and let $h_i : \mathrm{R}^n \to \mathrm{R}^2$ be defined by $h_i(x) := (x_i - l_i, g_i(x))$. We then have that $\Phi_i(x) = \phi(h_i(x))$. Our first step is to show that $\partial\Phi_i(x) = \partial\phi(h_i(x))\partial h_i(x)$.

We consider two cases. In the first case, suppose that $h_i(x) \neq (0,0)$. It follows that ϕ is continuously differentiable at $h_i(x)$. Furthermore, since f is continuously differentiable, and ϕ is Lipschitz, h_i is locally-Lipschitz at x. Thus, by Clarke (1983, Theorem 2.6.6), $\partial\Phi_i(x) = \partial\phi(h_i(x))\partial h_i(x)$.

In the second case, suppose that $h_i(x) = (0,0)$. It then follows that $u_i - x_i = u_i - l_i > 0$, so ϕ is continuously differentiable at $(u_i - x_i, -f_i(x))$, and therefore h_i is continuously differentiable at x. By the corollary to Clarke (1983, Proposition 2.2.1), h_i is strictly differentiable at x. Furthermore, since ϕ is Lipschitz and convex (Fischer 1992), then by Clarke (1983, Proposition 2.3.6(b)), ϕ is regular everywhere. Thus, by Clarke (1983, Theorem 2.3.9(iii)), $\partial\Phi_i(x) = \partial\phi(h_i(x))\partial h_i(x)$.

We now look at the terms $\partial\phi(h_i(x))$ and $\partial h_i(x)$. It is known (Facchinei & Soares 1994) that

$$\partial\phi(a,b) = \begin{cases} \left\{\left(\dfrac{a}{\|(a,b)\|} - 1, \dfrac{b}{\|(a,b)\|} - 1\right)\right\} & (a,b) \neq 0 \\ \{(\xi - 1, \rho - 1) \mid \|\xi, \rho\| \leq 1\} & (a,b) = 0. \end{cases}$$

Also,

$$\partial h_i(x)^\top = \left\{(e^i, \sigma^i) \,\middle|\, \sigma^i \in \partial g_i(x)\right\}.$$

Thus,

$$\partial\Phi_i(x) = \left\{ a_i(x)e^{i^\top} + b_i(x)\sigma^i \;\middle|\; \sigma^i \in \partial g_i(x), a_i(x), b_i(x) \text{ satisfy (14) and (15)} \right\}.$$

By similar arguments, we get

$$\partial g_i(x) = \left\{ c_i(x)e^{i^\top} + d_i(x)\nabla f_i(x)^\top \mid c_i(x), d_i(x) \text{ satisfy (16)and (17)} \right\}.$$

Combining these last two relations, we see that (18) is satisfied as an equality. □

Figure 7 describes a simple procedure for calculating an element of $\partial_B\Phi(x)$.

Theorem 3.2.11 *The matrix H calculated by the procedure given in Figure 7 is an element of $\partial_B\Phi(x)$.*

Proof In similar fashion to the proof of De Luca et al. (1995, Theorem 7.1), we build a sequence of points $\{y^k\}$ where $\Phi(x)$ is differentiable and such that $\nabla\Phi(y^k)^\top$ tends to H. The theorem then follows by the definition of the B-subdifferential.

Let $y^k := x + \epsilon_k z$, where z is the vector of Step 2 and $\{\epsilon_k\}$ is a sequence of positive numbers converging to 0. For $i \notin \beta_l \bigcup \beta_u$, either $x_i \neq l_i$ and $x_i \neq u_i$, or $f_i(x) \neq 0$; and for $i \in \beta_l \bigcup \beta_u$, $z_i \neq 0$. Thus, if ϵ_k is small enough, either $y_i^k \neq l_i$ and $y_i^k \neq u_i$, or $f_i(y^k) \neq 0$. In either case, Φ is differentiable at y^k.

We now show that for each i, $\lim_{k\to\infty} \nabla\Phi_i(y^k)^\top = H_i(x)$. If either l_i or u_i is infinite, the result is given by De Luca et al. (1995, Theorem 7.1) by a simple change of variables. Thus, without loss of generality, we assume that l_i and u_i are both finite.

By Lemma 3.2.10, $\nabla\Phi_i(y^k)$ is given by

$$(a_i(y^k) + b_i(y^k)c_i(y^k))e^i + b_i(y^k)d_i(y^k)\nabla f_i(y^k)$$

where a_i, b_i, c_i, d_i are defined by (14) and (16).

We now consider three cases.

Case 1: $i \notin \beta_l \bigcup \beta_u$: In this case, by continuity, $\lim_{k\to\infty} \nabla\Phi_i(y^k)^\top = H_i$.

Case 2: $i \in \beta_u$: In this case, $x_i = u_i$, so $y_i^k - u_i = \epsilon_k z_i$, so

$$\begin{aligned} c_i(y^k) &= \frac{\epsilon_k z}{\|(\epsilon_k z, f_i(y^k))\|} + 1 \\ d_i(y^k) &= \frac{f_i(y^k)}{\|(\epsilon_k z, f_i(y^k))\|} + 1 \end{aligned} \qquad (19)$$

Figure 7: Procedure to evaluate an element of $\partial_B \Phi(x)$

Step 1 Set $\beta_l := \{i \mid x_i - l_i = 0 = f_i(x)\}$ and $\beta_u := \{i \mid u_i - x_i = 0 = f_i(x)\}$

Step 2 Choose $z \in \mathrm{R}^n$ such that $z_i \neq 0$ for all $i \in \beta_l \bigcup \beta_u$.

Step 3 For each i, if $i \notin \beta_u$, set

$$\begin{aligned} c_i(x) &:= \frac{x_i - u_i}{\|(x_i - u_i, f_i(x))\|} + 1 \\ d_i(x) &:= \frac{f_i(x)}{\|(x_i - u_i, f_i(x))\|} + 1; \end{aligned}$$

else if $i \in \beta_u$, set

$$\begin{aligned} c_i(x) &:= \frac{z_i}{\|(z_i, \nabla f_i(x)^\top z)\|} + 1 \\ d_i(x) &:= \frac{\nabla f_i(x)^\top z}{\|(z_i, \nabla f_i(x)^\top z)\|} + 1. \end{aligned}$$

Step 4 For each i, if $i \notin \beta_l$, set

$$\begin{aligned} a_i(x) &:= \frac{x_i - l_i}{\|(x_i - l_i, \phi(u_i - x_i, -f_i(x))\|} - 1 \\ b_i(x) &:= \frac{\phi(u_i - x_i, -f_i(x))}{\|(x_i - l_i, \phi(u_i - x_i, -f_i(x))\|} - 1; \end{aligned}$$

else if $i \in \beta_l$, set

$$\begin{aligned} a_i(x) &:= \frac{z_i}{\|(z_i, c_i(x) z_i + d_i(x) \nabla f_i(x)^\top z)\|} - 1 \\ b_i(x) &:= \frac{c_i(x) z_i + d_i(x) \nabla f_i(x)^\top z)}{\|(z_i, c_i(x) z_i + d_i(x) \nabla f_i(x)^\top z)\|} - 1. \end{aligned}$$

Step 5 For each i, set

$$H_i := (a_i(x) + b_i(x) c_i(x)) e^{i\top} + b_i(x) d_i(x) \nabla f_i(x)^\top.$$

Since f is continuously differentiable, we can use a Taylor series expansion to get

$$f_i(y^k) = f_i(x) + \epsilon_k \nabla f_i(\zeta^k)^\top z \quad \text{with } \zeta^k \in [x, y^k].$$

Substituting this expression into (19), we see that

$$\begin{aligned} \lim_{k\to\infty} c_i(y^k) &= \frac{z_i}{\|(z_i, \nabla f_i(x)^\top z)\|} + 1 \\ \lim_{k\to\infty} d_i(y^k) &= \frac{\nabla f_i(x)^\top z}{\|(z_i, \nabla f_i(x)^\top z)\|} + 1 \end{aligned}$$

Thus, $\lim_{k\to\infty} \nabla\Phi_i(y^k)^\top = H_i$.

Case 3: $i \in \beta_l$**:** In this case, $x_i = l_i$ and $f_i(x) = 0$. Clearly, $x_i \neq u_i$, so ϕ is continuously differentiable in a neighborhood of $(u_i - x_i, -f_i(x))$. Thus, using an argument similar to the above we get

$$\lim_{k\to\infty} a_i(y^k) = \frac{z_i}{\|(z_i, \nabla\phi(u_i - x_i, -f_i(x))^\top z)\|} - 1 \tag{20}$$

$$\lim_{k\to\infty} b_i(y^k) = \frac{\nabla\phi(u_i - x_i, -f_i(x))^\top z)}{\|(z_i, \nabla\phi(u_i - x_i, -f_i(x))^\top z)\|} - 1 \tag{21}$$

Finally, $\nabla\phi(u_i - x_i, -f_i(x)) = c_i(x)e^i + d_i(x)\nabla f_i(x)$, where $c_i(x)$ and $d_i(x)$ are given by (16). Substituting this expression into (20) and (21), we see that $\lim_{k\to\infty} \nabla\Phi_i(y^k)^\top = H_i$.

□

3.2.2 SEMICOMP Algorithm

We now use the proximal perturbation strategy to improve the robustness of the SEMISMOOTH algorithm. The resulting SEMICOMP algorithm is given in Figure 8.

The SEMICOMP algorithm was coded in ANSI C, using double precision arithmetic. As with QPCOMP and PROXI, an interface with the GAMS modeling language was incorporated. For testing purposes, we used the same choices of parameters that were used by the PROXI algorithm. SEMICOMP was tested on all of the problems from MCPLIB and GAMSLIB, as well as the example problem given in Section 2.1. Table 5 shows the results of this testing in comparison to the unperturbed algorithm SEMISMOOTH. To run SEMISMOOTH, we simply used the SEMICOMP algorithm with no limit on the line search. To save space, we have omitted from the table any problems that both algorithms solved in less than one second.

Figure 8: Algorithm SEMICOMP

Step 1 [Initialization] Given a starting vector $x^0 \in \mathbf{B}$ and a convergence tolerance $\epsilon > 0$, choose $\delta > 0$, $\mu \in (0, 1)$, and set $k = 0$.

Step 2 [Attempt SEMISMOOTH Algorithm] Run the SEMISMOOTH algorithm with starting point x^k, with $tol = \epsilon$. This generates a point $\tilde{x}$.

Step 3 [Termination check] If $\tilde{x}$ solves MCP(f, B), stop; otherwise continue with step 4.

Step 4 [Generate better starting point] Set $\theta_{best} := \theta(\tilde{x})$, set $y^0 = \tilde{x}$, set $j = 0$, and choose $\lambda > 0$, and choose a positive sequence $\{\eta_j\} \downarrow 0$.

- Step 4a Run the SEMISMOOTH algorithm to solve the perturbed problem MCP$(f^{\lambda,y^j}, \mathrm{B})$ from starting point y^j, with $tol = \eta_j/(1 + \|y^j\|)$. This generates a point $\tilde{y}$.
- Step 4b If $\tilde{y}$ fails to solve the perturbed problem to the requested accuracy, set $\lambda \geq \lambda + \delta$ and goto step 4a; otherwise, continue.
- Step 4c [Check point] If $\theta(\tilde{y}) \leq \mu\theta_{best}$, set $x^{k+1} = \tilde{y}$ and return to step 2, with k replaced by $k + 1$. Otherwise, set $y^{j+1} := \tilde{y}$ and return to step 4a, with j replaced by $j + 1$.

Table 5: SEMISMOOTH vs. SEMICOMP

Problem Name	st. pt.	SEMISMOOTH sec	SEMISMOOTH f(J)	SEMICOMP sec	SEMICOMP f(J)
bert_oc	1	13.50	21(11)	11.38	21(11)
bert_oc	2	54.41	143(42)	46.44	143(42)
bert_oc	3	17.99	41(15)	15.52	41(15)
bert_oc	4	5.91	7(6)	5.80	7(6)
bertsekas	1	fail	fail	0.64	251(42)
bertsekas	2	fail	fail	0.59	327(38)
billups	1	0.90	6903(345)	0.10	631(76)
bratu	1	fail	fail	7452.38	3164(538)
cammcp	1	fail	fail	fail	fail
choi	1	2.93	6(5)	2.95	6(5)
co2mge	2	2.42	63(16)	2.02	62(15)
co2mge	6	fail	fail	fail	fail
colvdual	2	fail	fail	fail	fail
dmcmge	1	fail	fail	fail	fail
dmcmge	2	fail	fail	133.73	3099(661)
ehl_k60	1	8.21	43(18)	8.25	43(18)
ehl_k60	2	fail	fail	55.26	488(78)
ehl_k60	3	fail	fail	fail	fail
ehl_k80	1	10.38	37(16)	11.29	37(16)
ehl_k80	2	29.01	121(32)	29.74	121(32)
ehl_k80	3	123.01	568(100)	126.13	568(100)
ehl_kost	1	15.02	32(15)	18.99	32(15)
ehl_kost	2	58.25	125(34)	49.06	125(34)
ehl_kost	3	240.12	671(114)	233.23	671(114)
etamge	1	1.34	30(17)	1.27	30(17)
finmge	2	fail	fail	fail	fail
finmge	3	1.58	8(5)	1.65	8(5)
finmge	4	fail	fail	fail	fail
finmge	5	1.36	9(6)	1.73	9(6)
freebert	1	fail	fail	0.51	266(46)
freebert	3	fail	fail	0.55	206(42)
freebert	4	fail	fail	0.60	240(42)
freebert	6	fail	fail	0.53	200(40)

Table 5: SEMISMOOTH vs. SEMICOMP (cont.)

Problem Name	st. pt.	SEMISMOOTH sec	SEMISMOOTH f(J)	SEMICOMP sec	SEMICOMP f(J)
gemmge	2	3.60	16(9)	3.31	16(9)
gemmge	3	3.92	10(8)	2.88	10(8)
gemmge	4	3.22	8(7)	2.84	8(7)
gemmge	5	6.93	31(13)	5.32	31(13)
hanskoop	1	fail	fail	fail	fail
hanskoop	2	fail	fail	fail	fail
hanskoop	3	fail	fail	fail	fail
hanskoop	4	fail	fail	fail	fail
hanskoop	7	fail	fail	fail	fail
harkmcp	4	fail	fail	fail	fail
harmge	1	fail	fail	1.52	672(75)
harmge	2	fail	fail	0.01	3(2)
mr5mcp	1	2.01	26(13)	2.09	26(13)
nsmge	1	1.65	23(12)	1.69	23(12)
obstacle	1	5.59	15(14)	6.86	15(14)
obstacle	2	15.56	17(14)	18.01	17(14)
obstacle	3	9.45	14(13)	11.77	14(13)
obstacle	4	10.66	17(16)	11.01	17(16)
obstacle	5	14.59	8(7)	15.08	8(7)
obstacle	6	21.14	20(13)	19.62	20(13)
obstacle	7	15.52	17(12)	12.84	17(12)
obstacle	8	14.32	10(7)	14.76	10(7)
opt_cont127	1	45.58	27(12)	46.05	27(12)
opt_cont255	1	110.61	31(14)	107.97	31(14)
opt_cont31	1	4.45	11(9)	5.55	11(9)
opt_cont511	1	360.42	73(20)	348.63	73(20)
pgvon105	1	fail	fail	fail	fail
pgvon105	2	fail	fail	fail	fail
pgvon105	3	fail	fail	fail	fail
pgvon105	4	fail	fail	28.09	352(42)
pgvon106	1	fail	fail	fail	fail
pgvon106	2	fail	fail	fail	fail
pgvon106	3	fail	fail	fail	fail

Table 5: SEMISMOOTH vs. SEMICOMP (cont.)

Problem Name	st. pt.	SEMISMOOTH sec	SEMISMOOTH f(J)	SEMICOMP sec	SEMICOMP f(J)
pgvon106	4	fail	fail	38.30	412(65)
pgvon106	5	fail	fail	fail	fail
pgvon106	6	fail	fail	fail	fail
sammge	2	1.96	645(53)	0.13	38(7)
sammge	3	fail	fail	0.17	41(8)
sammge	5	fail	fail	0.36	84(14)
sammge	6	fail	fail	0.40	103(17)
sammge	7	fail	fail	0.23	54(11)
sammge	8	fail	fail	0.39	114(16)
sammge	9	fail	fail	0.65	168(29)
sammge	11	1.76	577(48)	0.19	62(9)
sammge	13	fail	fail	0.23	61(14)
sammge	14	fail	fail	0.23	74(11)
sammge	15	fail	fail	0.38	122(20)
sammge	16	fail	fail	0.31	80(12)
sammge	17	fail	fail	0.20	61(11)
sammge	18	fail	fail	0.50	148(22)
scarfasum	3	fail	fail	fail	fail
scarfbnum	1	fail	fail	1.01	241(51)
scarfbnum	2	fail	fail	7.36	1497(341)
scarfbsum	2	fail	fail	1.22	276(43)
threemge	11	fail	fail	0.82	215(26)
transmcp	1	fail	fail	0.23	193(105)
transmcp	2	fail	fail	0.00	1(1)
transmcp	3	fail	fail	0.03	15(8)
transmcp	4	fail	fail	0.04	43(10)
vonthmcp	1	fail	fail	fail	fail
vonthmge	1	fail	fail	fail	fail

3.3 Effects of Proximal Perturbation Strategy

The test results given in Tables 3–5 demonstrate the robustness of the proximal perturbation strategy in convincing fashion. In all three of the perturbed algorithms, the perturbation strategy significantly improved the robustness, with no significant loss in efficiency. Indeed, all three algorithms were extremely successful at solving the problems in the model libraries.

It is interesting to note that the robustness of the underlying algorithms plays little role in the performance of the perturbed algorithms. Indeed PROXI had the fewest failures of all the algorithms tested even though the underlying algorithm, NE/NEWT, is the least robust algorithm tested. In contrast, the least successful of the perturbed algorithms, SEMICOMP, is based on the most robust of the underlying algorithms, SEMISMOOTH. This observation suggests that a promising approach for improving the capabilities of complementarity solvers on very large scale problems may be to use a simple (but possibly not very robust) basic algorithm and then enhance its robustness by using the proximal perturbation strategy. It may therefore be worthwhile to reexamine some algorithmic approaches that have been abandoned due to deficiencies in robustness.

3.4 Comparison with PATH and SMOOTH

In this section we compare the performance of the algorithms described in this thesis, namely PROXI, SEMICOMP and QPCOMP with PATH version 28, and SMOOTH version 3. The PATH algorithm is based upon the path search scheme of Ralph (1994), but includes several enhancements that greatly improve its performance. In particular, PATH employs a projected Newton preprocessor as a tool for rapidly identifying the active set. PATH also uses a nonmonotone linesearch as well as the watchdog technique described by Chamberlain, Powell & Lemaréchal (1982). A detailed description of PATH is given by Dirkse (1994).

The SMOOTH algorithm is based upon the smoothing technique of Chen & Mangasarian (1995*b*) and Chen (1995), which replaces the minimum map with a sequence of smooth approximations. SMOOTH version 3 employs the same projected Newton preprocessor that was developed for PATH version 27. However, the use of this preprocessor is slightly different. In SMOOTH, the preprocessor is used actually to solve the problem. Thus, the smoothing technique is used only if the problem is not solved by the preprocessor. In contrast, PATH uses the preprocessor only to identify the active set.

We tested the five algorithms on all of the problems included in the GAMSLIB and MCPLIB problem libraries, with the exception of QPCOMP, which was tested only on problems with fewer than 110 variables. In our testing, the algorithms were run using their default options on all problems. Execution times for these tests are reported in Table 6. To save space we have omitted any problems that all the algorithms solved in less than one second.

From these results it is clear that PROXI and SEMICOMP are comparable to SMOOTH and PATH in terms of efficiency and robustness. When only smaller problems are considered, QPCOMP is also competitive in terms of robustness, but is considerably slower.

Table 6: Execution Times (sec.)

Problem Name	st. pt.	PATH	PROXI	QPCOMP	SEMICOMP	SMOOTH
bert_oc	1	2.63	2.61	–	11.38	3.23
bert_oc	2	3.13	3.24	–	46.44	2.57
bert_oc	3	2.10	2.78	–	15.52	2.55
bert_oc	4	2.29	2.67	–	5.80	2.62
bertsekas	1	0.08	0.39	2.83	0.64	0.24
bertsekas	2	0.04	0.27	2.41	0.59	0.05
bertsekas	3	0.09	0.12	0.33	0.49	0.21
billups	1	fail	0.02	0.11	0.10	fail
bratu	1	138.52	149.37	–	7452.38	135.48
cafemge	1	0.29	0.50	20.11	0.50	0.41
cafemge	2	0.26	0.35	14.19	0.50	0.25
cammcp	1	0.21	2.89	–	fail	0.23
choi	1	2.09	2.03	2.28	2.95	2.10
cirimge	3	0.42	0.33	–	0.86	1.25
co2mge	1	fail	0.27	–	0.10	0.42
co2mge	2	0.50	0.48	–	2.02	0.52
co2mge	6	0.46	fail	–	fail	1.96
colvdual	1	0.11	0.25	5.76	0.12	0.11
colvdual	2	0.09	0.50	5.39	fail	0.10
colvnlp	1	0.05	0.09	2.13	0.08	0.06
colvnlp	2	0.03	0.05	1.62	0.06	0.05
dmcmge	1	3.75	fail	–	fail	5.42
dmcmge	2	0.55	fail	–	133.73	0.60
ehl_k60	1	1.56	9.47	16.91	8.25	1.59
ehl_k60	1	25.16	10.72	147.22	55.26	14.71
ehl_k60	1	44.97	142.28	492.33	fail	fail
ehl_k80	1	2.37	8.20	313.15	11.29	2.93
ehl_k80	1	131.99	29.26	129.02	29.74	6.57
ehl_k80	1	56.58	51.59	729.89	126.13	85.26
ehl_kost	1	3.86	18.50	611.41	18.99	4.73
ehl_kost	2	13.56	37.67	250.28	49.06	12.58
ehl_kost	3	9.76	64.88	866.08	233.23	90.38
etamge	1	0.49	1.11	–	1.27	0.97
finmge	2	1.95	11.34	–	fail	5.16
finmge	3	0.94	0.98	–	1.65	0.94

Table 6: Execution Times (sec.) (cont.)

Problem Name	st. pt.	PATH	PROXI	QPCOMP	SEMICOMP	SMOOTH
finmge	4	1.72	12.34	–	fail	9.18
finmge	5	0.91	2.01	–	1.73	1.09
freebert	1	0.07	0.39	2.72	0.51	0.04
freebert	2	0.03	0.07	0.61	0.06	0.04
freebert	3	0.05	0.25	2.86	0.55	0.04
freebert	4	0.09	0.31	2.47	0.60	fail
freebert	5	0.04	0.12	1.38	0.15	0.04
freebert	6	0.08	0.33	3.02	0.53	fail
gemmge	2	3.24	3.31	–	3.31	4.18
gemmge	3	1.85	1.89	–	2.88	1.85
gemmge	4	2.51	2.37	–	2.84	1.84
gemmge	5	8.85	5.00	–	5.32	2.28
hanskoop	1	0.05	0.10	0.37	fail	0.33
hanskoop	2	0.06	0.01	0.05	fail	0.02
hanskoop	3	0.11	0.09	0.42	fail	0.23
hanskoop	4	0.05	0.01	0.05	fail	0.02
hanskoop	7	0.05	0.09	0.86	fail	0.22
hansmcp	1	0.47	0.14	fail	0.16	0.13
hansmge	1	0.36	0.70	2.86	0.84	0.64
harkmcp	1	0.05	0.08	1.06	0.07	0.07
harkmcp	4	0.12	0.21	9.31	fail	0.37
harmge	1	0.06	0.44	1.86	1.52	0.09
harmge	6	0.06	fail	3.24	0.02	2.08
hydroc20	1	0.38	0.44	13.31	0.54	0.36
josephy	6	fail	0.02	0.05	0.01	0.02
kormcp	1	0.08	0.06	2.82	0.07	0.05
mr5mcp	1	0.62	2.17	–	2.09	0.62
nsmge	1	0.91	1.64	–	1.69	2.40
obstacle	1	2.36	3.40	–	6.86	2.39
obstacle	2	5.90	7.33	–	18.01	6.39
obstacle	3	5.03	8.85	–	11.77	6.27
obstacle	4	4.84	9.29	–	11.01	6.12
obstacle	5	8.04	4.52	–	15.08	7.13
obstacle	6	8.86	9.92	–	19.62	10.07

Table 6: Execution Times (sec.) (cont.)

Problem Name	st. pt.	PATH	PROXI	QPCOMP	SEMICOMP	SMOOTH
obstacle	7	7.39	7.57	–	12.84	7.97
obstacle	8	13.84	7.54	–	14.76	10.58
opt_cont31	1	1.36	1.51	–	5.55	1.65
opt_cont127	1	8.14	9.91	–	46.05	6.38
opt_cont255	1	14.86	18.71	–	107.97	13.80
opt_cont511	1	39.51	43.19	–	348.63	37.52
pgvon105	1	1.54	7.99	fail	fail	fail
pgvon105	2	0.77	2.18	50.91	fail	fail
pgvon105	3	1.58	52.13	58.80	fail	fail
pgvon105	4	fail	fail	fail	28.09	fail
pgvon106	1	19.77	13.21	fail	fail	125.46
pgvon106	2	1.80	fail	fail	fail	5.37
pgvon106	3	1.29	fail	fail	fail	8.48
pgvon106	4	fail	2.46	fail	38.30	fail
pgvon106	5	fail	fail	fail	fail	fail
pgvon106	6	fail	fail	fail	fail	3.76
pies	1	0.13	0.29	7.26	0.11	0.27
sammge	1	0.01	0.01	fail	0.00	0.00
sammge	10	0.01	0.01	fail	0.01	0.01
sammge	17	0.09	0.57	1.05	0.20	0.17
scarfasum	2	0.04	0.15	1.51	0.15	0.10
scarfasum	3	0.07	0.15	0.37	fail	0.05
scarfbnum	1	0.39	0.57	6.42	1.01	0.32
scarfbnum	2	0.44	0.43	6.09	7.36	0.32
scarfbsum	1	fail	0.49	8.77	0.39	0.24
scarfbsum	2	3.43	5.16	31.11	1.22	0.66
shovmge	2	0.09	0.09	1.11	0.09	0.10
shovmge	4	0.08	0.10	1.96	0.11	0.08
threemge	7	0.06	fail	–	0.14	0.05
threemge	8	0.06	fail	–	0.12	0.05
threemge	11	0.05	fail	–	0.82	0.05
tobin	1	0.08	0.13	1.49	0.11	0.13
tobin	2	0.10	0.14	1.78	0.09	0.09

Table 6: Execution Times (sec.) (cont.)

Problem Name	st. pt.	PATH	PROXI	QPCOMP	SEMICOMP	SMOOTH
transmcp	1	0.04	0.09	1.22	0.23	0.05
transmcp	2	0.01	0.00	fail	0.00	0.00
vonthmcp	1	fail	fail	–	fail	fail
vonthmge	1	1.06	fail	fail	fail	17.14

Chapter 4

An Infeasible Interior-Point Algorithm for LMCP

The purpose of this chapter is to develop an infeasible interior-point algorithm for the linear mixed complementarity problem. As discussed in the introduction, such an algorithm could be extremely useful in the context of solving very large scale monotone nonlinear mixed complementarity problems. In particular, if a successive LCP technique is used to solve the nonlinear MCP, then a fast algorithm that can be easily warm-started is needed for solving the linear subproblems. This exactly describes the infeasible interior-point methods. In particular, they are extremely efficient on very large scale monotone problems; but, unlike other interior-point methods, the infeasible interior-point methods are easily warm-started.

The algorithm we present in this chapter is based upon two algorithms proposed in Zhang (1994) and Wright (1994). Zhang's algorithm solves the horizontal linear complementarity problem (HLCP), and is proven, under certain assumptions, to converge to a solution at a global Q-linear rate. Unfortunately, Zhang's convergence analysis places a restriction on the starting point. This restriction can easily be satisfied by making the starting point large in every component; but this completely defeats the purpose of warm-starting since changing the starting point may take us farther away from the solution. Moreover, this restriction presents difficulties when Zhang's results are used in the analysis of the LMCP algorithm presented later in this chapter. Fortunately, the restriction on the starting point is unnecessary. The proof of this fact is one of the major results of this chapter.

Wright's algorithm is a sophisticated special case of Zhang's algorithm that solves the linear complementarity problem and achieves local Q-subquadratic convergence in addition to the global Q-linear rate proved by Zhang. Unfortunately, Wright's analysis, like Zhang's, places a restriction on the starting point. But again, we shall be able to remove this restriction.

The algorithm we propose for the LMCP is derived directly from Wright's algorithm. The strategy we use exploits the fact that the LMCP can be reformulated as an LCP. However, since this reformulated problem has a great deal of structure, we shall take care to ensure that this structure is exploited to full advantage. With this in mind, the LMCP algorithm is derived as as follows: first, we reformulate the LMCP as an equivalent LCP. Next, we substitute this reformulated LCP directly into Wright's algorithm, producing a naive algorithm for the LMCP whose iterates correspond directly with the iterates of the LCP. Finally, we simplify this algorithm, taking advantage of the structure inherent in the reformulated problem. This produces the LMCP algorithm, which in general, involves fewer variables than the naive algorithm.

To prove global and local Q-subquadratic convergence, we construct a one-to-one function T, which maps the iterates of the LMCP algorithm to the iterates of the naive algorithm. With this mapping, the task of proving convergence results for the LMCP algorithm is reduced to proving convergence results for the naive algorithm. But since the iterates of the naive algorithm are actually iterates of Wright's LCP algorithm, we can establish convergence rates for the LMCP algorithm directly from Wright's convergence results.

But now the restriction on the starting point becomes significant. Recall that this restriction is easily satisfied by making the starting point large in every component. But, this cannot be done in our analysis since we require that the starting point of the LCP lies in the range of T. In general, this places an upper bound on some components of the iterates of the LCP. In other words, it may not be possible to find *any* starting point for the LMCP for which the corresponding starting point for the LCP satisfies Wright's restriction. However, by extending Zhang's and Wright's convergence results to arbitrary positive starting points, this difficulty is removed.

The chapter is organized as follows. We begin by presenting Zhang's algorithm for the HLCP and extending the global convergence results to apply to arbitrary starting points. We then present Wright's algorithm for the LCP and extend the global and local convergence results to apply to arbitrary starting points. Finally, we present the LMCP algorithm and prove global Q-linear convergence and local Q-subquadratic convergence.

4.1 Zhang's HLCP Algorithm

In Zhang (1994), two algorithms are presented for solving the horizontal linear complementarity problem. The first algorithm is a very general algorithm about which a number of useful lemmas can be proved. The second algorithm is a special case of the first for which Zhang proves a global Q-linear convergence result. In this section, we describe both of these algorithms and then extend Zhang's convergence analysis to apply to arbitrary positive starting points.

Because our analysis is intimately connected with Zhang's analysis, it is desirable to be consistent with the notation used in Zhang's paper. We therefore restate the HLCP as follows:

$$\text{HLCP}(M,N,h): \qquad \begin{array}{l} \text{Find } (x,y) \in \mathrm{R}^n \times \mathrm{R}^n \text{ such that} \\ F(x,y) := \begin{bmatrix} Mx + Ny - h \\ XYe \end{bmatrix} = 0, \quad (x,y) \geq 0, \end{array}$$

where $M, N \in \mathrm{R}^{n\times n}, e, h \in \mathrm{R}^n, X = \mathrm{diag}(x), Y = \mathrm{diag}(y), e = (1,1,\cdots,1)^\top$.

For convenience of discussion, Zhang defines the following sets:

$$\begin{aligned} \mathcal{S} &= \{(x,y) \in \mathrm{R}^{2n} : h = Mx + Ny, (x,y) \geq 0, x^\top y = 0\}, \text{i.e., the solution set,} \\ \mathcal{A} &= \{(x,y) \in \mathrm{R}^{2n} : h = Mx + Ny\}, \\ \mathcal{F} &= \{(x,y) \in \mathcal{A} : (x,y) \geq 0\}, \text{i.e., the set of feasible points.} \end{aligned}$$

Zhang's algorithms can be described as centered and damped Newton methods that work as follows: given a starting point $(x^0, y^0) > 0$, both algorithms generate a sequence of strictly positive iterates $\{(x^k, y^k)\}$ that, under appropriate assumptions, converge to a solution (x^*, y^*) of the HLCP.

To prove his results, Zhang makes the following assumptions on the problem:

Assumption 4.1.1 *For any $(x,y) \in \mathcal{A}$ and $(\hat{x},\hat{y}) \in \mathcal{A}, (x-\hat{x})^\top(y-\hat{y}) \geq 0$, i.e. $\mathcal{A}$ is the graph of a monotone operator.*

Assumption 4.1.2 *$\mathcal{F} \neq \emptyset$, i.e., a feasible point exists.*

It is known that Assumptions4.1.1 and 4.1.2 imply the existence of a solution (x^*, y^*) to HLCP(M,N,h) (Güler 1995, Theorem 3.1). It is also well-known that Assumption 4.1.1 is satisfied by linear programs, convex quadratic programs, and monotone linear complementarity problems.

In addition to these two *explicit* assumptions, Zhang also makes an *implicit* assumption about the starting point. Given a point $(u^0, v^0) \in \mathcal{A}$ (such a point exists by Assumption 4.1.2), Zhang proves his convergence results by choosing a starting point $(x^0, y^0) > 0$ that satisfies $(x^0, y^0) \geq (u^0, v^0)$. It is easy to find such an (x^0, y^0) (simply choose $x^0 = \max(\zeta, u^0)$ and $y^0 = \max(\zeta, v^0)$ for some $\zeta > 0$). However, since we are interested in warm-starting the algorithm, we do not want to change the starting point. Thus, given a *fixed* starting point (x^0, y^0), Zhang's results are based on the following implicit assumption:

Assumption 4.1.3 *There exists* $(u^0, v^0) \in \mathcal{A}$ *such that* $(x^0, y^0) \geq (u^0, v^0)$.

We now prove Zhang's results without this implicit assumption. We start with Zhang's first algorithm, which is given in Figure 9.

Figure 9: Zhang's First Algorithm

Given $(x^0, y^0) > 0$, for $k = 0, 1, 2, \ldots$, do

1. Choose $\sigma_k \in [0, 1)$ and let $\mu_k = \frac{1}{n} {x^k}^\top y^k$. Solve the following linear system for $(\Delta x^k, \Delta y^k)$

(1)
$$\begin{bmatrix} M & N \\ Y & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} h - Mx^k - Ny^k \\ -X^k Y^k + \sigma_k \mu_k e \end{bmatrix}.$$

2. Choose a steplength $\alpha_k \in (0, 1]$ and

$$\alpha_k < \hat{\alpha}_k = \frac{-1}{\min((X^k)^{-1}\Delta x^k, (Y^k)^{-1}\Delta y^k, -1/2)}.$$

   Let $x^{k+1} = x^k + \alpha_k \Delta x^k$ and $y^{k+1} = y^k + \alpha_k \Delta y^k$.

We shall prove a number of technical lemmas about the algorithm given in Figure 9, which we will then use to prove global Q-linear convergence of Zhang's second algorithm, which is a special case of Zhang's first algorithm.

We begin as Zhang does by constructing an auxiliary sequence $\{(u^k, v^k)\}$. Given a pair $(u^0, v^0) \in \mathcal{A}$, for $k = 0, 1, \ldots$ we define

(2)
$$u^{k+1} := u^k + \alpha_k(\Delta x^k + x^k - u^k), \quad v^{k+1} := v^k + \alpha_k(\Delta y^k + y^k - v^k),$$

where x^k, y^k, Δx^k and Δy^k are defined in Figure 9. The sequence is strictly a tool for analysis and is not actually computed. The following lemma summarizes some of the properties of the auxiliary sequence.

Lemma 4.1.4 *Let* $\{(x^k, y^k)\}$ *and* $\{\alpha_k\}$ *be generated by the algorithm in Figure 9,* $\{(u^k, v^k)\}$ *be given by (2) and* $\nu_k := \prod_{j=0}^{k-1}(1 - \alpha_j)$. *Then for* $k \geq 0$

1. $(u^k, v^k) \in \mathcal{A}$, *i.e.,* $h = Mu^k + Nv^k$;
2. $x^k - u^k = \nu_k(x^0 - u^0)$ *and* $y^k - v^k = \nu_k(y^0 - v^0)$;

3. $|x^k - u^k| = \nu_k |x^0 - u^0| \le |x^0 - u^0|$ *and* $|y^k - v^k| = \nu_k |y^0 - v^0| \le |y^0 - v^0|$.

Proof Statements 1 and 2 are proven in Zhang (1994, Lemma 4.1). Statement 3 follows immediately from Statement 2 and the fact that $0 \le \nu_k < 1$. □

Lemma 4.1.5 *Let* $(x^0, y^0) \in \boldsymbol{R}^n_{++} \times \boldsymbol{R}^n_{++}, (u^0, v^0) \in \mathcal{A}, (\hat{x}, \hat{y}) \in \mathcal{F}$, *and let* $\{(x^k, y^k)\}$ *be generated by the algorithm in Figure 9. Then, under Assumptions 4.1.1–4.1.2, for all* $k \ge 0$,

$$\begin{aligned}
&(\hat{x} - (x^0 - u^0)_-)^\top y^k + (\hat{y} - (y^0 - v^0)_-)^\top x^k \\
&\quad \le \quad \hat{x}^\top \hat{y} + {x^k}^\top y^k + \nu_k \left(|x^0 - u^0|^\top \hat{y} + |y^0 - v^0|^\top \hat{x} + \nu_k |x^0 - u^0|^\top |y^0 - v^0| \right).
\end{aligned}$$

Proof Define $\{(u^k, v^k)\}$ according to (2). Then, by Lemma 4.1.4(1), $(u^k, v^k) \in \mathcal{A}$, so $(\hat{x} - u^k)^\top (\hat{y} - v^k) \ge 0$, by Assumption 4.1.1.

Using this fact,

$$\begin{aligned}
& \hat{x}^\top y^k + \hat{y}^\top x^k + (x^k - u^k)^\top y^k + (y^k - v^k)^\top x^k \\
\le \;& \hat{x}^\top y^k + \hat{y}^\top x^k + (x^k - u^k)^\top y^k + (y^k - v^k)^\top x^k + (\hat{x} - u^k)^\top (\hat{y} - v^k) \\
= \;& \hat{x}^\top \hat{y} + {x^k}^\top y^k + (x^k - u^k)^\top \hat{y} + (y^k - v^k)^\top \hat{x} + (x^k - u^k)^\top (y^k - v^k).
\end{aligned} \tag{3}$$

Thus,

$$\begin{aligned}
&(\hat{x} - (x^0 - u^0)_-)^\top y^k + (\hat{y} - (y^0 - v^0)_-)^\top x^k \\
= \;& \hat{x}^\top y^k + \hat{y}^\top x^k - (x^0 - u^0)_-^\top y^k - (y^0 - v^0)_-^\top x^k \\
\le \;& \hat{x}^\top y^k + \hat{y}^\top x^k - (x^k - u^k)_-^\top y^k - (y^k - v^k)_-^\top x^k \quad \text{(by Lemma 4.1.4(2))} \\
\le \;& \hat{x}^\top y^k + \hat{y}^\top x^k + (x^k - u^k)^\top y^k + (y^k - v^k)^\top x^k \\
\le \;& \hat{x}^\top \hat{y} + {x^k}^\top y^k + (x^k - u^k)^\top \hat{y} + (y^k - v^k)^\top \hat{x} + (x^k - u^k)^\top (y^k - v^k) \quad \text{(by (3))} \\
\le \;& \hat{x}^\top \hat{y} + {x^k}^\top y^k + |x^k - u^k|^\top \hat{y} + |y^k - v^k|^\top \hat{x} + |x^k - u^k|^\top |y^k - v^k| \\
& \quad \text{(since } (\hat{x}, \hat{y}) \ge 0) \\
= \;& \hat{x}^\top \hat{y} + {x^k}^\top y^k + \nu_k (|x^0 - u^0|^\top \hat{y} + |y^0 - v^0|^\top \hat{x} + \nu_k |x^0 - u^0|^\top |y^0 - v^0|) \\
& \text{(by Lemma 4.1.4(3)).}
\end{aligned}$$

□

Lemma 4.1.6 *Let* $\{(x^k, y^k)\}$ *be generated by the algorithm in Figure 9 in such a way that* $\phi_0 \ge {x^k}^\top y^k$ *for some* $\phi_0 > 0$. *Further, let* (x^*, y^*) *be a solution to HLCP(M, N, h). If for some* i, $x_i^* > 0$, *then the sequence* $\{y_i^k\}$ *is bounded. Similarly, if* $y_i^* > 0$, *then* $\{x_i^k\}$ *is bounded.*

Proof Define $\tilde{x} := x^* - (x^0 - x^*)_-$ and $\tilde{y} := y^* - (y^0 - y^*)_-$. By applying Lemma 4.1.5 with $(\hat{x}, \hat{y}) = (u^0, v^0) = (x^*, y^*)$, and noting that ${x^*}^\top y^* = 0$, we get

$$\begin{aligned}
&\tilde{x}^\top y^k + \tilde{y}^\top x^k \\
&\quad \leq \; {x^k}^\top y^k + \nu_k(|x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + \nu_k |x^0 - x^*|^\top |y^0 - y^*|) \\
&\quad \leq \; \phi_0 + |x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + |x^0 - x^*||y^0 - y^*| =: C.
\end{aligned}$$

Thus,

$$\sum_{i=1}^{n} \left(\tilde{x}_i y_i^k + x_i^k \tilde{y}_i \right) \leq C. \tag{4}$$

Now, $\tilde{y}_i = \min(y_i^0, y_i^*) \geq 0$ and $\tilde{x}_i = \min(x_i^0, x_i^*) \geq 0$. So, each term on the left side of (4) is nonnegative. Therefore, for all i,

$$x_i^k \tilde{y}_i \leq C, \quad \text{and} \quad \tilde{x}_i y_i^k \leq C.$$

Thus, if $\tilde{y}_i > 0$, then $\{x_i^k\}$ is bounded. Similarly, if $\tilde{x}_i > 0$, then $\{y_i^k\}$ is bounded. $\square$

The next lemma is the counterpart to Zhang (1994, Lemma 6.1).

Lemma 4.1.7 *Let $\{(x^k, y^k)\}$ be generated by the algorithm in Figure 9 in such a way that $\phi_0 \geq {x^k}^\top y^k \geq \beta \nu_k$ for some $\phi_0, \beta > 0$. For any $(x^*, y^*) \in \mathcal{S}$, let $(u^0, v^0) := (x^*, y^*)$ and generate $\{(u^k, v^k)\}$ according to (2). Then there exists $K > 0$ such that*

$$\frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k} \leq K.$$

Proof Partition the indices $\{1, \ldots, n\}$ as follows:

$$H_1 := \{i : x_i^0 \geq x_i^*, y_i^0 \geq y_i^*\}, \; H_2 := \{i : x_i^0 < x_i^*\}, \; H_3 := \{i : y_i^0 < y_i^*\}.$$

. Note that $H_2 \bigcap H_3 = \emptyset$ since one of x_i^* and y_i^* is zero for each i and (x^0, y^0) is strictly positive.

By Lemma 4.1.4(2), $x_i^k - u_i^k = \nu_k(x_i^0 - x_i^*)$ and $y_i^k - v_i^k = \nu_k(y_i^0 - y_i^*)$, so

$$\begin{aligned}
c|x_i^k - u_i^k| &= \begin{cases} (x_i^k - u_i^k), & \text{for } i \in H_1 \bigcup H_3, \\ (u_i^k - x_i^k), & \text{for } i \in H_2, \end{cases} \\
|y_i^k - v_i^k| &= \begin{cases} (y_i^k - v_i^k), & \text{for } i \in H_1 \bigcup H_2, \\ (v_i^k - y_i^k), & \text{for } i \in H_3. \end{cases}
\end{aligned}$$

By Lemma 4.1.6, $i \in H_2 \Rightarrow x_i^* > 0 \Rightarrow \{y_i^k\}$ is bounded $\Rightarrow \{v_i^k\}$ is bounded. Similarly, for $i \in H_3$, $\{x_i^k\}$ and $\{u_i^k\}$ are bounded. Thus, there exists $K_1 > 0$ such that $|y_i^k + v_i^k| < K_1$ for $i \in H_2$, and $|x_i^k + u_i^k| < K_1$ for $i \in H_3$. Thus,

$$\begin{aligned}
&|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k \\
&= \sum_{i\in H_1} \left\{(x_i^k - u_i^k)y_i^k + (y_i^k - v_i^k)x_i^k\right\} + \sum_{i\in H_2} \left\{(u_i^k - x_i^k)y_i^k + (y_i^k - v_i^k)x_i^k\right\} \\
&\quad + \sum_{i\in H_3} \left\{(x_i^k - u_i^k)y_i^k + (v_i^k - y_i^k)x_i^k\right\} \\
(5) \quad &= {x^k}^\top y^k - {u^k}^\top v^k + \sum_{i\in H_1} (x_i^k - u_i^k)(y_i^k - v_i^k) \\
&\quad + \sum_{i\in H_2} (u_i^k - x_i^k)(y_i^k + v_i^k) + \sum_{i\in H_3} (x_i^k + u_i^k)(v_i^k - y_i^k) \\
&\le {x^k}^\top y^k - {u^k}^\top v^k + \sum_{i\in H_1} |x_i^k - u_i^k||y_i^k - v_i^k| \\
&\quad + K_1 \left(\sum_{i\in H_2} |x_i^k - u_i^k| + \sum_{i\in H_3} |y_i^k - v_i^k|\right)
\end{aligned}$$

Note also that

$$\begin{aligned}
0 &\le {x^*}^\top y^k + {y^*}^\top x^k + (x^* - u^k)^\top (y^* - v^k) \quad \text{(By Assumption 4.1.1)} \\
(6) \quad &= {x^*}^\top y^* + (x^k - u^k)^\top y^* + (y^k - v^k)^\top x^* + {u^k}^\top v^k \\
&\le |x^k - u^k|^\top y^* + |y^k - v^k|^\top x^* + {u^k}^\top v^k \quad (\text{since } {x^*}^\top y^* = 0)
\end{aligned}$$

Combining (5) and (6), we get

$$\begin{aligned}
&|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k \\
&\le {x^k}^\top y^k + |x^k - u^k|^\top y^* + |y^k - v^k|^\top x^* + \sum_{i\in H_1} |x_i^k - u_i^k||y_i^k - v_i^k| \\
&\quad + \sum_{i\in H_2} |x_i^k - u_i^k| K_1 + \sum_{i\in H_3} K_1 |y_i^k - v_i^k| \\
&\le {x^k}^\top y^k + \nu_k \Bigg(|x^0 - u^0|^\top y^* + |y^0 - v^0|^\top x^* + \sum_{i\in H_1} |x_i^0 - u_i^0||y_i^0 - v_i^0| \\
&\quad + \sum_{i\in H_2} |x_i^0 - u_i^0| K_1 + \sum_{i\in H_3} K_1 |y_i^0 - v_i^0|\Bigg) \quad \text{(By Lemma 4.1.4(3)).}
\end{aligned}$$

So,

$$\frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k}$$

$$\leq \; 1 + \frac{1}{\beta}\left(|x^0 - u^0|^\top y^* + |y^0 - u^0|^\top x^* + \sum_{i \in H_1} |x_i^0 - u_i^0||y_i^0 - v_i^0|\right.$$
$$\left. + \sum_{i \in H_2} |x_i^0 - u_i^0| K_1 + \sum_{i \in H_3} K_1 |y_i^0 - v_i^0|\right) =: K.$$

□

We are now ready to discuss Zhang's second algorithm. This algorithm is identical to Zhang's first algorithm except that the steplength α_k is defined more precisely. We use the following merit function:

$$\phi(x, y) := x^\top y + \|Mx + Ny - h\|.$$

For convenience we make several additional definitions:

$$x(\alpha) := x^k + \alpha \Delta x^k, \quad y(\alpha) := y^k + \alpha \Delta y^k,$$
$$\phi(\alpha) := \phi(x(\alpha), y(\alpha)), \quad \phi_k := \phi(x^k, y^k).$$

The steplength α_k is chosen so as to minimize the merit function $\phi(\alpha)$ subject to the following constraints:

$$\alpha \in [0, 1], \tag{7}$$
$$x(\alpha) > 0, \quad y(\alpha) > 0, \tag{8}$$
$$x(\alpha)^\top y(\alpha) \geq (1 - \alpha)\nu_k {x^0}^\top y^0, \tag{9}$$
$$x(\alpha)_i y(\alpha)_i \geq (\gamma/n) x(\alpha)^\top y(\alpha), \quad i = 1, \ldots, n. \tag{10}$$

where $\gamma \in (0, 1)$ is chosen so that $\gamma \leq \min(X^0 Y^0 e)/({x^0}^\top y^0/n)$.

Condition (9) implies that

$$\frac{{x^k}^\top y^k}{{x^0}^\top y^0} \geq \nu_k. \tag{11}$$

Zhang's second algorithm is given in Figure 10.

Note that Zhang's second algorithm is a special case of his first algorithm, so all the lemmas proved for the algorithm in Figure 9 also apply for to the algorithm in Figure 10. In particular, since ϕ_k is a decreasing sequence, it follows that ${x^k}^\top y^k \leq \phi_0$, for all k. We now show that the Zhang's second algorithm achieves global Q-linear convergence from any strictly positive starting point.

Theorem 4.1.8 *Let $\{\phi_k\}$ be generated by the algorithm in Figure 10 with σ_k satisfying $0 < \sigma \leq \sigma_k \leq 1/2$. Then $\{\phi_k\}$ converges to zero at a global Q-linear rate, i.e., there exists $\delta \in (0, 1)$ such that*

$$\phi_{k+1} \leq (1 - \delta)\phi_k, \quad k = 0, 1, 2, \cdots.$$

Figure 10: Zhang's Second Algorithm

Given $(x^0, y^0) > 0$, for $k = 0, 1, 2, \ldots$, do

1. Choose $\sigma_k \in [0,1)$ and let $\mu_k = \frac{1}{n}{x^k}^\top y^k$. Solve the linear system (1) for $(\Delta x^k, \Delta y^k)$
2. Set the steplength α_k by minimizing $\phi(\alpha)$ subject to the constraints (7). Let $x^{k+1} = x^k + \alpha_k \Delta x^k$, and $y^{k+1} = y^k + \alpha_k \Delta y^k$.

The proof is nearly identical to Zhang's proof except that we can no longer depend on $(x^k - u^k)$ and $(y^k - v^k)$ being nonnegative. To compensate, we redefine Zhang's constants ξ_k and η_k using absolute values as follows:

$$\xi_k := \left(\frac{n}{\gamma}\right)^{1/2} \frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k}, \tag{12}$$

$$\eta_k := 1 - 2\sigma_k + \frac{{\sigma_k}^2}{\gamma} + \frac{2\nu_k |x^0 - u^0|^\top |y^0 - v^0|}{{x^0}^\top y^0}, \tag{13}$$

With these definitions, the remainder of the proof is identical to Zhang's except that we use the following lemma in place of Zhang (1994, Lemma 6.2).

Lemma 4.1.9 *Let $\{(x^k, y^k)\}$ and $\{(\Delta x^k, \Delta y^k)\}$ be generated by the algorithm in Figure 10 and let $D^k := (Y^k)^{1/2}(X^k)^{-1/2}$. Then*

$$\left\| D^k \Delta x^k \right\|^2 + \left\| (D^k)^{-1} \Delta y^k \right\|^2 \leq \omega_k {x^k}^\top y^k,$$

where

$$\omega_k := \left(\xi_k + \sqrt{\xi_k^2 + \eta_k}\right)^2.$$

Moreover, the sequence $\{\omega_k\}$ is bounded, i.e. there is a constant $\omega > 0$ such that $\omega_k \leq \omega$, for all k.

Proof Define

$$t_k := \left(\left\| D^k \Delta x^k \right\|^2 + \left\| (D^k)^{-1} \Delta y^k \right\|^2 \right)^{1/2}.$$

The proof is nearly identical to Zhang's proof, so we only outline the arguments, pointing out where differences occur. Using identical arguments to Zhang's, the following three inequalities are easily established. These differ from Zhang's results only by the introduction of absolute value signs:

$$\begin{array}{rcl} \Delta {x^k}^\top \Delta y^k & \geq & -e^\top (D^k)^{-1} |y^k - v^k| t_k \\ & & -e^\top D^k |x^k - u^k| t_k - (x^k - u^k)^\top (y^k - v^k). \end{array} \tag{14}$$

$$e^\top D^k |x^k - u^k| \leq \left(\frac{n {x^k}^\top y^k}{\gamma} \right)^{1/2} \frac{|x^k - u^k|^\top y^k}{{x^k}^\top y^k},$$

and

$$e^\top (D^k)^{-1} |y^k - v^k| \leq \left(\frac{n {x^k}^\top y^k}{\gamma} \right)^{1/2} \frac{|y^k - v^k|^\top x^k}{{x^k}^\top y^k}.$$

Combining these inequalities,

$$\Delta {x^k}^\top \Delta y^k \geq - \left(\frac{n {x^k}^\top y^k}{\gamma} \right)^{1/2} \frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k} t_k - (x^k - u^k)^\top (y^k - v^k),$$

which by (12) is equivalent to

$$\Delta {x^k}^\top \Delta y^k \geq -({x^k}^\top y^k)^{1/2} \xi_k t_k - (x^k - u^k)^\top (y^k - v^k). \tag{15}$$

By Lemma 4.1.4(2) and (11),

$$\begin{array}{rcl} (x^k - u^k)^\top (y^k - v^k) & = & (\nu_k)^2 (x^0 - u^0)^\top (y^0 - v^0) \\ & \leq & (\nu_k)^2 |x^0 - u^0|^\top |y^0 - v^0| \\ & \leq & \dfrac{\nu_k {x^k}^\top y^k |x^0 - u^0|^\top |y^0 - v^0|}{{x^0}^\top y^0}. \end{array} \tag{16}$$

Now, using identical logic as Zhang,

$$\begin{array}{rl} & {x^k}^\top y^k [1 - 2\sigma_k + \sigma_k^2/\gamma] \\ \geq & (t_k)^2 + 2\Delta {x^k}^\top \Delta y^k \\ \geq & (t_k)^2 - 2({x^k}^\top y^k)^{1/2} \xi_k t_k - 2(x^k - u^k)^\top (y^k - v^k) \quad \text{(by (15))} \\ \geq & (t_k)^2 - 2({x^k}^\top y^k)^{1/2} \xi_k t_k - 2 \dfrac{\nu_k {x^k}^\top y^k |x^0 - u^0|^\top |y^0 - v^0|}{{x^0}^\top y^0} \quad \text{(by (16))}. \end{array} \tag{17}$$

Thus, from (13),

$${t_k}^2 - 2({x^k}^\top y^k)^{1/2} \xi_k t_k - {x^k}^\top y^k \eta_k \leq 0.$$

The remainder of the proof is identical to Zhang's proof, except that Lemma 4.1.7 is used in place of Zhang (1994, Lemma 6.1). □

4.2 Wright's Algorithm for the Linear Complementarity Problem

In Wright (1994), a locally Q-subquadratic algorithm is presented for solving the LCP which is based on Zhang's algorithm for solving the HLCP. In fact, Wright's algorithm is a special case of Zhang's first algorithm. Unfortunately, Wright's convergence results, like Zhang's, suffer from the same restriction on the starting point (x^0, y^0) (see Assumption 4.1.3). In this section, we will remove this assumption so that Wright's results will apply to arbitrary strictly positive starting points.

Again, it is desirable to be consistent with the notation used in Wright's paper. We therefore restate the linear complementarity problem as follows:

$$\text{LCP}(M, -h): \qquad \text{Find } (x, y) \in \mathrm{R}^n \times \mathrm{R}^n \text{ such that}$$
$$y = Mx - h, \quad (x, y) \geq (0, 0), \quad x^\top y = 0.$$

Note that this is a special case of $\text{HLCP}(M, N, h)$ with $N = -I$.

The calculation of the search direction in Wright's algorithm is exactly the same as in Zhang's algorithms. By substituting $N = -I$ into (1), we get the following equation for calculating the search direction $(\Delta x^k, \Delta y^k)$.

$$\begin{bmatrix} M & -I \\ Y^k & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = \begin{bmatrix} h - Mx^k + y^k \\ -X^k Y^k e + \sigma_k \mu_k e \end{bmatrix}. \tag{18}$$

The same substitution into the merit function gives us $\phi(x, y) := x^\top y + \|y - Mx + h\|$. The definitions of $\mu_k, X^k, Y^k, x(\alpha)$, and $y(\alpha)$ are unchanged.

The difference in Wright's algorithm is that the constraints placed on the steplength are relaxed in order to achieve local Q-subquadratic convergence. Given the parameters $\gamma_k \in (0, 1)$, and $\beta_k \in [0, 1)$, the steplength α_k is calculated by minimizing the function $\phi(\alpha) := \phi(x(\alpha), y(\alpha))$ subject to the following constraints:

$$\alpha \in [0, 1], \tag{19}$$
$$x(\alpha) > 0, \quad y(\alpha) > 0, \tag{20}$$
$$x(\alpha)^\top y(\alpha) \geq (1 - \beta_k)(1 - \alpha)\nu_k {x^0}^\top y^0, \tag{21}$$
$$x(\alpha)_i y(\alpha)_i \geq (\gamma_k / n) x(\alpha)^\top y(\alpha), \quad i = 1, \ldots, n. \tag{22}$$

The condition (21) is a relaxation of the condition (9) enforced by Zhang's second algorithm. Setting $\beta_k > 0$ allows the reduction in the complementarity gap to exceed the reduction in the feasibility, thereby allowing larger steps. Note that by setting $\beta_k = 0$ we get Zhang's algorithm.

Another notable difference is that Wright's algorithm can use a different γ_k at each iteration in condition (22). In fact, the local Q-subquadratic convergence is dependent on being able to choose successively smaller choices of γ_k at each iteration.

The complete algorithm is given in Figure 11.

Figure 11: Wright's Algorithm

Given $\gamma \in (0, 1/2), \sigma \in (0, 1/2), \rho \in (0, \gamma)$,
    $\phi > 0$, and $(x^0, y^0) > (0, 0)$, with $x_i^0 y_i^0 \geq 2\gamma\mu_0$;
$t_0 \leftarrow 1, \gamma_0 \leftarrow 2\gamma$;
for $k = 0, 1, 2, \ldots$
  if $\phi_k := \phi(x^k, y^k) \leq \phi$
  then Compute a "fast" step by setting $\sigma_k \leftarrow \mu_k, \beta_k \leftarrow \gamma^{t_k}$,
    and $\gamma_k \leftarrow \gamma(1 + \gamma^{t_k})$ and solving (18)–(19)
    to calculate $(\Delta x^k, \Delta y^k)$ and α_k;
    if $\phi(x^k + \alpha_k \Delta x^k, y^k + \alpha_k \Delta y^k) \leq \rho\phi_k$
    then $(x^{k+1}, y^{k+1}) \leftarrow (x^k, y^k) + \alpha_k(\Delta x^k, \Delta y^k)$
      $t_{k+1} \leftarrow t_k + 1$;
      go to next k;
    end if
  end if
  Compute a "safe" step by setting $\sigma_k \in [\sigma, 1/2], \beta_k = 0$,
  and $\gamma_k = \gamma_{k-1}$, and solving (18)–(19)
  to calculate $(\Delta x^k, \Delta y^k)$ and α_k;
  $(x^{k+1}, y^{k+1}) \leftarrow (x^k, y^k) + \alpha_k(\Delta x^k, \Delta y^k)$
  $t_{k+1} \leftarrow t_k$;
  go to next k;
end for.

At each iteration, either a *safe step* or a *fast step* is taken. A safe step works exactly like Zhang's algorithm; we set $\beta_k = 0$ and hold γ_k constant for the next iteration. A fast step works by setting $\beta_k > 0$ and $\sigma_k = \mu_k$. It is these fast steps that allow the algorithm to attain local Q-subquadratic convergence. Unfortunately, a fast step requires reducing the size of γ_k for subsequent iterations. Therefore, the fast step is only taken if it results in a significant decrease in ϕ. If it doesn't, the step is discarded and a "safe step" is taken instead.

We again will find it convenient to refer to the sets $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{F}$ defined earlier. For convenience we restate their definitions here with $N = -I$:

$\mathcal{S} = \{(x,y) \in \mathrm{R}^{2n} : y = Mx - h, (x,y) \geq 0, x^\top y = 0\}$, i.e., the solution set,
$\mathcal{A} = \{(x,y) \in \mathrm{R}^{2n} : y = Mx - h\}$,
$\mathcal{F} = \{(x,y) \in \mathcal{A} : (x,y) \geq 0\}$, i.e., the set of feasible points,

Wright proves two convergence results for his algorithm. First, he shows that the algorithm has global Q-linear convergence. Second, he shows that the algorithm attains local Q-subquadratic convergence. His results are based on the following explicit assumptions:

Assumption 4.2.1 *M is positive semidefinite.*

Assumption 4.2.2 *LCP($M,-h$) has a strictly feasible point $(\bar{x},\bar{y})$.*

Assumption 4.2.3 *The solution set for LCP($M,-h$) is nonempty and, moreover, there is a strictly complementary solution (x^*,y^*).*

Wright's results, like Zhang's, are also dependent on Assumption 4.1.3. These assumptions are more restrictive than Zhang's assumptions. Assumption 4.2.1 is equivalent to Assumption 4.1.1 in the case of LCP, but Assumption 4.2.2 is stronger than Assumption 4.1.2. In fact, Zhang's assumptions are sufficient to prove the global Q-linear convergence. However, Wright's more restrictive assumptions are used to prove the local Q-subquadratic convergence. We now proceed to prove global Q-linear convergence of Wright's algorithm using only Assumptions 4.1.1–4.1.2.

Note that since Wright's algorithm is a special case of Zhang's first algorithm, Lemmas 4.1.4–4.1.7 are applicable for it. We shall also need the following result from Wright (1994):

Lemma 4.2.4 *Let $\hat{\beta} := \prod_{k=0}^{\infty}(1-\beta_k)$ where β_k is defined in Figure 11, and let $\mu_k := {x^k}^\top y^k/n$. Then $\hat{\beta} > 0$ and*

$$\begin{array}{rcl} \mu_k & \geq & \hat{\beta}\nu_k\mu_0, \quad \textit{and} \\ {x^k}^\top y^k & \geq & \hat{\beta}\nu_k {x^0}^\top y^0. \end{array}$$

Proof (Wright 1994), Lemmas 3.1 and 3.2. □

We now define the quantities $\hat{\xi}_k, \hat{\eta}_k$, and $\hat{\omega}_k$, which we use to establish the convergence rates:

$$\begin{aligned}
\hat{\xi}_k &:= \left(\frac{n}{\gamma_k}\right)^{1/2} \frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k}, \\
\hat{\eta}_k &:= 1 - 2\sigma_k + \frac{(\sigma_k)^2}{\gamma_k} + \frac{2\nu_k |x^0 - u^0|^\top |y^0 - v^0|}{\hat{\beta} {x^0}^\top y^0} \\
\hat{\omega}_k &:= \left(\hat{\xi}_k + \sqrt{(\hat{\xi}_k)^2 + \hat{\eta}_k}\right)^2.
\end{aligned}$$

Note the similarity to the definitions of ξ_k and η_k in, (12), (13). $\hat{\xi}_k$ is identical to ξ_k except that it has γ_k in the denominator instead of γ. $\hat{\eta}_k$ differs from η_k only by dividing the last term by $\hat{\beta}$.

Lemma 4.2.5 *Let $\{(x^k, y^k)\}$ and $\{(\Delta x^k, \Delta y^k)\}$ be generated by the algorithm in Figure 11 and let $D^k := (Y^k)^{1/2}(X^k)^{-1/2}$. Then*

$$\left\| D^k \Delta x^k \right\|^2 + \left\| (D^k)^{-1} \Delta y^k \right\|^2 \le \hat{\omega}_k {x^k}^\top y^k. \tag{23}$$

Moreover, the sequence $\{\hat{\omega}_k\}$ is bounded.

Proof We can make minor modifications to the proof of Lemma 4.1.9 to prove (23). These changes are as follows:

1. Replace ξ_k, η_k, and ω_k by $\hat{\xi}_k, \hat{\eta}_k$, and $\hat{\omega}_k$, respectively.
2. Replace (16) with the inequality

$$(x^k - u^k)^\top (y^k - v^k) \le \frac{\nu_k {x^k}^\top y^k |x^0 - u^0|^\top |y^0 - v^0|}{\hat{\beta} {x^0}^\top y^0}. \tag{24}$$

   which we justify by Lemmas 4.1.4(2) and 4.2.4.
3. Replace the last line of (17) with

$$\begin{aligned}
&{x^k}^\top y^k [1 - 2\sigma_k + \sigma_k^2/\gamma_k] \\
&\ge \ (t_k)^2 - 2({x^k}^\top y^k)^{1/2} \hat{\xi}_k t_k - 2\frac{\nu_k {x^k}^\top y^k |x^0 - u^0|^\top |y^0 - v^0|}{\hat{\beta} {x^0}^\top y^0} \\
&\qquad \text{by (24).}
\end{aligned}$$

To show that $\{\hat{\omega}\}$ is bounded, observe that $\{\hat{\eta}_k\}$ is bounded. Moreover, by Lemmas 4.1.7 and 4.2.4,

$$|\hat{\xi}_k| = \left(\frac{n}{\gamma_k}\right)^{1/2} \frac{|x^k - u^k|^\top y^k + |y^k - v^k|^\top x^k}{{x^k}^\top y^k} \leq \left(\frac{n}{\gamma}\right)^{1/2} K,$$

so $\{\hat{\xi}_k\}$ is also bounded. Hence, $\{\hat{\omega}_k\}$ is bounded. $\square$

We can now state the global Q-linear convergence theorem.

Theorem 4.2.6 *Under Assumptions 4.1.1–4.1.2, there is a constant $\delta \in (0,1)$ such that*

$$\phi_{k+1} \leq (1-\delta)\phi_k, \quad k = 0, \ldots,$$

that is, the Wright's algorithm converges globally and Q-linearly.

Proof The proof is identical to the proof of Wright (1994, Theorem 4.2) but using $\hat{\xi}_k$, $\hat{\eta}_k$, and $\hat{\omega}_k$ in place of ξ_k, η_k, and ω_k, and also using Lemma 4.2.5 in place of Wright (1994, Lemma 4.1). $\square$

We now turn our attention toward proving global Q-subquadratic convergence. We shall need to use Wright's stronger assumptions 4.2.1–4.2.3.

We first prove two lemmas which place bounds on the iterates (x^k, y^k). We need the following definitions:

$$B = \{i | x_i^* > 0\}, \quad N = \{i | y_i^* > 0\}.$$

where (x^*, y^*) is the strictly complementary solution guaranteed by Assumption 4.2.3. Note that $N \bigcup B = \{1, 2, \ldots, n\}$ and $N \bigcap B = \emptyset$.

Lemma 4.2.7 *Let $\{(x^k, y^k)\}$ be generated by the algorithm in Figure 11. There is a constant $C_1 > 0$ such that*

$$\begin{array}{ll} i \in N \Rightarrow x_i^k \leq C_1 \mu_k, & y_i^k \geq \gamma / C_1, \\ i \in B \Rightarrow y_i^k \leq C_1 \mu_k, & x_i^k \geq \gamma / C_1. \end{array} \tag{25}$$

Proof Define $\tilde{x} := x^* - (x^0 - x^*)_-$ and $\tilde{y} := y^* - (y^0 - y^*)_-$, where (x^*, y^*) is the strictly complementary solution guaranteed by Assumption 4.2.3. By applying Lemma 4.1.5 with $(\hat{x}, \hat{y}) = (u^0, v^0) = (x^*, y^*)$, and noting that ${x^*}^\top y^* = 0$, we get

$$\begin{aligned}
&\tilde{x}^\top y^k + \tilde{y}^\top x^k \\
&\quad \leq \; {x^k}^\top y^k + \nu_k (|x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + |x^0 - x^*|^\top |y^0 - y^*|) \\
&\quad = \; {x^k}^\top y^k \left(1 + \nu_k \left(\frac{|x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + |x^0 - x^*|^\top |y^0 - y^*|}{{x^k}^\top y^k}\right)\right) \\
&\quad \leq \; \bar{C}_1 {x^k}^\top y^k, \quad \text{by Lemmas 4.1.4(3) and 4.2.4,}
\end{aligned}$$

where

$$\bar{C}_1 := \left(1 + \frac{|x^0 - x^*|^\top y^* + |y^0 - y^*|^\top x^* + |x^0 - x^*|^\top |y^0 - y^*|}{\hat{\beta} {x^0}^\top y^0}\right).$$

Thus,

$$\sum_{i=1}^{n} \left(\tilde{x}_i^\top y_i^k + {x_i^k}^\top \tilde{y}_i\right) \leq n\bar{C}_1 \mu_k. \tag{26}$$

Now, $\tilde{y}_i = \min(y_i^0, y_i^*) \geq 0$ and $\tilde{x}_i = \min(x_i^0, x_i^*) \geq 0$. So, each term on the left side of (26) is nonnegative. Therefore,

$$x_i^k \tilde{y}_i \leq n\bar{C}_1 \mu_k, \quad \text{and} \quad \tilde{x}_i y_i^k \leq n\bar{C}_1 \mu_k.$$

Note further that for $i \in N$, $\tilde{y}_i > 0$, so

$$x_i^k \leq \frac{n\bar{C}_1}{\tilde{y}_i} \mu_k,$$

Similarly, for $i \in B, \tilde{x}_i > 0$ and

$$y_i^k \leq \frac{n\bar{C}_1}{\tilde{x}_i} \mu_k,$$

Finally, we obtain our result by taking

$$C_1 := n\bar{C}_1 \max\left(\max_{i \in B} \frac{1}{\tilde{x}_i}, \max_{i \in N} \frac{1}{\tilde{y}_i}\right).$$

Then for $i \in B, x_i^k \leq C_1 \mu_k$, and by (22)

$$x_i^k y_i^k \geq \gamma_k \mu_k \Rightarrow y_i^k \geq \frac{\gamma_k \mu_k}{x_i^k} \geq \frac{\gamma_k}{C_1} \geq \frac{\gamma}{C_1}.$$

Similarly, for $i \in N, y_i^k \leq C_1 \mu_k$ and $x_i^k \geq \gamma / C_1$. $\square$

Lemma 4.2.8 *Let $\{(x^k, y^k)\}$ be generated by the algorithm in Figure 11. There is a constant $C_2 > 0$ such that*

$$0 < x_i^k \leq C_2, \quad 0 < y_i^k \leq C_2. \tag{27}$$

Proof Define $\check{x} := \bar{x} - (x^0 - \bar{x})_-$ and $\check{y} := \bar{y} - (y^0 - \bar{y})_-$, where $(\bar{x}, \bar{y})$ is the strictly feasible point guaranteed by Assumption 4.2.2. Note that $(\check{x}, \check{y}) > 0$. Now, by applying Lemma 4.1.5 with $(\hat{x}, \hat{y}) = (u^0, v^0) = (\bar{x}, \bar{y})$, we get

$$
\begin{aligned}
&\check{x}^\top y^k + \check{y}^\top x^k \\
&\quad \leq \ \bar{x}^\top \bar{y} + {x^k}^\top y^k + \nu_k(|x^0 - \bar{x}|^\top \bar{y} + |y^0 - \bar{y}|^\top \bar{x} + |x^0 - \bar{x}|^\top |y^0 - \bar{y}|) \\
&\quad \leq \ \bar{x}^\top \bar{y} + {x^0}^\top y^0 + |x^0 - \bar{x}|^\top \bar{y} + |y^0 - \bar{y}|^\top \bar{x} + |x^0 - \bar{x}|^\top |y^0 - \bar{y}| \\
&\quad =: \ \bar{C}_2.
\end{aligned}
$$

Hence,

$$
0 < y_i^k \leq \frac{\bar{C}_2}{\check{x}_i}, \quad 0 < x_i^k \leq \frac{\bar{C}_2}{\check{y}_i}, \quad i = 1, 2, \ldots, n.
$$

The result is obtained by setting

$$
C_2 := \bar{C}_2 \max\left(\max_{i=1,\ldots,n} \frac{1}{\check{x}_i}, \max_{i=1,\ldots,n} \frac{1}{\check{y}_i} \right).
$$

□

The remainder of Wright's results can now be proved simply by replacing all references to Wright (1994, Lemmas 3.4 and 3.5) by references to our Lemmas 4.2.7 and 4.2.8.

Theorem 4.2.9 *Under Assumptions 4.2.1–4.2.3, there is an $\hat{\epsilon} > 0$ such that if $\bar{K}$ is the smallest integer such that $\phi_{\bar{K}} \leq \hat{\epsilon}$, then*

1. *the algorithm will take fast steps at iteration $\bar{K}$ and at all subsequent iterations, and*

2. *the sequences $\{\mu_k\}$ and $\{\phi_k\}$ converge Q-subquadratically to zero.*

Proof The proof is identical to the proof of Wright (1994, Theorems 6.3 and 6.4), but using Lemmas 4.2.7 and 4.2.8 in place of Wright (1994, Lemmas 3.4 and 3.5). □

Corollary 4.2.10 *The algorithm in Figure 11 has local Q-subquadratic convergence.*

Proof Follows immediately from Theorem 4.2.9. □

4.3 Algorithm for Linear MCP

We now turn our attention to the linear mixed complementarity problem. To employ the infeasible interior-point methodology, we reformulate the LMCP as follows:

$$
\begin{array}{ll}
\mathrm{LMCP}(M,q,\mathrm{B}): & \text{Given } M \in \mathrm{R}^{n\times n}, q \in \mathrm{R}^n, \text{ find a vector} \\
& \text{triple } (z,w,v) \in \mathrm{R}^n \times \mathrm{R}^n \times \mathrm{R}^n \text{ such that} \\
& \quad w - v = Mz + q, \\
& \quad z \in \mathrm{B}, \quad w \geq 0, \quad v \geq 0 \\
& \quad \langle z - l, w\rangle = 0 \quad \text{and} \quad \langle u - z, v\rangle = 0.
\end{array}
$$

The algorithm presented in this section makes the additional assumption that for each index i, at least one of l_i and u_i is finite. This restriction was motivated by a desire to avoid complications imposed by a lineality space. Without this restriction, an adaptation of our analysis would seem to require an additional assumption regarding the invertibility of a submatrix over the lineality space. However, Cao & Ferris (1995*a*) show that such an assumption is unnecessary by using a reduction technique to remove the lineality. Unfortunately, the linear algebra involved in this approach would seem to make it impractical in this context.

In the analysis of this section, it will be convenient to define a generalized inner product $\langle(\cdot,\cdot),\cdot\rangle : \mathrm{R}^n \times \mathrm{R}^n \times \mathrm{R}^n \to \mathrm{R}$, by

$$\langle (w,v), z\rangle := \sum_{i\in\{i:l_i>-\infty\}} w_i(z_i - l_i) + \sum_{i\in\{i:u_i<\infty\}} v_i(u_i - z_i).$$

The first step in deriving an algorithm of the LMCP is to reformulate it as an LCP. To do this, we first make some observations about $\mathrm{LMCP}(M,q,\mathrm{B})$. Let us partition the indices according to which bounds are finite.

$$H := \{i : -\infty < l_i, u_i < \infty\}, \quad J := \{i : u_i = \infty\}, \quad K := \{i : l_i = -\infty\}.$$

Note that H, J, and K are disjoint and further that $H \bigcup J \bigcup K = \{1,\ldots,n\}$. Without loss of generality, we can assume that the rows and columns of M and the vectors q, l, and u are ordered so that the indices in H occur first, those in J occur second, and those in K occur last. Let p, s, and t be the cardinality of the sets H, J, and K respectively.

Note that if (z^*, w^*, v^*) is a solution to $\mathrm{LMCP}(M,q,\mathrm{B})$, then $w_K^* = 0$, and $v_J^* = 0$. Thus, we can remove w_K and v_J from the problem. This motivates the definition of the set

$$\mathcal{G}_1 := \{(w,v) \in \mathrm{R}^n \times \mathrm{R}^n : w_K = 0, v_J = 0\}.$$

We now define an invertible linear map $L : \mathcal{G}_1 \to \mathrm{R}^{p+s} \times \mathrm{R}^{p+t}$ by

$$L(w, v) := (\hat{w}, \hat{v}),$$

$$\text{where} \quad \hat{w} := \begin{bmatrix} w_H \\ w_J \end{bmatrix} \quad \textit{and} \quad \hat{v} := \begin{bmatrix} v_H \\ v_K \end{bmatrix}.$$

Our plan now is to create an algorithm that generates iterates $\{(z^k, \hat{w}^k, \hat{v}^k)\}$ such that $\{(z^k, w^k, v^k)\} := \{(z^k, L^{-1}(\hat{w}^k, \hat{v}^k))\}$ converges to a solution (z^*, w^*, v^*) of $\mathrm{LMCP}(M, q, \mathrm{B})$. To do this, we exploit the fact that $\mathrm{LMCP}(M, q, \mathrm{B})$ can be reformulated as an LCP with higher dimension.

Define the maps

$$\begin{aligned} \mathcal{X} &: \mathrm{R}^n \times \mathcal{G}_1 \to \mathrm{R}^{n+p} := (z, w, v) \mapsto (z_H - l_H, z_J - l_J, u_K - z_K, v_H), \\ \mathcal{Y} &: \mathrm{R}^n \times \mathcal{G}_1 \to \mathrm{R}^{n+p} := (z, w, v) \mapsto (w_H, w_J, v_K, u_H - z_H). \end{aligned}$$

In order to refer to the last p components of $\mathcal{X}$ and $\mathcal{Y}$, we define the set of indices $\hat{H} = H + n$. For example, if $x = \mathcal{X}(z, w, v)$, then $x_{\hat{H}} = v_H$. Now, define the set

$$\mathcal{G}_2 := \{(x, y) \in \mathrm{R}^{n+p} \times \mathrm{R}^{n+p} : x_H + y_{\hat{H}} = u_H - l_H\}.$$

We can now define an invertible linear map $T : \mathrm{R}^n \times \mathcal{G}_1 \to \mathcal{G}_2$ by the relation

$$T(z, w, v) := (\mathcal{X}(z, w, v), \mathcal{Y}(z, w, v)).$$

Using this mapping, the LMCP can be reformulated as follows:

$$(x, y) := T(z, w, v), \tag{28}$$

$$\hat{M} := \begin{bmatrix} M_{HH} & M_{HJ} & -M_{HK} & I \\ M_{JH} & M_{JJ} & -M_{JK} & 0 \\ -M_{KH} & -M_{KJ} & M_{KK} & 0 \\ -I & 0 & 0 & 0 \end{bmatrix}, \tag{29}$$

$$\hat{h} := \begin{bmatrix} -q_H - M_{HH} l_H - M_{HJ} l_J - M_{HK} u_K \\ -q_J - M_{JH} l_H - M_{JJ} l_J - M_{JK} u_K \\ q_K + M_{KH} l_H + M_{KJ} l_J + M_{KK} u_K \\ -u_H + l_H \end{bmatrix}, \quad m := n + p. \tag{30}$$

With these definitions, $\mathrm{LMCP}(M, q, \mathrm{B})$ is equivalent to the linear complementarity problem formed by replacing M, h, and n in $\mathrm{LCP}(M, -h)$ by $\hat{M}$, $\hat{h}$, and m, respectively. Thus, given a starting point $(z^0, w^0, v^0) \in \mathrm{ri}(\mathrm{B} \times \mathcal{G}_1)$ we can solve $\mathrm{LMCP}(M, q, \mathrm{B})$ simply by applying Wright's algorithm with the starting point $(x^0, y^0) := T(z^0, w^0, v^0)$. If the algorithm finds a solution (x^*, y^*) of $\mathrm{LCP}(M, -h)$

with $M = \hat{M}$, $h = \hat{h}$, then $(x^*, y^*) \in \mathcal{G}_2$ and $(z^*, w^*, v^*) := T^{-1}(x^*, y^*)$ is a solution of LMCP$(M, q, \mathbf{B})$.

Our plan now is to substitute (28)–(30) into Wright's algorithm and to simplify in order to produce an algorithm that generates iterates $\{(z^k, \hat{w}^k, \hat{v}^k)\}$, such that for all k,

$$T(z^k, L^{-1}(\hat{w}^k, \hat{v}^k)) = (x^k, y^k),$$

where $\{(x^k, y^k)\}$ are the iterates generated by Wright's algorithm. Throughout our discussion, we will occasionally find it convenient to refer to (w^k, v^k). In such cases, we are implying the relationship $(w^k, v^k) = L^{-1}(\hat{w}^k, \hat{v}^k)$.

We look first at the equation used to calculate the search direction. Direct substitution into (18) yields

$$\begin{bmatrix} M_{HH} & M_{HJ} & -M_{HK} & I_p & -I_p & 0 & 0 & 0 \\ M_{JH} & M_{JJ} & -M_{JK} & 0 & 0 & -I_s & 0 & 0 \\ -M_{KH} & -M_{KJ} & M_{KK} & 0 & 0 & 0 & -I_t & 0 \\ -I_p & 0 & 0 & 0 & 0 & 0 & 0 & -I_p \\ W_H^k & 0 & 0 & 0 & R_H^k & 0 & 0 & 0 \\ 0 & W_J^k & 0 & 0 & 0 & R_J^k & 0 & 0 \\ 0 & 0 & V_K^k & 0 & 0 & 0 & S_K^k & 0 \\ 0 & 0 & 0 & S_H^k & 0 & 0 & 0 & V_H^k \end{bmatrix} \begin{bmatrix} \Delta x_H^k \\ \Delta x_J^k \\ \Delta x_K^k \\ \Delta x_{\hat{H}}^k \\ \Delta y_H^k \\ \Delta y_J^k \\ \Delta y_K^k \\ \Delta y_{\hat{H}}^k \end{bmatrix} \tag{31}$$

$$= \begin{bmatrix} w_H^k - v_H^k - M_{H\cdot}z^k - q_H \\ w_J^k - M_{J\cdot}z^k - q_J \\ v_K^k + M_{K\cdot}z^k + q_K \\ 0 \\ -W_H^k R_H^k e + \sigma_k \mu_k e \\ -W_J^k R_J^k e + \sigma_k \mu_k e \\ -V_K^k S_K^k e + \sigma_k \mu_k e \\ -V_H^k S_H^k e + \sigma_k \mu_k e \end{bmatrix} \tag{32}$$

where $R := \operatorname{diag}(z - l)$, and $S := \operatorname{diag}(u - z)$. By the fourth row of this system, $\Delta x_H^k = -\Delta y_{\hat{H}}^k$. We can thus replace the last equation of (31) with $-V^k \Delta x_H^k + S_H^k \Delta x_{\hat{H}}^k = -V_H^k S_H^k e + \sigma_k \mu_k e$. Removing, the fourth row and the last column, we get

$$\begin{bmatrix} M_{HH} & M_{HJ} & M_{HK} & I_p & -I_p & 0 & 0 \\ M_{JH} & M_{JJ} & M_{JK} & 0 & 0 & -I_s & 0 \\ M_{KH} & M_{KJ} & M_{KK} & 0 & 0 & 0 & I_t \\ W_H^k & 0 & 0 & 0 & R_H^k & 0 & 0 \\ 0 & W_J^k & 0 & 0 & 0 & R_J^k & 0 \\ 0 & 0 & -V_K^k & 0 & 0 & 0 & S_K^k \\ -V_H^k & 0 & 0 & S_H^k & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_H^k \\ \Delta x_J^k \\ -\Delta x_K^k \\ \Delta x_{\hat{H}}^k \\ \Delta y_H^k \\ \Delta y_J^k \\ \Delta y_K^k \end{bmatrix} \tag{33}$$

$$(34) \qquad = \begin{bmatrix} w_H^k - v_H^k - M_{H.}z^k - q_H \\ w_J^k - M_{H.}z^k - q_J \\ -v_K^k - M_{H.}z^k - q_K \\ -W_H^k R_H^k e + \sigma_k \mu_k e \\ -W_J^k R_J^k e + \sigma_k \mu_k e \\ -V_K^k S_K^k e + \sigma_k \mu_k e \\ -V_H^k S_H^k e + \sigma_k \mu_k e \end{bmatrix}.$$

Finally, moving column 4 to column 6, and switching rows 6 and 7, we obtain the equation

$$(35) \qquad \begin{bmatrix} M & -I_W & I_V \\ \hat{W}^k & \hat{R}^k & 0 \\ -\hat{V}^k & 0 & \hat{S}^k \end{bmatrix} \begin{bmatrix} \Delta z^k \\ \Delta \hat{w}^k \\ \Delta \hat{v}^k \end{bmatrix} = \begin{bmatrix} -q - Mz^k - v^k + w^k \\ -\hat{R}^k \hat{W}^k e + \sigma_k \mu_k e \\ -\hat{S}^k \hat{V}^k e + \sigma_k \mu_k e \end{bmatrix},$$

where

$$\Delta z^k := \begin{bmatrix} \Delta x_H^k \\ \Delta x_J^k \\ -\Delta x_K^k \end{bmatrix}, \quad \Delta \hat{w}^k := \begin{bmatrix} \Delta y_H^k \\ \Delta y_J^k \end{bmatrix}, \quad \Delta \hat{v}^k := \begin{bmatrix} \Delta x_{\hat{H}}^k \\ \Delta y_K^k \end{bmatrix},$$

$$I_W := \begin{bmatrix} I_p & 0 \\ 0 & I_s \\ 0 & 0 \end{bmatrix}, \quad I_V := \begin{bmatrix} I_p & 0 \\ 0 & 0 \\ 0 & I_t \end{bmatrix},$$

$$\hat{W}^k := \begin{bmatrix} W_H^k & 0 & 0 \\ 0 & W_J^k & 0 \end{bmatrix}, \quad \hat{V}^k := \begin{bmatrix} V_H^k & 0 & 0 \\ 0 & 0 & V_K^k \end{bmatrix},$$

$$\hat{R} := \begin{bmatrix} R_H & 0 \\ 0 & R_J \end{bmatrix}, \quad \hat{S} := \begin{bmatrix} S_H & 0 \\ 0 & S_k \end{bmatrix}.$$

We now turn our attention to the equations governing the calculation of the steplength. Let us define the merit function

$$\psi(z, \hat{w}, \hat{v}) := \phi(x, y), \qquad \text{where } (x, y) = T(z, w, v).$$

Then $\psi(z, \hat{w}, \hat{v}) = \langle (w, v), z \rangle + \|w - v - Mz - q\|$. Define

$$\begin{aligned} z(\alpha) &:= z^k + \alpha \Delta z^k, \\ \hat{w}(\alpha) &:= \hat{w}^k + \alpha \Delta \hat{w}^k, \\ \hat{v}(\alpha) &:= \hat{v}^k + \alpha \Delta \hat{v}^k. \end{aligned}$$

Direct substitution into (19) gives the following equations for calculating the steplength for the LMCP algorithm:

$$(36)\qquad \alpha_k = \arg\min_\alpha \psi(z(\alpha), \hat{w}(\alpha), \hat{v}(\alpha))$$

subject to

$$(37)\qquad \alpha \in [0,1],$$

$$(38)\qquad l < z(\alpha) < u, \quad \text{and} \quad (\hat{w}(\alpha), \hat{v}(\alpha)) > 0,$$

$$(39)\qquad \langle (w(\alpha), v(\alpha)), z(\alpha)\rangle \geq (1-\beta_k)(1-\alpha)\langle (w^k, v^k), z^k\rangle,$$

$$(40)\qquad (z(\alpha)_i - l_i)w(\alpha)_i \geq (\gamma_k/(n+p))\langle (w(\alpha), v(\alpha)), z(\alpha)\rangle, \quad i \in H \bigcup J,$$

$$(41)\qquad (u_i - z(\alpha)_i)v(\alpha)_i \geq (\gamma_k/(n+p))\langle (w(\alpha), v(\alpha)), z(\alpha)\rangle, \quad i \in H \bigcup K.$$

Finally, we note that from Figure 11, $\mu_k = {x^k}^\top y^k/m = \langle (w^k, v^k), z^k\rangle/(n+p)$. The complete algorithm is given in Figure 12.

By construction, there is a 1-1 correspondence between the iterates $\{(x^k, y^k)\}$ of Wright's algorithm and the iterates $\{(z^k, \hat{w}^k, \hat{v}^k)\}$ of the LMCP algorithm. This 1-1 correspondence is given by $T(z^k, L^{-1}(\hat{w}^k, \hat{v}^k)) = (x^k, y^k)$. Thus, we can prove convergence results for the iterates $\{(z^k, \hat{w}^k, \hat{v}^k)\}$ of the LMCP algorithm simply by analyzing the iterates $\{(x^k, y^k)\}$ of Wright's algorithm.

We now state several convergence theorems for the LMCP algorithm. These results are based on the following assumptions:

Assumption 4.3.1 *M is positive semidefinite.*

Assumption 4.3.2 *LMCP(M, q, B) has a point $(\bar{z}, \bar{w}, \bar{v}) \in \mathcal{I}_+ := \{(z, w, v) : l < z < u, L(w,v) > 0, w_K = 0, v_J = 0\}$ and $\bar{w} - \bar{v} = M\bar{z} + q$.*

Assumption 4.3.3 *The solution set for LMCP(M, q, B) is nonempty and, moreover, there is a strictly complementary solution (z^*, w^*, v^*), that is $z_i^* = l_i \Longrightarrow w_i^* > 0$ and $z_i^* = u_i \Longrightarrow v_i^* > 0$.*

The following lemma shows that the above assumptions guarantee the assumptions for the convergence of Wright's algorithm.

Lemma 4.3.4 *Given the relationship between LMCP(M, q, B) and LCP($M, -h$) defined by equations (28) – (30), (i) Assumption 4.3.1 $\Rightarrow$ Assumption 4.2.1; (ii) Assumption 4.3.2 $\Rightarrow$ Assumption 4.2.2; (iii) Assumption 4.3.3 $\Rightarrow$ Assumption 4.2.3;*

Proof (i)

$$\begin{aligned}
&[x^\top, y^\top, z^\top, w^\top]\hat{M}[x; y; z; w] \\
&\quad = x^\top M_{HH}x + x^\top M_{HJ}y + y^\top M_{JH}x - x^\top M_{HK}z - z^\top M_{KH}x + x^\top I w \\
&\qquad -w^\top I x + y^\top M_{JJ}y - y^\top M_{JK}z - z^\top M_{KJ}y + z^\top M_{KK}z \\
&\quad = [x^\top, y^\top, -z^\top]M[x; y; -z],
\end{aligned}$$

Figure 12: LMCP Algorithm

Given $\gamma \in (0, 1/2), \sigma \in (0, 1/2), \rho \in (0, \gamma)$,
    $\psi > 0$, and $(z^0, \hat{w}^0, \hat{v}^0)$ with $l < z^0 < u$, $(\hat{w}^0, \hat{v}^0) > 0$,
    $(z_i - l_i)w_i \geq 2\gamma\mu_0$ for $i \in H \bigcup J$, and $v_i(u_i - z_i) \geq 2\gamma\mu_0$ for $i \in H \bigcup K$;
$t_0 \leftarrow 1, \gamma_0 \leftarrow 2\gamma$;
for $k = 0, 1, 2, \ldots$
  if $\psi_k := \psi(z^k, \hat{w}^k, \hat{v}^k) \leq \psi$
  then Compute a "fast" step by setting $\sigma_k \leftarrow \mu_k, \beta_k \leftarrow \gamma^{t_k}$,
    and $\gamma_k \leftarrow \gamma(1 + \gamma^{t_k})$ and solving (35)–(37) to calculate
    $(\Delta z^k, \Delta\hat{w}^k, \Delta\hat{v}^k)$ and α_k;
    if $\psi_k(z^k + \alpha_k \Delta z^k, \hat{w}^k + \alpha_k \Delta\hat{w}^k, \hat{v}^k + \alpha_k \Delta\hat{v}^k) \leq \rho\psi_k$
    then $(z^{k+1}, \hat{w}^{k+1}, \hat{v}^{k+1}) \leftarrow (z^k, \hat{w}^k, \hat{v}^k) + \alpha_k(\Delta z^k, \Delta\hat{w}^k, \Delta\hat{v}^k)$,
      $t_{k+1} \leftarrow t_k + 1$;
      go to next k;
    end if
  end if
  Compute the "safe" step by setting $\sigma_k \in [\sigma, 1/2], \beta_k \leftarrow 0, \gamma_k \leftarrow \gamma_{k-1}$
  and solving (35)–(37) to calculate $(\Delta z^k, \Delta\hat{w}^k, \Delta\hat{v}^k)$ and α_k;
  $(z^{k+1}, \hat{w}^{k+1}, \hat{v}^{k+1}) \leftarrow (z^k, \hat{w}^k, \hat{v}^k) + \alpha_k(\Delta z^k, \Delta\hat{w}^k, \Delta\hat{v}^k)$,
  $t_{k+1} \leftarrow t_k$;
  $(z^{k+1}, \hat{w}^{k+1}, \hat{v}^{k+1}) \leftarrow (z(\alpha_k), \hat{w}(\alpha_k), \hat{v}(\alpha_k))$;
  go to next k;
end for.

so $\hat{M}$ is positive semidefinite whenever M is positive semidefinite.

(ii) If $(\bar{z}, \bar{w}, \bar{v}) \in \mathcal{I}_+$, then we can define $(\bar{x}, \bar{y}) := T(\bar{z}, \bar{w}, \bar{v})$. Clearly, $(\bar{x}, \bar{y}) > 0$. Moreover,

$$\hat{M}\bar{x} - \hat{h} = \hat{M} \begin{bmatrix} \bar{z}_H - l_H \\ \bar{z}_J - l_J \\ u_K - \bar{z}_K \\ \bar{v}_H \end{bmatrix} - \hat{h} = \begin{bmatrix} M_{H.}\bar{z} + q_H \\ M_{J.}\bar{z} + q_J \\ -M_{K.}\bar{z} - q_K \\ u_H - \bar{z}_H \end{bmatrix} = \begin{bmatrix} \bar{w}_H \\ \bar{w}_J \\ \bar{v}_K \\ u - \bar{z}_H \end{bmatrix} = \bar{y}.$$

Thus, $(\bar{x}, \bar{y})$ is a strictly feasible point for LCP$(M, -h)$.

(iii) By a similar argument to (ii), if (z^*, w^*, v^*) is a strictly complementary solution of LMCP(M, q, B), then $(x^*, y^*) := T(z^*, w^*, v^*)$ is a solution to LCP$(M, -h)$. It is easy to check from Assumption 4.3.3 and the definition of T that $x_i^* = 0$ implies $y_i^* > 0$, so that (x^*, y^*) is strictly complementary. □

We can now state the following convergence theorems for the LMCP algorithm.

Theorem 4.3.5 *Under Assumptions 4.3.1–4.3.3, there is a constant* $\delta \in (0, 1)$ *such that*

$$\psi_{k+1} \leq (1 - \delta)\psi_k, \quad k = 0, 1, 2, \ldots,$$

that is, the LMCP algorithm converges globally to a solution of LMCP(M, q, B) at a Q-linear rate.

Proof By construction, $\psi_k = \phi_k$, where ϕ_k is as defined in Wright's algorithm. The result follows from Theorem 4.2.6. □

Theorem 4.3.6 *Under Assumptions 4.3.1–4.3.3, there is an* $\hat{\epsilon} > 0$ *such that if* K *is the smallest positive integer such that* $\psi_K \leq \hat{\epsilon}$*, then*

1. *the LMCP algorithm will take fast steps at iteration* K *and at all subsequent iterations, and*

2. *the sequences* $\{\mu_k\}$ *and* $\{\psi_k\}$ *converge Q-subquadratically to zero.*

Proof Follows directly from the definitions of μ_k and ψ_k and Theorem 4.2.9. □

4.4 Summary

In this chapter, we have extended the convergence results of Zhang and Wright to apply to arbitrary strictly positive starting points. This extension is important because it allows the convergence theory to be applied to cases where the algorithms are warm-started from points not satisfying Zhang's and Wright's restrictions.

The extension also plays an important role in proving the convergence results of the LMCP algorithm presented in Section 4.3. Recall that the restriction imposed by Zhang and Wright on the starting point can easily be satisfied simply by making the starting point (x^0, y^0) large in every component. However, in the analysis of the LMCP algorithm, we defined $(x^0, y^0) = T(z^0, w^0, v^0)$ so that (x^0, y^0) is required to lie in the range of T. In particular, $x_H + y_{\hat{H}} = u_H - l_H$, so increases in the components of x_H must be offset by decreases in the components of $y_{\hat{H}}$. Thus, for the LMCP(M, q, B), it may not be possible to find a starting point that meets the restriction. By removing the restriction from the convergence results, this difficulty is eliminated.

Chapter 5

An Algorithm for Solving Affine Generalized Equations

Given a maximal monotone multifunction $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ and a continuously differentiable function $F : \Omega \supset \mathrm{dom}(T) \to \mathbb{R}^n$, the generalized equation is to

$$\mathrm{GE}(F,T): \qquad \text{find } z \in \mathrm{dom}(T) \text{ such that } 0 \in F(z) + T(z).$$

If F is an affine function, and if T is polyhedral, then we get the *affine generalized equation*.

As was mentioned in the introduction, complementarity problems and finite dimensional variational inequalities are special cases of the generalized equation. To date, most of the algorithmic development for generalized equations has been focused on the special case where $T := N_C$, the normal cone to a convex set C. A powerful tool for addressing this special case is the normal map (Robinson 1992) given by

$$F_C(x) := F(\pi_C(x)) + x - \pi_C(x). \tag{42}$$

In fact, the normal map is the basis for MILES (Rutherford 1993) and PATH (Dirkse & Ferris 1995*b*), two of the most successful algorithms available for solving nonlinear MCPs.

In this chapter, we use the algorithmic framework of Eaves (1976) to generate an algorithm for determining zeros of coherently oriented piecewise affine maps. As special cases of this algorithm, we obtain the affine variational inequality algorithm of Cao & Ferris (1995*b*) and an algorithm for solving affine generalized equations. To demonstrate these special cases, we introduce the T-map, a generalization of the normal map, which can be used to solve generalized equations involving operators T that do not necessarily correspond to the normal cone to any set.

In Section 5.1, we present a path following algorithm for piecewise affine maps and prove that under the assumption of coherent orientation, the algorithm finds a zero after a finite number of steps. In Section 5.2, we define the T-map for general F and T and prove several useful properties regarding this map. The remainder of the chapter focuses on the case where T is polyhedral and can be viewed as an extension of the special case where $T := N_C$, with C polyhedral. In particular, it is shown that the previously described algorithm is applicable to this case, and generates an algorithm for affine generalized equations. Section 5.4 addresses the special case where T is separable. Finally, in Section 5.5 we discuss how the algorithm can be applied to the piecewise linear-quadratic programming problem.

5.1 Algorithm for Finding Zeros of $\mathcal{M}$-PA Maps

The theoretical basis for the algorithm of this section is derived from the theory of piecewise-linear homotopies given in Eaves (1976). Our motivation was to generalize the algorithm for solving affine variational inequalities over convex polyhedral sets that was described in Cao & Ferris (1995*b*).

In order to describe the algorithm carefully, some preliminary definitions are required to set up our framework.

Definition 5.1.1 (cell) *A polyhedral convex set* $\sigma \subset \mathbb{R}^n$ *is called a* cell*. If* $\dim(\sigma) = k$ *then* σ *is called a* k-cell*. Let* $\sigma := \{x | Ax \leq a\}$*, where* $A \in \mathbb{R}^{p \times n}$*, and* $a \in \mathbb{R}^p$*, with* p *a nonnegative integer. Then* (p, A, a) *is said to represent* σ*. If* p *is the smallest number for which a representation of* σ *exists, then* (p, A, a) *is called a* minimal representation *of* σ*. A set* $\tau \in \mathbb{R}^n$ *is called a* face *of* σ *if for some set of indices* $\alpha \subset \{1, \ldots, p\}$*,* $\tau = \{x \in \sigma : A_{\alpha\cdot}x = b_\alpha\}$*. If* $\dim(\tau) = i$*, then* τ *is called an* i*-face of* σ*.*

Clearly any cell has a minimal representation.

Definition 5.1.2 (piecewise affine) *Let* $\mathcal{M}$ *be a collection of* n*-cells and let* $M := \bigcup_{\sigma \in \mathcal{M}} \sigma$*. A function* $F : M \to \mathbb{R}^m$ *is said to be* piecewise-affine *with respect to* $\mathcal{M}$*, denoted* $\mathcal{M}$-PA*, if for each* $\sigma \in \mathcal{M}$*,* $F_{|\sigma}$ *(i.e. the restriction of* F *to* σ*) is affine. If* F *is* $\mathcal{M}$*-PA for some* $\mathcal{M}$ *satisfying the above assumptions, then we say that* F *is piecewise affine.*

Note that in the above definition, if M is convex, then the function F must be continuous on M. Furthermore, in contrast to the work of Eaves (1976), $\mathcal{M}$ is not required to correspond to a subdivided manifold.

Definition 5.1.3 (function representation) *Let $\mathcal{M}$ be a collection of n-cells in R^n, let F be a $\mathcal{M}$-PA function, and let σ be an n-cell of $\mathcal{M}$. Let $b^\sigma \in R^m$ and let B^σ be an $m \times n$ matrix. (B^σ, b^σ) is said to* represent F *on σ if $F(x) = B^\sigma x + b^\sigma$ for all $x \in \sigma$.*

We now describe an algorithm to find a zero of an $\mathcal{M}$-PA function G, for a given collection of cells $\mathcal{M}$. We will assume that representations of the cells of $\mathcal{M}$ and of the map G have already been constructed. The basis of the algorithm is to construct a piecewise affine homotopy mapping $F(x, \mu)$ with the following properties

1. $(x^*, 0)$ is a zero of F if and only if x^* is a zero of G.

2. A point (x^1, μ_1), and a direction $(d^1, -1)$ is known such that $\mu_1 \geq 0$ and $F(x^1 - \mu d^1, \mu^1 + \mu) = 0$ for all $\mu \geq 0$.

The algorithm uses a method described by Eaves (1976) to trace the zero curve of F, proceeding in the direction $(d^1, -1)$ from the starting point (x^1, μ_1). To prove that the algorithm finds a solution in a finite number of steps, we restrict ourselves to the case where G is *coherently oriented*:

Definition 5.1.4 (coherent orientation) *Let G be an $\mathcal{M}$-PA map with representation (B^σ, b^σ) on each $\sigma \in \mathcal{M}$. We say that G is* coherently oriented *if*

$$sgn(\det(B^\sigma))$$

is nonzero and constant for all σ in $\mathcal{M}$, where

$$sgn(x) := \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0. \end{cases}$$

Since $\mathcal{M}$ is finite and $\bigcup_{\sigma \in \mathcal{M}} \sigma = \mathrm{R}^n$, it follows that $\mathrm{R}^n = \bigcup_{\sigma \in \mathcal{M}} \operatorname{rec}(\sigma)$, and further that there is a σ such that $\operatorname{int}(\operatorname{rec}(\sigma)) \neq \emptyset$. Choose d such that $-d \in \operatorname{int}(\operatorname{rec}(\sigma))$. Then for any x^0 in R^n, and for all μ sufficiently large, $x^0 - \mu d \in \operatorname{int}(\sigma)$.

In the AVI algorithm described by Cao and Ferris, the cell σ and the direction d were constructed by finding an extreme point x^e of the set C. The cell was then given by $\sigma := x^e + N_C(x^e)$, and the direction d was chosen such that $-d$ was in the interior of $N_C(x^e)$. For our algorithm, rather than constructing the cell and direction, we can rely instead on the fact that since $\mathrm{R}^n = \bigcup_{\sigma\in\mathcal{M}} \mathrm{rec}(\sigma)$, then for any direction d, there will be a cell σ_d for which $-d \in \mathrm{rec}(\sigma_d)$. Note further that for each cell σ, the boundary $\mathrm{rec}(\sigma)\setminus\mathrm{int}(\mathrm{rec}(\sigma))$ of $\mathrm{rec}(\sigma)$ has Lebesgue measure zero. Therefore, since the number of cells is finite, $\bigcup_{\sigma\in\mathcal{M}} \mathrm{rec}(\sigma)\setminus\mathrm{int}(\mathrm{rec}(\sigma))$ has measure zero. Thus, for almost all d, there will be a cell σ_d for which $-d \in \mathrm{int}(\mathrm{rec}(\sigma_d))$.

Thus, if x^0 is any point in R^n, then for all μ sufficiently large, $x^0 - \mu d$ will lie interior to the cell σ_d. In other words, the cell can be chosen simply by picking an arbitrary d and proceeding in the direction $-d$ until a cell σ_d is reached for which $-d$ is in the recession cone of σ_d. For almost all d, $-d$ will be in the *interior* of $\mathrm{rec}(\sigma_d)$. We note, however, that for some special cases, construction of an extreme point may still be preferable.

Once d and σ_d have been identified, the homotopy map can be constructed. Let (B, b) be the representation of G in σ_d. Define a function $F : \mathrm{R}^n \times \mathrm{R}_+ \to \mathrm{R}^n$ by

$$F(x,\mu) := G(x) + \mu B d \tag{43}$$

Note that $F(x, 0) = 0$ exactly when $G(x) = 0$. Under the assumption that G is coherently oriented, B is invertible. Let $x^0 := -B^{-1}b$ and define

$$w(\mu) = x^0 - \mu d.$$

Then, since $-d \in \mathrm{int}(\mathrm{rec}(\sigma_d)))$, there exists $\mu_0 \geq 0$ such that $w(\mu) \in \mathrm{int}(\sigma_d), \forall \mu > \mu_0$. Thus, for $\mu \geq \mu_0$,

$$\begin{aligned} F(w(\mu),\mu) &= G(w(\mu)) + \mu B d \\ &= B w(\mu) + b + \mu B d \\ &= B(x^0 - \mu d) + b + \mu B d \\ &= -b - \mu B d + b + \mu B d \\ &= 0. \end{aligned} \tag{44}$$

By choosing $\mu_1 > \mu_0$, $x^1 = w(\mu_1)$, and $d^1 = d$, we see that F satisfies the conditions needed for the homotopy map. We are now ready to state the algorithm, which is given in Figure 13. Note that by normalizing d in the discussion above to be a unit vector, we can start the algorithm from the point (x^1, μ_1) constructed above with $\sigma_1 := \sigma_d$.

Some comments about Algorithm AGE are in order:

1. Most of the work in the algorithm is in step 8 where the direction (d^{k+1}, v_{k+1}) is calculated. At the end of this section in Theorem 5.1.13 we show that $B^{k+1} - B^k$ is a rank-1 matrix. Thus, an efficient implementation of the algorithm can be obtained by keeping the B matrices in factored form and performing rank-1 updates of the factors at each step of the algorithm.

Figure 13: Algorithm AGE

Given a finite collection of n-cells $\mathcal{M}$ such that $\bigcup_{\sigma \in \mathcal{M}} \sigma = \mathrm{R}^n$, and an $\mathcal{M}$-PA function G on R^n. Let G have representation (B^k, b^k) on $\sigma_k \in \mathcal{M}$.

1) **[Initialization]** Determine $(x^1, \mu_1, \sigma_1, d^1)$ satisfying
   $\|d^1\| = 1$,
   $x^1 \in \mathrm{int}(\sigma_1)$,
   $x^1 - \mu d^1 \in \mathrm{int}(\sigma_1), \quad \forall \mu \geq 0$,
   $B^1 x^1 + \mu_1 B^1 d^1 + b^1 = 0$.

2) $v_1 := -1$.

Repeat for $k = 1, 2, \ldots$

3) $\theta_k := \sup\{\theta | x^k + \theta d^k \in \sigma_k, \mu_k + \theta v_k \geq 0\}$.
   If $\theta_k = +\infty$, then
   4) **output**("ray termination"); **return**.
   Else
   5) $x^{k+1} := x^k + \theta_k d^k$
   6) $\mu_{k+1} := \mu_k + \theta_k v_k$
      If $\mu_{k+1} = 0$ then
      7) **output**("solution found at", x^{k+1}); **return**.
      Else
      8) determine σ_{k+1} (possibly using lexicographic ordering), d^{k+1}, and v_{k+1} such that
         $x^{k+1} \in \sigma_{k+1}$,
         $B^{k+1} d^{k+1} + v_{k+1} B^1 d^1 = 0$,
         $\left\| d^{k+1} \right\| = 1$,
         d^{k+1} points into σ_{k+1} from x^{k+1},
         and $\sigma_{k+1} \in \mathcal{M} \setminus \sigma_k$.
      9) goto next k.

2. At step 8 in the algorithm, there may be more than one possible choice of cells σ_{k+1}. However, a lexicographic ordering, as described by Eaves (1976, Section 15), can be used to resolve any ambiguity concerning which cell to choose. The use of such a lexicographic ordering will be assumed in the convergence proof, and will be presented in more detail in the discussion preceding Lemma 5.1.9.

3. The requirement that $\left\|d^{k+1}\right\| = 1$ is arbitrarily chosen to force the choice of d^{k+1} to be unique.

4. The requirement that $x^1 - \mu d^1 \in \operatorname{int}(\sigma_1)$, $\forall \mu \geq 0$ guarantees that the zero curve of $F(x,\mu) := G(x) + \mu B d^1$ contains a ray, and therefore assures us that it will not have any loops. This fact will be useful in our convergence proof. However, we shall also show that, under the assumption of coherent orientation, v_k is always negative, which by itself guarantees that no loops occur. Thus, under the assumption of coherent orientation, it is not necessary to find a ray start. However, in future work, we will prove convergence for a broader class of problems, in which case the ray start requirement will be useful.

The next few pages are devoted to proving the following convergence theorem:

Theorem 5.1.5 *Let $\mathcal{M}$ be a finite collection of n-cells whose relative interiors are disjoint and whose union is R^n. Let G be a coherently oriented, $\mathcal{M}$-PA function. Algorithm AGE, using lexicographic ordering, terminates after finitely many steps with a zero x^* of G.*

Proof (Outline) There are three main parts to the proof. First, we will show that the algorithm terminates at a solution if $\mathcal{M}$ is a *subdivision* of R^n (see Definition 5.1.6). This result is given in Lemma 5.1.9. Second, we will show that even if $\mathcal{M}$ is not a subdivision of R^n, there is a *refinement* (see Definition 5.1.10) $\mathcal{N}$ of $\mathcal{M}$ that is a subdivision. This result is given in Lemma 5.1.11. Finally, we show in Lemma 5.1.12 that if a subdivision $\mathcal{N}$ is a refinement of $\mathcal{M}$, then running the algorithm using $\mathcal{N}$ will generate exactly the same path as would be generated by using $\mathcal{M}$. Thus, the fact that the algorithm terminates at a solution using $\mathcal{N}$ guarantees that it will terminate at a solution using $\mathcal{M}$. $\square$

We now set about proving the three lemmas mentioned above. At this point, we recommend that the impatient reader skip ahead to Theorem 5.1.13.

Our proof technique is based on the work of Eaves (1976). Eaves' analysis relies heavily on the notion of a subdivided manifold:

Definition 5.1.6 (subdivided manifold) *Let N be a set in some Euclidean space, and let $\mathcal{N}$ be a finite or countable collection of n-cells in that space such that $N = \bigcup_{\sigma \in \mathcal{N}} \sigma$. Let $\tilde{\mathcal{N}}$ be the collection of all faces of elements of $\mathcal{N}$. $(N, \tilde{\mathcal{N}})$ is a* subdivided n-manifold *if*

1. *any two n-cells of $\mathcal{N}$ are either disjoint or meet in a common face;*
2. *each point of N has a neighborhood meeting only finitely many n-cells of $\mathcal{N}$;*
3. *each $(n-1)$-cell of $\tilde{\mathcal{N}}$ lies in at most two n-cells;*

If $(N, \tilde{\mathcal{N}})$ is a subdivided n-manifold for some subdivision $\mathcal{N}$, we call N an n-manifold *and we call $\mathcal{N}$ a* subdivision *of N.*

The following lemma shows that when $N = \mathrm{R}^n$, item 3 in Definition 5.1.6 is redundant. This result was proved by Robinson (1992) in the proof of Proposition 2.4. While Robinson's proposition is stated for the normal manifold, his proof is valid for general subdivisions of R^n.

Lemma 5.1.7 *If $\mathcal{N}$ is a collection of cells whose union is R^n and if $\mathcal{N}$ satisfies 1 and 2 of Definition 5.1.6, then $\mathcal{N}$ is a subdivision of R^n.*

The next step in our analysis is to prove that the algorithm works whenever $\mathcal{M}$ is a subdivision of R^n. In this case, by defining $\mathcal{S} := \{\sigma \times \mathrm{R}_+ | \sigma \in \mathcal{M}\}$, we see that $\mathcal{S}$ is a subdivision of $\mathrm{R}^n \times \mathrm{R}_+$ and further that F is $\mathcal{S}$-PA. The starting point (x^1, μ_1) of the algorithm lies interior to the cell $\eta_1 := \sigma_1 \times \mathrm{R}_+$ of $\mathcal{S}$. Further, the ray $\{(x^1, \mu_1) - \mu(d^1, -1) | \mu \geq 0\}$ lies within η_1. Let $\tilde{\mathcal{S}}$ be the collection of all faces of elements of $\mathcal{S}$. Algorithm AGE is then seen to be equivalent to the algorithm described by Eaves (1976, Section 10.2), with the following relationships between the algorithms:

Eaves' Algorithm	Algorithm AGE
$\mathcal{M}$	$\tilde{\mathcal{S}}$
$F(x)$	$F(x, \mu) := G(x) + \mu B^1 d^1$
x_k	(x^k, μ_k)
σ_k	η_k
v_k	(d^k, v_k)

To discuss the behavior of this algorithm in more detail, we need some definitions from Eaves (1976).

Definition 5.1.8 (regularity) *Let* $(N, \tilde{\mathcal{N}})$ *be a subdivided* $(n+1)$*-manifold, let* $\mathcal{N}$ *be the collection of* n*-cells in* $\tilde{\mathcal{N}}$*, and let* $F : N \to \boldsymbol{R}^n$ *be a* $\mathcal{N}$*-PA map. A point* x *in* N *is said to be* degenerate *(otherwise* regular*) if* x *lies in a cell* σ *of* $\tilde{\mathcal{N}}$ *with* $\dim(F(\sigma)) < n$*. A value* y *in* $F(N)$ *is said to be a* degenerate value *(otherwise a* regular value*) if* $F^{-1}(y)$ *contains a degenerate point.*

Note that if y is a regular value, then $F^{-1}(y)$ cannot intersect any k-cells of $\mathcal{S}$ with $k < n$.

By the assumption of coherent orientation, G is one-to-one in every n-cell of $\mathcal{M}$. Thus, $\dim(F(\eta)) = n$ for all $(n+1)$-cells η of $\mathcal{S}$. Since the starting point (x^1, μ_1) of the algorithm is interior to η_1, it is a regular point of F. According to Eaves (1976, Theorem 15.13), since $\mathcal{S}$ is finite, the algorithm generates, in finitely many steps, either a point (x^*, μ_*) in the boundary of $\mathrm{R}^n \times \mathrm{R}_+$, or a ray in $F^{-1}(0)$ different from the starting ray. In the first case, we know that $\mu_* = 0$, since the boundary of $\mathrm{R}^n \times \mathrm{R}_+$ is $\mathrm{R}^n \times \{0\}$. It then follows, from our earlier remarks that x^* satisfies $G(x^*) = 0$. Therefore, to guarantee that the algorithm finds a solution, we need only show that it cannot produce a ray different from the starting ray.

We first consider the case when 0 is a *regular value* of F. In this case, by Eaves (1976, Theorem 9.1), $F^{-1}(0)$ is a 1-manifold which is subdivided by sets of the form $\eta \bigcap F^{-1}(0)$. Further, since $F^{-1}(0)$ cannot intersect any k-cells with $k < n$, each point on $F^{-1}(0)$ is in at most two $(n+1)$-cells of $\mathcal{S}$. Thus, in step 8 of the algorithm, the choice of σ_{k+1} is well-defined. (The only difficulty would be if (x^{k+1}, μ_{k+1}) lies in only one $(n+1)$-cell η_k so that no σ_{k+1} could be selected. But in this case, (x^{k+1}, μ_{k+1}) would be a boundary point of $\mathrm{R}^n \times \mathrm{R}_+$. Thus, $\mu_{k+1} = 0$, so the algorithm would have terminated in step 5.)

Let (d^k, v_k) be the direction of the path within the $(n+1)$-cell η_k of $\mathcal{S}$, and let G have representation (B^k, b^k) on the n-cell σ_k of $\mathcal{M}$. Then by Eaves (1976, Lemma 12.3), the *curve index*, given by

$$(\mathrm{sgn} v_k)(\mathrm{sgn}\det B^k)$$

is constant everywhere along the path. Since $v_1 = -1$ for the starting direction (d^1, v_1), and since G is coherently oriented, it follows that v_k is negative in each cell that the path enters. But this means that the parameter μ decreases strictly in each cell. Thus, after finitely many steps, we must have $\mu = 0$.

When 0 is a degenerate value of F, $F^{-1}(0)$ may intersect a k-cell of $\mathcal{S}$ with $k < n$. Thus, in step 8 of the algorithm, there may be multiple choices for which cell σ_{k+1} to enter next. To address this problem, a lexicographic ordering can be used to resolve ambiguities concerning which cell the path will enter. Such a scheme is conceptually equivalent to solving a perturbed problem, which we now describe.

Let $X = [\xi^1, \ldots, \xi^n]$ be an $(n+1) \times n$ matrix such that $[X, d^1]$ is of rank $(n+1)$. Define the vector $[\epsilon] := (\epsilon^1, \epsilon^2, \ldots, \epsilon^n)^\top$ (note: the superscripts here refer to exponentiation). Define $(x^1(\epsilon); \mu_1(\epsilon)) := (x^1; \mu_1) + X[\epsilon]$. Since $(x^1; \mu_1)$ in Algorithm AGE is interior to η_1, then $(x^1(\epsilon); \mu_1(\epsilon)) \in \text{int}(\eta_1)$ for small enough ϵ. Further, since $(-d^1, 1) \in \text{int}(\text{rec}(\eta_1))$, $(x^1(\epsilon), \mu_1(\epsilon)) + \mu(-d^1, 1) \in \text{int}(\eta_1)$, for all $\mu \geq 0$. Thus, $x^1(\epsilon), \mu_1(\epsilon), \sigma_1$, and d^1 satisfy the starting conditions needed to apply the algorithm to the perturbed problem given by

$$0 = F(x, \mu) - p(\epsilon),$$

where $p(\epsilon) := F(x^1(\epsilon), \mu_1(\epsilon))$. Observe that

$$\begin{aligned} p(\epsilon) &= F(x^1, \mu_1) + [B^1, B^1 d^1]X[\epsilon] \\ &= [B^1, B^1 d^1]X[\epsilon] \\ &= Y[\epsilon], \end{aligned} \tag{45}$$

where $Y := [B^1, B^1 d^1]X$. Y is an invertible $n \times n$ matrix, so that by Eaves (1976, Lemma 14.2), $p(\epsilon)$ is a regular value for all ϵ sufficiently small. Thus, by the arguments given above for regular values, using Algorithm AGE to solve the perturbed problem will, after a finite number of steps J, produce a point $(x^J(\epsilon))$ such that $G(x^J(\epsilon)) = p(\epsilon)$.

Let $(x^k(\epsilon), \mu_k(\epsilon))$ be the sequence of points generated by the algorithm for the perturbed problem. By the discussion in Eaves (1976, Section 15), there is a sequence of matrices $X^k \in \mathrm{R}^{(n+1)\times n}$ and a sequence of points (x^k, μ_k) such that $(x^k(\epsilon); \mu_k(\epsilon)) := (x^k; \mu_k) + X^k[\epsilon]$ for all small ϵ. The points (x^k, μ_k) are exactly the sequence of points generated by the algorithm for solving the unperturbed problem using the lexicographic ordering. Since the algorithm terminates after J steps for all small ϵ, we see that $\mu_J(\epsilon) = 0$ and $G(x^J(\epsilon)) = p(\epsilon)$. It follows that $\mu_J = 0$ and further that $G(x^J) = 0$. Thus, using a lexicographic ordering, the algorithm finds a solution after a finite number of steps.

We have proved the following lemma:

Lemma 5.1.9 *Let $\mathcal{M}$ be a subdivision of $\boldsymbol{R}^n$ and A_T be a coherently oriented, $\mathcal{M}$-PA function. Algorithm AGE, using lexicographic ordering, terminates after finitely many steps with a zero x^* of G.*

We now address the case where $\mathcal{M}$ is not a subdivision of R^n. We begin by proving that $\mathcal{M}$ can be *refined* to produce a subdivision.

Definition 5.1.10 (refinement) *Let $\mathcal{M}$ and $\mathcal{N}$ be finite collections of n-cells. $\mathcal{N}$ is said to be a* refinement *of $\mathcal{M}$ if each cell σ of $\mathcal{M}$ is the union of a finite collection of cells τ_i of $\mathcal{N}$, and if each cell of $\mathcal{N}$ is contained in some cell of $\mathcal{M}$.*

The following lemma is proved by Hudson (1969, Lemma 1.5), however, using different nomenclature. In particular, the term "subdivision" is used in place of our term "refinement".

Lemma 5.1.11 *Let $\mathcal{M} := \{C_i\}$ be a collection of $J < \infty$ n-cells which covers $\boldsymbol{R}^n$. There exists a subdivision $\mathcal{N}$ of $\boldsymbol{R}^n$ such that $\mathcal{N}$ is a refinement of $\mathcal{M}$.*

We now show that using $\mathcal{N}$, the algorithm follows the same path as it would by using $\mathcal{M}$.

Lemma 5.1.12 *Let G be an $\mathcal{M}$-PA function, where $\mathcal{M}$ is a finite collection of relatively disjoint n-cells whose union is $\boldsymbol{R}^n$. Let $\mathcal{N}$ be a refinement of $\mathcal{M}$ such that $\mathcal{N}$ is a subdivision of $\boldsymbol{R}^n$. Then Algorithm AGE, using lexicographic ordering, will find a solution x^* to $G(x) = 0$ in a finite number of steps. Furthermore, the sequence of points generated by the algorithm using $\mathcal{M}$ is a subsequence of the points that would be generated using $\mathcal{N}$.*

Proof Consider first running the algorithm using $\mathcal{N}$ instead of $\mathcal{M}$. By Lemma 5.1.9, the algorithm will terminate after some finite number of steps J. The algorithm will visit a sequence of n-cells $\{\tau_k\} \subset \mathcal{N}$, and will generate a sequence of points $\{(x^k, \mu_k)\}$ and directions $\{(d^k, v_k)\}$, for $k = 1, \ldots, J$.

Let $j_1 := 1$ and let σ_1 be the unique cell in $\mathcal{M}$ that contains τ_1. Then for $i = 2, \ldots$, let j_i be the smallest index greater than j_{i-1} such that $\tau_{j_i} \not\subset \sigma_{i-1}$, and if $j_i \leq J$, let σ_i be the unique cell in $\mathcal{M}$ that contains τ_{j_i}. Let K be such that $j_K = J$. This process defines a sequence of cells $\{\sigma_i\}, i = 1, \ldots, K$ and indices $\{j_i\}, i = 1, \ldots, K+1$ such that $\tau_k \subset \sigma_i$ whenever $j_i \leq k < j_{i+1}$.

We will show that if the algorithm is run using $\mathcal{M}$, then the sequence of points $\{(\xi^i, \nu_i)\}$ generated by the algorithm satisfies the equation $(\xi^i, \nu_i) = (x^{j_i}, \mu_{j_i})$, for each i. Thus, $(\xi^K, \nu_K) = (x^J, \mu_J)$, so the algorithm finds a solution after a finite number of steps.

Let $\{(\delta^i, \zeta_i)\}$ be the sequence of directions chosen by the algorithm using $\mathcal{M}$. Clearly, since the algorithm is started at the point (x^1, μ_1) in the direction $(d^1, -1)$, the following is true: $(\xi^1, \nu_1) = (x^{j_1}, \nu_{j_1})$, $(\delta^1, \zeta_1) = (d^{j_1}, v_{j_1})$, and the first cell visited by the algorithm is σ_1.

We now proceed by induction: Assume that $(\xi^i, \nu_i) = (x^{j_i}, \nu_{j_i})$, $(\delta^i, \zeta_i) = (d^{j_i}, v_{j_i})$, and that, using $\mathcal{M}$, the ith cell visited by the algorithm is σ_i. We shall prove that $(\xi^{i+1}, \nu_{i+1}) = (x^{j_{i+1}}, \nu_{j_{i+1}})$, $(\delta^{i+1}, \zeta_{i+1}) = (d^{j_{i+1}}, v_{j_{i+1}})$, and that the $(i+1)$st cell visited by the algorithm is σ_{i+1}.

Let (B^i, b^i) be the representation of G on σ_i. This is also the representation of G on τ_k whenever $j_i \leq k < j_{i+1}$. Thus, in step 8 of the algorithm using $\mathcal{N}$, the direction (d^{k+1}, v_{k+1}) chosen when entering cell τ_{k+1} must satisfy

$$B^i d^{k+1} + v_{k+1} B^1 d^1 = 0$$

for $j_i \leq k < j_{i+1}$. Since G is coherently oriented, B^i is invertible. Further, $\left\| d^{k+1} \right\| = 1$ and, by our earlier discussion, v_{k+1} is negative. Thus, the direction is uniquely determined by the representation. In particular, $(d^{j_i}, v_{j_i}) = (d^{j_i+1}, v_{j_i+1}) = \cdots = (d^{j_{i+1}}, v_{j_{i+1}}) = (\delta^i, \zeta_i)$. From this it is clear that $x^{j_{i+1}}$ lies on the ray $\{\xi^i(\theta) | \theta \geq 0\}$, where $\xi^i(\theta) := \xi^i + \theta\delta^i$. Further, $x^{j_{i+1}}$ is on the boundary of σ_i.

If the ray $\{\xi^i(\theta) | \theta \geq 0\}$ contains a point in the interior of σ_i, then the ray cannot be extended past $x^{j_{i+1}}$ without exiting σ_i. Thus, $x^{j_{i+1}} = \xi^i(\theta_i)$ where $\theta_i := \sup\{\theta | \xi^i(\theta) \in \sigma_i\}$. In other words, $(x^{j_{i+1}}, \mu_{j_{i+1}}) = (\xi^{i+1}, \nu_{i+1})$.

If the ray $\{\xi^i(\theta) | \theta \geq 0\}$ does not contain an interior point of σ_i, then we must resort to the lexicographic ordering to prove that $x^{j_{i+1}} = \xi^i(\theta_i)$. Since σ_i and $\tau_{j_{i+1}}$ are relatively disjoint convex sets, there exists a separating hyperplane H_i defined by a vector c^i, and a scalar α_i such that ${c^i}^\top x < \alpha_i$, $\forall x \in \operatorname{int}(\sigma_i)$, and ${c^i}^\top x \geq \alpha_i$, $\forall x \in \tau_{j_{i+1}}$. Suppose we run the algorithm using $\mathcal{N}$ to solve the perturbed problem $G_\epsilon(x) := G(x) - p(\epsilon) = 0$, where $p(\epsilon)$ is defined by (45). Then, for ϵ small enough, the algorithm will visit the same sequence of cells $\{\tau_k\}$ as it visits in the unperturbed problem. Also, by our earlier discussion, the algorithm will generate the sequence of points $\{(x^k(\epsilon); \mu_k(\epsilon))\} = \{(x^k; \mu_k) + X^k[\epsilon]\}$, where $\{X^k\}$ is a fixed sequence of matrices.

Since 0 is a regular value of G_ϵ, $\dim(G_\epsilon(\tau_k \bigcap \tau_{k+1})) \geq n - 1$ for any k. Thus, $G_\epsilon^{-1}(0)$ contains only one point in $\tau_k \bigcap \tau_{k+1}$, namely x^{k+1}. Therefore, the direction d^{k+1} must point into the interior of τ_{k+1}.

By similar arguments as before, $x^{j_{i+1}}$ lies on the ray $\{\xi^i_\epsilon(\theta) | \theta \geq 0\}$, where $\xi^i_\epsilon(\theta) := x^{j_i}(\epsilon) + \theta(d^{j_i})$. But, since d^{k+1} points into the interior of τ_{j_i}, this ray must contain a point $\hat{x}$ in the interior of σ_i. Thus, ${c^i}^\top \hat{x} < \alpha_i$. But ${c^i}^\top x^{j_{i+1}} \geq \alpha_i$ since $x^{j_{i+1}}$ is in $\tau_{j_{i+1}}$. It follows that ${c^i}^\top d^{j_i} > 0$. Thus, even for the unperturbed problem, the ray $\xi^i(\theta)$ cannot be extended past the point $x^{j_{i+1}}$ without crossing the hyperplane H_i, and thereby exiting σ_i. Thus, $x^{j_{i+1}} = \xi^i(\theta_i)$, and as before, $(x^{j_{i+1}}, \mu_{j_{i+1}}) = (\xi^{i+1}, \nu_{i+1})$.

Finally, note that for all small ϵ, the point $x^{j_{i+1}}(\epsilon)$ is a regular point, so $\tau_{(j_{i+1}-1)}$ and $\tau_{j_{i+1}}$ are the only n-cells of $\mathcal{N}$ that contain $x^{j_{i+1}}(\epsilon)$. Thus, σ_i and σ_{i+1} are the only n-cells of $\mathcal{M}$ that contain $x^{j_{i+1}}(\epsilon)$. Thus, for all small ϵ, the algorithm, using $\mathcal{M}$ will enter cell σ_{i+1} at the next iteration. But this means that using lexicographic ordering the algorithm will enter cell σ_{i+1} next when solving the unperturbed problem. Finally, since the representation of G on σ_{i+1} is identical to the representation of G on $\tau_{j_{i+1}}$, we must have $(\delta^{i+1}, \zeta_{i+1}) = (d^{j_{i+1}}, v^{j_{i+1}})$.

The lemma is now proved by induction. □

This completes the proof of Theorem 5.1.5. Our final task in this section is to establish the claim made in Comment 1 following Algorithm AGE.

Theorem 5.1.13 *Under the hypothesis of Theorem 5.1.5, let $\{\sigma_k\}$ be the sequence of cells chosen in Step 8 of Algorithm AGE using lexicographic ordering, and let (B^k, b^k) represent A_T on σ_k. Then $B^{k+1} - B^k$ has rank 1.*

Proof Using lexicographical ordering, the algorithm will choose the same cell σ_{k+1} in step 8 as it would when solving the perturbed problem for small ϵ. However, 0 is a regular value for the perturbed problem, so $\tau_k := \sigma_k \bigcap \sigma_{k+1}$ must have dimension $n-1$. Now, for any two points $x^1, x^2 \in \tau_k$

$$B^{k+1}(x^1 - x^2) = B^k(x^1 - x^2) \Longrightarrow (x^1 - x^2) \in \ker(B^{k+1} - B^k)$$

Thus, $\dim\ker(B^{k+1} - B^k) = n - 1$ so $\operatorname{rank}(B^{k+1} - B^k) = 1$. □

5.2 The T-map

The T-map, denoted F_T, is a generalization of the normal map that is formed by replacing the projection operator π_C in (42) by the resolvent operator $P_T := (I+T)^{-1}$. Specifically, the T-map is given by

$$F_T(x) := F(P_T(x)) + x - P_T(x). \tag{46}$$

Minty (1962) showed that P_T is a continuous, single-valued, nonexpansive function defined on all of R^n. Since the image of P_T is $\mathrm{dom}(T)$, it follows that F_T is a single-valued function defined on all of R^n.

By Brézis (1973, Example 2.1.2), $I + T$ is monotone, and therefore P_T is monotone. We now show that solving $\mathrm{GE}(F, T)$ is equivalent to finding a zero of F_T.

Theorem 5.2.1 *Given a maximal monotone multifunction $T : \boldsymbol{R}^n \rightrightarrows \boldsymbol{R}^n$, and a function $F : \Omega \subset R^n \to \boldsymbol{R}^n$, let F_T be defined by (46). If x is a zero of F_T, then $z := P_T(x)$ solves $GE(F,T)$. Conversely, if z solves $GE(F,T)$, then $x := z - F(z)$ is a zero of F_T.*

Proof Suppose $F_T(x) = 0$ and let $z := P_T(x)$. Then $0 = F_T(x) = F(z) + x - z$, and

$$\begin{aligned} -F(z) &= x - z \\ &\in (I+T)(I+T)^{-1}(x) - z \\ &= (I+T)(z) - z \\ &= T(z). \end{aligned}$$

Conversely, suppose $-F(z) \in T(z)$ and let $x := z - F(z)$. Then $x \in z + T(z) = (I + T)(z)$, so $P_T(x) = (I+T)^{-1}(x) = z$. Thus $F_T(x) = F(z) + x - z = F(z) - F(z) = 0$.
$\square$

So far, we have not made any assumptions on T other than that it is maximal monotone. We now focus on the case where T is polyhedral.

Definition 5.2.2 (polyhedral) *A multifunction T is* polyhedral *if its graph is the union of finitely many polyhedral convex sets.*

Our first task will be to show that, for polyhedral T, the resolvent operator $P_T := (I + T)^{-1}$ is a piecewise-affine map.

Lemma 5.2.3 *A single-valued multifunction $T : \boldsymbol{R}^n \rightrightarrows \boldsymbol{R}^m$ whose graph is a convex polyhedron is affine on dom(T).*

Proof Assume $\mathrm{dom}(T) \neq \emptyset$. (Otherwise the lemma is true vacuously). Since the graph of T is a polyhedron, T can be written as $T = \{(x, y) | Ax + By \geq c\}$ for some $A \in \mathrm{R}^{p \times n}, B \in \mathrm{R}^{p \times m}$, and $c \in \mathrm{R}^p$, where p is some nonnegative integer. Let $\mathcal{K} := \{i | A_{i\cdot}x + B_{i\cdot}y = c_i, \forall (x, y) \in T\}$. In words, $\mathcal{K}$ is the set of row indices for which the corresponding constraint is active for *all* points in T.

We first establish the fact that $\ker B_{\mathcal{K}\cdot} = \{0\}$. To do this, let $\mathcal{H} := \{i | i \notin \mathcal{K}\}$. Then for each $i \in \mathcal{H}, \exists (x^i, y^i) \in T$ with $A_{i\cdot}x^i + B_{i\cdot}y^i > c_i$. Let $(\tilde{x}, \tilde{y}) = \sum_{i \in \mathcal{H}} (x^i, y^i)/|\mathcal{H}|$, where $|\mathcal{H}|$ is the cardinality of the index set $\mathcal{H}$. Note that $(\tilde{x}, \tilde{y})$ is a convex combination of points in T and is therefore also in T. Further, $A_{i\cdot}\tilde{x} + B_{i\cdot}\tilde{y} > c_i, \forall i \in \mathcal{H}$. Now, if $\hat{y} \in \ker B_{\mathcal{K}\cdot}$, then for $\epsilon > 0$ small enough, $A\tilde{x} + B(\tilde{y} + \epsilon\hat{y}) \geq c$. Thus, $(\tilde{x}, \tilde{y} + \epsilon\hat{y}) \in T$. But since T is single-valued, $\hat{y} = 0$. Thus, $\ker B_{\mathcal{K}\cdot} = \{0\}$.

Now, by the definition of $\mathcal{K}$, we have $(x, y) \in T \Rightarrow A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}}$. Conversely, suppose (x, y) satisfies $A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}}$. If $x \in \mathrm{dom}(T)$, then $\exists \hat{y}$ such that $(x, \hat{y}) \in T$. But this means that $A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}\hat{y} = c_{\mathcal{K}}$, which implies that $\hat{y} - y \in \ker B_{\mathcal{K}\cdot} = \{0\}$. That is $\hat{y} = y$. We have thus shown that

$$(x, y) \in T \Leftrightarrow x \in \mathrm{dom}(T) \text{ and } A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}}.$$

Finally, since $\ker B_{\mathcal{K}\cdot} = \{0\}$, $B_{\mathcal{K}\cdot}$ has a left inverse $R \in \mathrm{R}^{m \times p}$. Thus, for $x \in \mathrm{dom}(T)$,

$$\begin{aligned} (x, y) \in T &\Leftrightarrow A_{\mathcal{K}\cdot}x + B_{\mathcal{K}\cdot}y = c_{\mathcal{K}} \\ &\Leftrightarrow y = Rc_{\mathcal{K}} - RA_{\mathcal{K}\cdot}x. \end{aligned}$$

So T is an affine function on dom(T). $\square$

Theorem 5.2.4 *Given a maximal monotone polyhedral multifunction* $T : \boldsymbol{R}^n \rightrightarrows \boldsymbol{R}^n$*, the resolvent operator* $P_T := (I+T)^{-1}$ *is a piecewise affine function on all of* $\boldsymbol{R}^n$*.*

Proof Since T is polyhedral, $I + T$ is also polyhedral (Robinson 1979a) and therefore so is $P_T = (I+T)^{-1}$. Thus, $P_T = \bigcup \Gamma_i$, where $\{\Gamma_i\}$ is a finite collection of polyhedral convex sets. Let C_i be the projection of Γ_i onto the domain of P_T (i.e., $C_i = \pi_1(\Gamma_i)$, where $\pi_1 := (x,y) \mapsto x$). Define $\mathcal{M} := \{C_i | \dim(C_i) = n\}$.

Since P_T is defined on all of R^n, $\bigcup C_i = \mathrm{R}^n$. Let $M := \bigcup_{C_i \in \mathcal{M}} C_i$. Since M is closed, its complement, $\backslash M := \mathrm{R}^n \setminus M$, is open. Thus, $\backslash M$ is either the empty set, or it has nonempty interior. But $\backslash M \subset \bigcup_{\dim(C_i)<n} C_i$. Thus, $\backslash M$ has no interior. In other words $\backslash M = \emptyset$ and thus, $M := \mathrm{R}^n$.

To show that P_T is $\mathcal{M}$-PA, all that is needed is to show that for each $C_i \in \mathcal{M}$, the restriction of P_T to C_i is affine. However, since P_T is single-valued, the graph of P_T restricted to C_i is simply the convex polyhedral set Γ_i. By Lemma 5.2.3, P_T is affine on C_i. □

Corollary 5.2.5 *If* T *is polyhedral and* F *is affine, then the* T*-map,* F_T*, defined by (46) is piecewise affine.*

5.3 Affine Generalized Equations

We now show how to apply the algorithm of Section 5.1 to construct an algorithm to solve the *affine generalized equation*:

$$0 \in Ax - a + T(x), \tag{47}$$

where $A \in \mathrm{R}^{n \times n}$, $a \in \mathrm{R}^n$, and T is a maximal monotone polyhedral multifunction. For this problem, the T-map is given by

$$A_T := AP_T(x) + x - P_T(x) - a. \tag{48}$$

As was shown in Section 5.2, for polyhedral T, A_T is piecewise affine with respect to some *finite* collection $\mathcal{M}$ of n-cells whose union is R^n. Thus, to complete the description of the algorithm for affine generalized equations, it remains to show how to generate the representations.

The task of constructing $\mathcal{M}$ is dependent upon how T is described. For example, in Robinson (1992), T is taken as the normal cone N_C to a polyhedral convex set C. $\mathcal{M}$ is then chosen to be the *normal manifold*, which is defined in terms of the nonempty faces F_i of C. Specifically, the cells of the normal manifold are defined by

$$\sigma_i := F_i + N_{F_i},$$

where N_{F_i} is the common value of N_{F_i} for $x \in \mathrm{ri}(F_i)$. This particular choice of cells leads to the algorithm given in (Cao & Ferris 1995*b*).

For more general T, we assume that T is described as the union of a finite collection of polyhedral convex sets C_i. We can then describe P_T as the union of the sets $S_i := \{(x+y, x) \mid (x,y) \in C_i\}$. By projecting each S_i onto the domain of P_T, we produce a collection of sets

$$\sigma_i := \{x + y \mid (x,y) \in C_i\}.$$

Further, since we know $\mathrm{dom}(P_T) = \mathrm{R}^n$, it follows that $\bigcup \sigma_i = \mathrm{R}^n$. We then let $\mathcal{M} := \{\sigma_i \mid \mathrm{int}(\sigma_i) \neq \emptyset\}$.

To provide an example of this process, we return to the case where $T = N_C$. Observe that

$$N_C = \bigcup_{F_i} \bigcup_{x \in F_i} \{x\} \times N_F = \bigcup_{F_i} F_i \times N_{F_i}.$$

Thus, we see that N_C is the union of the polyhedral convex sets

$$C_i := \{x + y \mid (x,y) \in F_i \times N_{F_i}\} = F_i + N_{F_i}.$$

It follows that the process described above yields the normal manifold.

Robinson (1992, Proposition 2.4) proves that the normal manifold is a subdivision of R^n. However, in general, the collection of cells $\mathcal{M}$ generated by the above process is not a subdivision. This can be demonstrated by the following example. Let

$$\begin{array}{lll} C_1 & := & \Big\{(x,0) \in \mathrm{R}^2 \times \mathrm{R}^2 \mid x_1 \leq 0\Big\} \\ C_2 & := & \Big\{(x,0) \in \mathrm{R}^2 \times \mathrm{R}^2 \mid x_1 \geq 0, x_2 \geq 0\Big\} \\ C_3 & := & \Big\{(x,0) \in \mathrm{R}^2 \times \mathrm{R}^2 \mid x_1 \geq 0, x_2 \leq 0\Big\} \end{array}$$

and let $T := \bigcup_{i=1}^3 C_i$. Observe that T is simply the zero mapping, and is thus a maximal monotone multifunction. However, employing our procedure for constructing $\mathcal{M}$, we obtain $\sigma_1 = \Big\{x \in \mathrm{R}^2 \mid x_1 \leq 0\Big\}$, $\sigma_2 = \Big\{x \in \mathrm{R}^2 \mid x_1 \geq 0, x_2 \geq 0\Big\}$, $\sigma_3 = \Big\{x \in \mathrm{R}^2 \mid x_1 \geq 0, x_2 \leq 0\Big\}$. Since $\sigma_1 \bigcap \sigma_2$ is not a face of σ_1, we see that $\mathcal{M} := \{\sigma_1, \sigma_2, \sigma_3\}$ is not a subdivision of R^n.

Since P_T is single-valued, then by Lemma 5.2.3, P_T is affine on each cell $\sigma_i \in \mathcal{M}$. A representation of A_T on each cell is then given by (48). In order to have a workable description of these affine maps, it would appear necessary to exploit the underlying structure of T. One such case is the subject of the next section.

5.4 Separable T

A particularly important class of affine variational inequalities is that for which the set C is *rectangular*, i.e., C is defined by the constraints

$$l \leq z \leq u$$

where l and u are vectors in R^n, with $l_i \in [-\infty, \infty)$ and $u_i \in (-\infty, \infty]$ for $1 \leq i \leq n$. This problem class has a number of features that are very attractive for pivotal algorithms similar to Algorithm AGE. In particular, the cells of linearity of the normal map are rectangular, and furthermore the normal map itself takes on a very simple form. Specifically, for an affine function $F(z) := Az + b$, the matrix used to represent the normal map on any cell is formed simply by replacing some of the columns of A by the corresponding columns of the identity matrix.

Rectangular variational inequalities are also attractive from a theoretical standpoint. In particular, if at least one of l_i and u_i is finite for each i, then the normal map is coherently oriented with respect to C if and only if A is a *P-matrix*.

Definition 5.4.1 (Cottle et al. (1992)) *A matrix A is said to be a P-*matrix *if all its principal minors are positive.*

Note that when C is rectangular, then $N_C(z) = N_{[l,u]}(z) = \prod_{i=1}^{n} N_{[l_i,u_i]}(z_i)$. This suggests that we can extend the notion of rectangularity to generalized equations by requiring that the multifunction T be *separable*, i.e., it is of the form

$$T(z) = \begin{bmatrix} T_1(z_1) \\ T_2(z_2) \\ \vdots \\ T_n(z_n) \end{bmatrix}.$$

where for each i, T_i is a maximal monotone polyhedral multifunction from R to R. With such a T, we shall see that the cells of linearity of the T-map A_T are rectangular.

We begin by looking at the resolvent operator $P_T = (I + T)^{-1}$. Note that

$$P_T(x) = \begin{bmatrix} P_{T_1}(x_1) \\ P_{T_2}(x_2) \\ \vdots \\ P_{T_n}(x_n) \end{bmatrix},$$

where for each i, $P_{T_i} = (1 + T_i)^{-1}$. Since P_T is a continuous piecewise affine function, it follows that P_{T_i} is a a continuous piecewise affine function from R into R. Let k_i be the number of breakpoints of P_{T_i}. Then, for some strictly increasing sequence of breakpoints $\{\xi_{ij}\}$, $j = 1, \ldots, k_i$ and some set of coefficients $\{d_{ij}, b_{ij}\}$, $j = 0, \ldots, k_i$,

$$P_{T_i}(x) = \begin{cases} d_{i0}x + b_{i0} & x \leq \xi_{i1} \\ d_{ij}x + b_{ij} & \xi_{ij} \leq x \leq \xi_{i(j+1)}, \quad 1 \leq j < k_i \\ d_{ik_i}x + b_{ik_i} & \xi_{ik_i} \leq x. \end{cases}$$

Note that since P_T is monotonic and nonexpansive, $0 \leq d_{ij} \leq 1$.

The breakpoint sequence defines a subdivision of R given by $\mathcal{M}_i = \bigcup_{j=0}^{k_i} \sigma_{ij}$, where

$$\sigma_{ij} = \left\{ x \;\middle|\; \begin{array}{cc} x \leq \xi_{i1} & j = 0 \\ \xi_{ij} \leq x \leq \xi_{i(j+1)} & 0 < j < k_i \\ \xi_{ik_i} \leq x & j = k_i \end{array} \right\}.$$

We then define a subdivision of R^n by $\mathcal{M} = \prod_{i=1}^{n} \mathcal{M}_i$ with n-cells defined by $\sigma_{[j_1,j_2,\ldots,j_n]} = \sigma_{1j_1} \times \sigma_{2j_2} \times \ldots \times \sigma_{nj_n}$.

Clearly, P_T (and therefore A_T) is $\mathcal{M}$-PA. This establishes our earlier statement that the cells of linearity of A_T are rectangular. For each cell $\sigma_{[j_1,j_2,\ldots,j_n]}$ of $\mathcal{M}$, define a diagonal matrix $D_{[j_1,j_2,\ldots,j_n]}$ by $D_{[j_1,j_2,\ldots,j_n]}(i,i) = d_{ij_i}$. Further, define the vector $b_{[j_1,j_2,\ldots,j_n]} = [b_{1j_1}; b_{2j_2}; \ldots; b_{nj_n}]$. Then on $\sigma_{[j_1,j_2,\ldots,j_n]}$, P_T is represented by $(D_{[j_1,j_2,\ldots,j_n]}, b_{[j_1,j_2,\ldots,j_n]})$. Thus, on $\sigma_{[j_1,j_2,\ldots,j_n]}$, the T-map is given by

$$\begin{aligned} A_T(x) &= A(P_T(x)) + a + x - P_T(x) \\ &= (AD_{[j_1,j_2,\ldots,j_n]} + I - D_{[j_1,j_2,\ldots,j_n]})x + (A - I)b_{[j_1,j_2,\ldots,j_n]} + a \\ &= [d_{1j_1}A_{\cdot 1} + (1 - d_{1j_1})I_{\cdot 1}, d_{2j_2}A_{\cdot 2} + (1 - d_{2j_2})I_{\cdot 2}, \ldots, d_{nj_n}A_{\cdot n} \\ &\quad +(1 - d_{nj_n})I_{\cdot n}]\, x + (A - I)b_{[j_1,j_2,\ldots,j_n]} + a \\ &=: M_{[j_1,j_2,\ldots,j_n]}x + \hat{b}_{[j_1,j_2,\ldots,j_n]} \end{aligned}$$

Thus, we see that the matrix $M_{[j_1,j_2,\ldots,j_n]}$ which represents A_T on $\sigma_{[j_1,j_2,\ldots,j_n]}$ has columns which are convex combinations of columns of A and the corresponding columns of I.

We now set about proving the main result of this section. Namely, if A is a P-matrix, then A_T is coherently oriented for any separable polyhedral maximal monotone multifunction T. We first need to prove two technical lemmas.

Lemma 5.4.2 *If A and B are $n \times n$ matrices where B is rank-1 such that* $\det(A) > 0$ *and* $\det(A + B) > 0$*, then* $\det(A + \lambda B) > 0$ *for all* $\lambda \in [0, 1]$.

Proof

$$\det(A + \lambda B) = \sum_{(j_1,\ldots,j_n)} \det[C_{j_1}, \ldots, C_{j_n}],$$

where the summation is taken over all possible choices of $(j_1, \ldots, j_n)$ such that C_{j_i} is either $A_{\cdot i}$ or $\lambda B_{\cdot i}$. Since B is rank-1, the determinants in the above sum are zero for all choices that include at least two columns of λB. Thus,

$$\begin{aligned}\det(A+\lambda B) &= \det A + \textstyle\sum_{i=1}^{n} \det[A_{\cdot 1}, \ldots, \lambda B_{\cdot i}, \ldots, A_{\cdot n}] \\ &= \det A + \lambda \left(\textstyle\sum_{i=1}^{n} \det[A_{\cdot 1}, \ldots, B_{\cdot i}, \ldots, A_{\cdot n}]\right)\end{aligned}$$

Thus, $\det(A+\lambda B)$ is an affine function of λ, which is positive at $\lambda = 0$ and $\lambda = 1$. Thus, it is positive for all $\lambda \in [0, 1]$. □

Lemma 5.4.3 *Let A be an $n \times n$ matrix and let $\{B^1, \ldots, B^k\}$ be a collection of rank-1 $n \times n$ matrices. If $\det(A + \lambda_1 B^1 + \cdots + \lambda_k B^k) > 0$ for all choices of $\lambda_i = 0$ or 1, then $\det(A + \lambda_1 B^1 + \cdots + \lambda_k B^k) > 0$ for all choices of $\lambda_i \in [0, 1]$.*

Proof (By induction). The lemma is true for $k = 1$ by Lemma 5.4.2. Now, suppose the lemma is true for all $k < m$, we shall prove the lemma true for $k = m$.

Suppose $\{B^1, \ldots, B^m\}$ is a collection of rank-1 $n \times n$ matrices such that $\det(A + \lambda_1 B^1 + \cdots + \lambda_m B^m) > 0$ for all choices of $\lambda_i = 0$ or 1. Let $\hat{A} := A + B^m$. Then $\hat{A}$ and $\{B^1, \ldots, B^{m-1}\}$ satisfy the conditions of the lemma for $k = m - 1$. Thus, if $\lambda_i \in [0, 1]$, $\forall i$, then

$$\det((A + B^m) + \lambda_1 B^1 + \cdots + \lambda_{m-1} B^{m-1}) > 0.$$

Similarly, with $\hat{A} := A$, we have

$$\det(A + \lambda_1 B^1 + \cdots + \lambda_{m-1} B^{m-1}) > 0.$$

From these two results, we see that if we let $\tilde{A} := A + \lambda_1 B^1 + \cdots + \lambda_{m-1} B^{m-1}$, then $\tilde{A}$ and B^m satisfy the hypotheses for Lemma 5.4.2. Thus, for $\lambda_m \in [0, 1]$

$$\begin{aligned}0 &< \det(\tilde{A} + \lambda_m B^m), \\ &= \det(A + \lambda_1 B^1 + \cdots + \lambda_m B^m).\end{aligned}$$

□

Theorem 5.4.4 *If A is a P-matrix, then for* any *separable maximal monotone polyhedral multifunction T, the T-map A_T defined by (48) has the property that in any cell of linearity, the matrix representing A_T has positive determinant. In particular, A_T is coherently oriented.*

Proof Let A_T have the representation $(\hat{A}, b)$ in the n-cell σ. By the our earlier discussion, $\hat{A}$ can be formed by replacing columns of A by a convex combination of columns of A and the corresponding columns of the identity matrix. Thus, the matrix is of the form

$$\hat{A} = A + \lambda_1 B^1 + \cdots + \lambda_n B^n, \qquad \lambda_i \in [0, 1]$$

where $B^i := (I_{\cdot i} - A_{\cdot i})I_{\cdot i}^\top$. Observe that B^i is a rank-1 matrix.

Since A is a P-matrix, the matrix formed by replacing an arbitrary set of columns of A by corresponding columns of the identity matrix has positive determinant. Thus, the matrices $A, B^1, \ldots, B^n$ satisfy the hypotheses of Lemma 5.4.3. Thus, by Lemma 5.4.3, $\det(\hat{A}) > 0$. □

Corollary 5.4.5 *If A is a P-matrix and T is a separable maximal monotone polyhedral multifunction, then using lexicographic ordering, Algorithm AGE will find a solution to $A_T(x) = 0$ in a finite number of steps.*

5.5 Piecewise Linear-Quadratic Programming

We conclude by giving an example of a well known problem in mathematical programming that can be solved using the technique we have presented. The piecewise linear-quadratic programming problem (PLQP) is given by

$$\min h(x) = f(x) + \phi(Ax), \tag{49}$$

where $A \in \mathrm{R}^{m\times n}$, and $f : \mathrm{R}^n \to \mathrm{R}\bigcup\{\infty\}$ and $\phi : \mathrm{R}^m \to \mathrm{R}\bigcup\{\infty\}$ are convex *piecewise linear-quadratic* functions.

Definition 5.5.1 *A function $f : R^n \to R\bigcup\{\infty\}$ is* piecewise linear-quadratic *if $\mathrm{dom} f$ is closed and convex and there exists a finite subdivision $\mathcal{M}$ of $\mathrm{dom}(f)$ such that for each $\sigma \in \mathcal{M}$, $f_{|\sigma}$ is a quadratic function.*

Note that $\mathrm{dom} f$ is polyhedral, and further that since the cells in the subdivision are closed, f is a continuous function on $\mathrm{dom} f$.

The optimality conditions for PLQP are stated by the relation

$$0 \in \partial h(x),$$

where ∂ is the *convex subdifferential* operator defined by

$$\partial f(x) := \{z | f(w) \geq f(x) + z^\top(w - x), \forall w \in \mathrm{dom}(f)\}.$$

Under appropriate constraint qualifications (i.e. $\mathrm{ri}(A(\mathrm{dom}(f))) \bigcap \mathrm{ri}(\mathrm{dom}(\phi)) \neq \emptyset$), then

$$\partial h(x) = \partial f(x) + A^\top \partial \phi(Ax).$$

Thus, for the optimality conditions to be satisfied, there must be an $x \in \mathrm{dom}(f)$ and $y \in \partial\phi(Ax)$ such that $-A^\top y \in \partial f(x)$. By Rockafellar (1970, Theorem 23.5), it follows that

$$Ax \in \partial\phi^*(y),$$

where ϕ^* is the conjugate of ϕ. The optimality conditions are then

$$\begin{array}{rcl} -A^\top y & \in & \partial f(x) \\ Ax & \in & \partial\phi^*(y). \end{array}$$

Thus, if we let

$$\hat{A} := \begin{bmatrix} 0 & A^\top \\ -A & 0 \end{bmatrix}, \qquad T(x;y) := \begin{bmatrix} \partial f(x) \\ \partial\phi^*(y) \end{bmatrix},$$

then the optimality conditions for PLQP can stated as the generalized equation

$$-\hat{A}(x;y) \in T(x;y). \tag{50}$$

The fact that T is polyhedral was shown in Sun (1986). Thus, the optimality conditions for the piecewise linear-quadratic program can be expressed as an affine generalized equation, which can then be solved using our algorithm.

Chapter 6

Conclusions

In this thesis we have developed a number of algorithmic techniques aimed at solving mixed complementarity problems and affine generalized equations. First, we developed the proximal perturbation strategy, a useful tool for enhancing the robustness of descent-based algorithms for mixed complementarity problems. We established the effectiveness of this strategy both theoretically and practically. On the theoretical side, we proved several strong global convergence results that guarantee convergence based upon a pseudomonotonicity assumption at a solution. An important characteristic of this pseudomonotonicity assumption is that it does not preclude functions for which the associated merit function has local minima that are not global minima.

On the practical side, we implemented three algorithms using the proximal perturbation strategy. All three algorithms are significantly more robust than the underlying descent-based algorithms upon which they are based. Moreover, this increase in robustness is achieved at virtually no cost in efficiency. Indeed, all three algorithms maintain the fast local convergence properties of the underlying algorithms. Two of these algorithms, PROXI and SEMICOMP rival PATH and SMOOTH (the current state-of-the-art) in terms of efficiency and robustness. Therefore, we believe that the thesis demonstrates the effectiveness of the proximal perturbation strategy in convincing fashion.

Our computational results indicate that the best algorithm using the proximal perturbation, PROXI, was based upon the least robust basic algorithm, NE/NEWT. This suggests a very promising approach for improving the capabilities of complementarity solvers for large scale problems is to use a simple (and possibly not very robust) basic solver, and enhance its robustness using the proximal perturbation strategy.

In addition to developing the proximal perturbation strategy, we have proposed and implemented extensions of several algorithms to more general problem classes. Of particular practical importance are the generalizations of NCP algorithms to produce algorithms that solve the MCP. Such generalizations are important because many practical problems are more naturally formulated using the MCP framework rather than the NCP framework. Among the NCP algorithms generalized in this thesis are the NE/SQP algorithm of Pang & Gabriel (1993), the semismooth equations approach of De Luca et al. (1995), and the infeasible interior-point algorithm of Wright (1994).

The final algorithm presented in the thesis is a path-following algorithm that finds zeros of piecewise affine maps. In particular, the algorithm can be used to solve affine generalized equations. This algorithm is a generalization of the pivotal algorithm for affine variational inequalities due to Cao & Ferris (1995*b*), and is based in large part upon the piecewise linear homotopy theory of Eaves (1976). We showed that the algorithm generates a solution in a finite number of iterations assuming that the piecewise affine map is coherently oriented.

There are several areas in which the ideas developed in this thesis warrant additional research. In particular, because the PROXI and SEMICOMP algorithms have demonstrated computational success, it appears worthwhile to develop convergence results for these algorithms comparable to what we proved for the QPCOMP algorithm. Further, additional work is needed to understand how best to control the perturbation parameter used in forming the perturbed subproblems. Finally, additional computational testing and experimentation are needed to develop these algorithmic techniques into mature computer codes, which will thoroughly exploit the inherent strengths of the proximal perturbation strategy.

Bibliography

Ahlfors, L. V. (1966), *Complex Analysis, Second Edition*, International Series in Pure and Applied Mathematics, McGraw–Hill Book Company, New York.

Anitescu, M., Lesaja, G. & Potra, F. A. (1994), An infeasible–interior–point predictor–corrector algorithm for the P_x–geometric LCP, Reports on Computational Mathematics 62, Department of Mathematics, The University of Iowa, Iowa City, Iowa.

Anstreicher, K. M. (1989), 'A combined phase I–phase II projective algorithm for linear programming', *Mathematical Programming* **43**, 209–223.

Anstreicher, K. M. (1991), 'A combined phase I–phase II scaled potential algorithm for linear programming', *Mathematical Programming* **52**, 429–440.

Armijo, L. (1966), 'Minimization of functions having Lipschitz-continuous first partial derivatives', *Pacific Journal of Mathematics* **16**, 1–3.

Auslander, A. (1993), 'Convergence of stationary sequences for variational inequalities with maximal monotone operators', *Applied Mathematics & Optimization* **28**, 161–172.

Auslander, A. & Haddou, M. (1994), An interior-proximal method for convex linearly constrained problems and its extension to variational inequalities, Technical report, Université de Paris 1 & Laboratoire d'Econométrie de l'Ecole Polytechnique, Paris.

Bard, Y. (1972), An eclectic approach to nonlinear programming, *in* R. Anderson, L. Jennings & D. Ryan, eds, 'Optimization', University of Queensland Press, St. Lucia, Queensland, Australia, pp. 116–128.

Billups, S. C. & Ferris, M. C. (1994), Solutions to affine generalized equations using proximal mappings, Mathematical Programming Technical Report 94–15, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Billups, S. C. & Ferris, M. C. (1995*a*), 'Convergence of infeasible interior–point algorithms from arbitrary positive starting points', *SIAM Journal on Optimization, forthcoming*.

Billups, S. C. & Ferris, M. C. (1995*b*), QPCOMP: A quadratic programming based solver for mixed complementarity problems, Mathematical Programming Technical Report 95–09, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Brézis, H. (1973), *Opérateurs Maximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hilbert*, North-Holland.

Brooke, A., Kendrick, D. & Meeraus, A. (1988), *GAMS: A User's Guide*, The Scientific Press, South San Francisco, CA.

Calamai, P. H. & Moré, J. J. (1987), 'Projected gradient methods for linearly constrained problems', *Mathematical Programming* **39**, 93–116.

Cao, M. (1994), Piecewise Linear Homotopies and Affine Variational Inequalities, PhD thesis, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Technical Report 1210.

Cao, M. & Ferris, M. C. (1995*a*), 'Lineality removal for copositive–plus normal maps', *Communications on Applied Nonlinear Analysis* **2**, 1–10.

Cao, M. & Ferris, M. C. (1995*b*), 'A pivotal method for affine variational inequalities', *Mathematics of Operations Research, forthcoming*.

Chamberlain, R. M., Powell, M. J. D. & Lemaréchal, C. (1982), 'The watchdog technique for forcing convergence in algorithms for constrained optimization', *Mathematical Programming Study* **16**, 1–17.

Chen, B. & Harker, P. T. (1993), 'A non–interior continuation method for linear complementarity problems', *SIAM Journal on Matrix Analysis* **14**, 1168–1190.

Chen, B. & Harker, P. T. (1995), 'A non–interior continuation method for monotone variational inequalities', *Mathematical Programming, forthcoming*.

Chen, C. (1995), Smoothing Methods in Mathematical Programming, PhD thesis, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Chen, C. & Mangasarian, O. L. (1995*a*), 'A class of smoothing functions for nonlinear and mixed complementarity problems', *Computational Optimization and Applications, forthcoming.*

Chen, C. & Mangasarian, O. L. (1995*b*), 'Smoothing methods for convex inequalities and linear complementarity problems', *Mathematical Programming, forthcoming.*

Clarke, F. H. (1983), *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York.

Cottle, R. W. (1966), 'Nonlinear programs with positively bounded Jacobians', *Journal of the Society for Industrial and Applied Mathematics* **14**, 147–158.

Cottle, R. W. (1968), The principal pivoting method of quadratic programming, *in* G. B. Dantzig & J. A. F. Veinott, eds, 'Mathematics of the Decision Sciences, Part 1', American Mathematical Society, Providence, Rhode Island, pp. 144–162.

Cottle, R. W. & Dantzig, G. B. (1968), 'Complementary pivot theory of mathematical programming', *Linear Algebra and Its Applications* **1**, 103–125.

Cottle, R. W., Pang, J. S. & Stone, R. E. (1992), *The Linear Complementarity Problem*, Academic Press, Boston.

Dantzig, G. B. & Cottle, R. W. (1967), Positive (semi-)definite programming, *in* J. Abadie, ed., 'Nonlinear Programming', North-Holland, Amsterdam, pp. 55–73.

De Luca, T., Facchinei, F. & Kanzow, C. (1995), A semismooth equation approach to the solution of nonlinear complementarity problems, Submitted to Mathematical Programming.

Dirkse, S. P. (1994), Robust Solution of Mixed Complementarity Problems, PhD thesis, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Dirkse, S. P. & Ferris, M. C. (1995*a*), 'MCPLIB: A collection of nonlinear mixed complementarity problems', *Optimization Methods & Software, forthcoming.*

Dirkse, S. P. & Ferris, M. C. (1995*b*), 'The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems', *Optimization Methods & Software, forthcoming.*

Dirkse, S. P. & Ferris, M. C. (1995*c*), 'A pathsearch damped Newton method for computing general equilibria', *Annals of Operations Research, forthcoming.*

Dirkse, S. P., Ferris, M. C., Preckel, P. V. & Rutherford, T. (1994), The GAMS callable program library for variational and complementarity solvers, Mathematical Programming Technical Report 94-07, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Eaves, B. C. (1971), 'On the basic theorem of complementarity', *Mathematical Programming* **1**, 68–87.

Eaves, B. C. (1976), A short course in solving equations with PL homotopies, *in* R. W. Cottle & C. E. Lemke, eds, 'Nonlinear Programming', American Mathematical Society, SIAM–AMS Proceedings, Providence, RI, pp. 73–143.

Facchinei, F. & Soares, J. (1994), A new merit function for nonlinear complementarity problems and a related algorithm, Technical Report 15.94, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Rome, Italy.

Facchinei, F. & Soares, J. (1995), Testing a new class of algorithms for nonlinear complementarity problems, *in* F. Giannessi & A. Maugeri, eds, 'Variational Inequalities and Network Equilibrium Problems', Plenum Press, New York, pp. 69–83.

Ferris, M. C. & Lucidi, S. (1991), Globally convergent methods for nonlinear equations, Technical Report 1030, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin.

Ferris, M. C. & Pang, J. S. (1995), Engineering and economic applications of complementarity problems, Discussion Papers in Economics 95–4, Department of Economics, University of Colorado, Boulder, Colorado. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Ferris, M. C. & Ralph, D. (1995), Projected gradient methods for nonlinear complementarity problems via normal maps, *in* D. Du, L. Qi & R. Womersley, eds, 'Recent Advances in Nonsmooth Optimization', World Scientific Publishers, forthcoming. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Fischer, A. (1992), 'A special Newton–type optimization method', *Optimization* **24**, 269–284.

Fischer, A. (1995), An NCP–function and its use for the solution of complementarity problems, *in* D. Du, L. Qi & R. Womersley, eds, 'Recent Advances in Nonsmooth Optimization, forthcoming', World Scientific Publishers.

Fischer, A. & Kanzow, C. (1994), On finite termination of an iterative method for linear complementarity problems, Preprint MATH–NM–10–1994, Institute for Numerical Mathematics, Technical University of Dresden, Dresden, Germany.

Fletcher, R. (1987), *Practical Methods of Optimization*, second edn, John Wiley & Sons, New York.

Fukushima, M. (1992), 'Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems', *Mathematical Programming* **53**, 99–110.

Gabriel, S. A. (1992), Algorithms for the Nonlinear Complementarity Problem: The NE/SQP Method and Extensions, PhD thesis, The Johns Hopkins University, Baltimore, Maryland.

Gabriel, S. A. & Pang, J. S. (1992), 'An inexact NE/SQP method for solving the nonlinear complementarity problem', *Computational Optimization and Applications* **1**, 67–91.

Garcia, C. B. & Zangwill, W. I. (1981), *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Inc, Englewood Cliffs, New Jersey.

Geiger, C. & Kanzow, C. (1994), 'On the resolution of monotone complementarity problems', Preprint 82, Institute of Applied Mathematics, University of Hamburg. Bundesstrasse 55, D-20146 Hamburg Germany.

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

Güler, O. (1993), 'Existence of interior points and interior paths in nonlinear monotone complementarity problems', *Mathematics of Operations Research* **18**, 128–147.

Güler, O. (1995), 'Generalized linear complementarity problems', *Mathematics of Operations Research* **2**, 441–448.

Habetler, G. J. & Kostreva, M. M. (1978), 'On a direct algorithm for nonlinear complementarity problems', *SIAM Journal on Control and Optimization* **16**, 504–511.

Han, S.-P., Pang, J. S. & Rangaraj, N. (1992), 'Globally convergent Newton methods for nonsmooth equations', *Mathematics of Operations Research* **17**, 586–607.

Harker, P. T. & Pang, J. S. (1990), 'Finite–dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications', *Mathematical Programming* **48**, 161–220.

Harker, P. T. & Xiao, B. (1990), 'Newton's method for the nonlinear complementarity problem: A B–differentiable equation approach', *Mathematical Programming* **48**, 339–358.

Harrison, G. W., Rutherford, T. F. & Tarr, D. G. (1995), Quantifying the Uruguay round, *in* 'The Uruguay Round and the Developing Economies', World Bank Conference, Washington.

Hudson, J. F. P. (1969), *Piecewise Linear Topology*, Benjamin, New York.

Josephy, N. H. (1979), Newton's method for generalized equations, Technical Summary Report 1965, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin.

Kanzow, C. (1994), 'Some equation–based methods for the nonlinear complementarity problem', *Optimization Methods and Software* **3**, 327–340.

Kanzow, C. (1995), 'Global convergence properties of some iterative methods for linear complementarity problems', *SIAM Journal on Optimization, forthcoming*.

Karmarkar, N. (1984), 'A new polynomial time algorithm for linear programming', *Combinatorica* **4**, 373–395.

Kojima, M., Megiddo, N. & Mizuno, S. (1991), A primal-dual exterior point algorithm for linear programming, Technical Report RJ 8500, IBM Alamaden Research Center, San Jose, California.

Kojima, M., Megiddo, N. & Noma, T. (1991), 'Homotopy continuation methods for nonlinear complementarity problems', *Mathematics of Operations Research* **16**, 754–774.

Kojima, M., Mizuno, S. & Noma, T. (1989), 'A new continuation method for complementarity problems with uniform P-functions', *Mathematical Programming* **43**, 107–113.

Kojima, M., Mizuno, S. & Todd, M. (1992), Infeasible interior-point primal-dual potential-reduction algorithms for linear programming, Technical Report 1023, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

Kojima, M., Shindo, S. & Hara, S. (1994), Interior–point methods for the monotone semidefinite linear complementarity problem in symmetric matrices, Research Report on Information Sciences B–282, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan.

Lemke, C. E. & Howson, J. T. (1964), 'Equilibrium points of bimatrix games', *SIAM Journal on Applied Mathematics* **12**, 413–423.

Lustig, I. J., Marsten, R. E. & Shanno, D. F. (1991), 'Computational experience with a primal-dual interior point method for linear programming', *Linear Algebra and Its Applications* **152**, 191–222.

Lustig, I. J., Marsten, R. E. & Shanno, D. F. (1992), 'On implementing Mehrotra's predictor-corrector interior point method for linear programming', *SIAM Journal on Optimization* **2**, 435–449.

Lustig, I. J., Marsten, R. E. & Shanno, D. F. (1994), 'Interior point methods for linear programming: computational state of the art', *ORSA Journal on Computing* **6**(1), 1–38.

Mangasarian, O. L. (1969), *Nonlinear Programming*, McGraw–Hill, New York.

Mangasarian, O. L. (1976), 'Equivalence of the complementarity problem to a system of nonlinear equations', *SIAM Journal on Applied Mathematics* **31**, 89–92.

Mangasarian, O. L. & Solodov, M. V. (1993), 'Nonlinear complementarity as unconstrained and constrained minimization', *Mathematical Programming* **62**, 277–298.

Mathiesen, L. (1985), 'Computation of economic equilibria by a sequence of linear complementarity problems', *Mathematical Programming Study* **23**, 144–162.

Mathiesen, L. (1987), 'An algorithm based on a sequence of linear complementarity problems applied to a Walrasian equilibrium model: An example', *Mathematical Programming* **37**, 1–18.

McShane, K. A., Monma, C. L. & Shanno, D. F. (1989), 'An implementation of a primal-dual interior point method for linear programming', *ORSA Journal of Computing* **1**, 70–83.

Mehrotra, S. (1992), 'On the implementation of a primal-dual interior point method', *SIAM Journal on Optimization* **2**, 575–601.

Mifflin, R. (1977), 'Semismooth and semiconvex functions in constrained optimization', *SIAM Journal on Control and Optimization* **15**, 957–972.

Minty, G. J. (1962), 'Monotone (nonlinear) operators in Hilbert space', *Duke Mathematics Journal* **29**, 341–346.

Mizuno, S. (1993), Polynomiality of Kojima-Megiddo-Mizuno infeasible-interior-point algorithm for linear programming, Technical Report 1006, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

Mizuno, S., Jarre, F. & Stoer, J. (1994), A unified approach to infeasible–interior–point algorithms via geometrical linear complementarity problems, Preprint Nr. 213, Mathematische Institut der Universität Würzburg, Würzburg, Germany.

Monteiro, R. D. C. & Wright, S. J. (1993), A superlinear infeasible-interior-point affine scaling algorithm for LCP, Preprint MCS-P361-0693, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois.

Monteiro, R. D. C., Pang, J. S. & Wang, T. (1995), 'A positive algorithm for the nonlinear complementarity problem', *SIAM Journal on Optimization, forthcoming*.

Moré, J. J. (1994), Global methods for nonlinear complementarity problems, Technical Report MCS-P429-0494, Argonne National Laboratory, Argonne, Illinois.

Murtagh, B. A. & Saunders, M. A. (1983), MINOS 5.0 user's guide, Technical Report SOL 83.20, Stanford University.

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

Pang, J. & Wang, Z. (1990), Embedding methods for variational inequality and nonlinear complementarity problems, Technical report, Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland.

Pang, J. S. (1990), 'Newton's method for B–differentiable equations', *Mathematics of Operations Research* **15**, 311–341.

Pang, J. S. (1991), 'A B–differentiable equation based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems', *Mathematical Programming* **51**, 101–132.

Pang, J. S. & Gabriel, S. A. (1993), 'NE/SQP: A robust algorithm for the nonlinear complementarity problem', *Mathematical Programming* **60**, 295–338.

Pang, J. S. & Qi, L. (1993), 'Nonsmooth equations: Motivation and algorithms', *SIAM Journal on Optimization* **3**, 443–465.

Pang, J. S., Han, S.-P. & Rangaraj, N. (1991), 'Minimization of locally Lipschitzian functions', *SIAM Journal on Optimization* **1**, 57–82.

Potra, F. A. (1992*a*), An infeasible interior-point predictor-corrector algorithm for linear programming, Technical Report 26, Department of Mathematics, University of Iowa, Iowa City, Iowa.

Potra, F. A. (1992*b*), On a predictor-corrector method for solving linear programs from infeasible starting points, Reports on Computational Mathematics 34, Department of Mathematics, University of Iowa, Iowa City, Iowa.

Potra, F. A. (1992*c*), A quadratically convergent infeasible interior-point algorithm for linear programming, Technical Report 28, Department of Mathematics, University of Iowa, Iowa City, Iowa.

Potra, F. A. (1994), An $O(nL)$ infeasible–interior–point algorithm for LCP with quadratic convergence., Reports on Computational Mathematics 50, Department of Mathematics, The University of Iowa, Iowa City, Iowa.

Potra, F. A. & Bonnans, J. F. (1994), Infeasible path following algorithms for linear complementarity problems, Research Report RR–2445, INRIA.

Potra, F. A. & Sheng, R. (1994*a*), A large–step infeasible–interior–point method for the p_*–matrix LCP, Reports on Computational Mathematics 64, Department of Mathematics, The University of Iowa, Iowa City, Iowa.

Potra, F. A. & Sheng, R. (1994*b*), A modified O(nL) infeasible-interior-point algorithm for LCP with quadratic convergence, Reports on Computational Mathematics 54, Department of Mathematics, University of Iowa, Iowa City, Iowa.

Qi, L. (1993), 'Convergence analysis of some algorithms for solving nonsmooth equations', *Mathematics of Operations Research* **18**, 227–244.

Qi, L. & Chen, X. (1993), A globally convergent successive approximation method for nonsmooth equations, Technical report, School of Mathematics, University of New South Wales, Sydney.

Qi, L. & Jiang, H. (1994), Semismooth Karush–Kuhn–Tucker equations and convergence analysis of Newton methods and quasi–Newton methods for solving these equations, Technical Report AMR 94/5, School of Mathematics, University of New South Wales, Wales, Australia.

Qi, L. & Sun, J. (1993), 'A nonsmooth version of Newton's method', *Mathematical Programming* **58**, 353–368.

Ralph, D. (1994), 'Global convergence of damped Newton's method for nonsmooth equations, via the path search', *Mathematics of Operations Research* **19**, 352–389.

Reinoza, A. (1985), 'Solving generalized equations via homotopies', *Mathematical Programming* **31**, 307–320.

Robinson, S. (1979*a*), 'Some continuity properties of polyhedral multifunctions', *Mathematical Programming Study* **14**, 206–214.

Robinson, S. M. (1979*b*), 'Generalized equations and their solution: Part I: Basic theory', *Mathematical Programming Study* **10**, 128–141.

Robinson, S. M. (1980), 'Strongly regular generalized equations', *Mathematics of Operations Research* **5**, 43–62.

Robinson, S. M. (1987), 'Local structure of feasible sets in nonlinear programming, Part III: Stability and sensitivity', *Mathematical Programming Study* **30**, 45–66.

Robinson, S. M. (1992), 'Normal maps induced by linear transformations', *Mathematics of Operations Research* **17**, 691–714.

Rockafellar, R. T. (1970), *Convex Analysis*, Princeton University Press, Princeton, New Jersey.

Rockafellar, R. T. (1978), Monotone operators and augmented Lagrangian methods in nonlinear programming, *in* O. L. Mangasarian, R. R. Meyer & S. M. Robinson, eds, 'Nonlinear Programming 3', Academic Press, London, pp. 1–26.

Rutherford, T. F. (1993), MILES: A mixed inequality and nonlinear equation solver, Working Paper, Department of Economics, University of Colorado, Boulder.

Sellami, H. (1994), A Continuation Method for Normal Maps, PhD thesis, University of Wisconsin–Madison, Madison, Wisconsin.

Shanno, D. & Simantiraki, E. (1995), An infeasible interior–point method to solve the linear complementarity problem, Research Report B–282, Rutcor.

Shapiro, A. (1988), On concepts of directional differentiability, Research Report 73/88(18), Department of Mathematics and Applied Mathematics, University of South Africa, Pretoria, South Africa.

Solodov, M. V. & Tseng, P. (1994), Modified projection–type methods for monotone variational inequalities, Mathematical Programming Technical Report 94-04, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/.

Stoer, J. (1994), 'The complexity of an infeasible interior–point path–following method for the solution of linear programs', *Optimization Methods and Software* **3**, 1–12.

Subramanian, P. K. (1993), 'Gauss–Newton methods for the complementarity problem', *Journal of Optimization Theory and Applications* **77**, 467–482.

Sun, J. (1986), On Monotropic Piecewise Quadratic Programming, PhD thesis, University of Washington, Seattle, Washington.

Tikhonov, A. N. & Arsenin, V. Y. (1977), *Solutions of Ill–Posed Problems*, John Wiley & Sons, New York.

Todd, M. J. (1976), *Computation of Fixed Points and Applications*, Vol. 124 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Heidelberg.

Tseng, P. (1994), Growth behavior of a class of merit functions for the nonlinear complementarity problem, Technical report, Department of Mathematics, University of Washington, Seattle, Washington.

Tseng, P., Yamashita, N. & Fukushima, M. (1994), 'Equivalence of complementarity problems to differentiable minimization: a unified approach', *SIAM Journal on Optimization, forthcoming.*

Wang, T., Monteiro, R. D. C. & Pang, J. S. (1994), An interior point potential reduction method for constrained equations, Manuscript, Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland.

Watson, L. T. (1979), 'Solving the nonlinear complementarity problem by a homotopy method', *SIAM Journal on Control and Optimization* **17**, 36–46.

Wright, S. J. (1993), 'A path–following infeasible–interior–point algorithm for linear complementarity problems', *Optimization Methods and Software* **2**, 79–106.

Wright, S. J. (1994), 'An infeasible–interior–point algorithm for linear complementarity problems', *Mathematical Programming* **67**, 29–51.

Wright, S. J. & Ralph, D. (1993), A superlinear infeasible–interior–point algorithm for monotone complementarity problems, Technical Report MCS–P344–1292, Argonne National Laboratory, Argonne, Illinois.

Wright, S. J. & Zhang, Y. (1994), A superquadratic infeasible–interior–point method for linear complementarity problems, Preprint MCS–P418–0294, Argonne National Laboratory, Argonne, Illinois.

Xiao, B. & Harker, P. T. (1994*a*), 'A nonsmooth Newton method for variational inequalities: I: Theory', *Mathematical Programming* **65**, 151–194.

Xiao, B. & Harker, P. T. (1994*b*), 'A nonsmooth Newton method for variational inequalities: II: Numerical results', *Mathematical Programming* **65**, 195–216.

Zhang, Y. (1994), 'On the convergence of a class of infeasible interior–point methods for the horizontal linear complementarity problem', *SIAM Journal on Optimization* **4**, 208–227.

Zhang, Y. & Zhang, D. (1992), Superlinear convergence of infeasible interior-point methods for linear programming, Technical Report 92-15, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Maryland.

Zoutendijk, G. (1960), *Methods of Feasible Directions*, Elsevier Publishing Company, Amsterdam.