STUDIES ON
ALGORITHMS FOR LARGE-SCALE NONLINEAR OPTIMIZATION AND RELATED PROBLEMS

HIROSHIGE DAN

STUDIES ON
ALGORITHMS FOR LARGE-SCALE NONLINEAR OPTIMIZATION AND RELATED PROBLEMS

by

HIROSHIGE DAN

Submitted in partial fulfillment of
the requirement for the degree of
DOCTOR OF INFORMATICS
(Applied Mathematics and Physics)

KYOTO UNIVERSITY
KYOTO 606–8501, JAPAN
SEPTEMBER 2004

Preface

In this thesis, we study algorithms for solving the optimization problem, the system of equations and the complementarity problem. The optimization problem is to find a solution which satisfies constraints and minimizes or maximizes an objective function. This problem has many real world applications, and so it is an important problem. The system of equations appears in various fields, especially in engineering, and so extensive research has been done thus far. The optimization problem of special form can often be directly reduced to the system of equations. The complementarity problem can deal with the equilibrium problem and appears in the first order optimality condition, so it is closely related to the optimization problem. These problems have been studied for a long time, and many basic algorithms for solving them have been proposed. In this thesis, we propose algorithms which have better theoretical and numerical properties than previous ones.

In many cases, problems in the real world can be formulated as a linear model, but it is clear that problems exist which can only be formulated appropriately as a nonlinear model. Moreover, until now, most solvers have lacked the sophistication to solve nonlinear problems, and so those problems have been handled through linear approximations. However, advances in computer technology coupled with the development of efficient algorithms now make it possible to solve nonlinear problems without having resort to linear approximations. The increase in computer memory size allows us to solve a wider range of problems, and the improvement in CPU clock speed shortens calculation time.

However, current algorithms cannot necessarily solve all problems which are nonlinear and large-scale. For example, because of nonlinearity of the problems, we may fail to achieve fast convergence. In addition, when problems are large-scale, we must consider the numerical round off error, and the shortage of memory may occur even with the memory size of modern computers. This prompts us to develop more efficient algorithms to solve large-scale nonlinear optimization problems, systems of equations, and complementarity problems. In designing algorithms, it is important to fully examine the convergence property and robustness. We particularly take this into consideration in this thesis.

The author hopes that the results of this thesis will contribute further research in this field.

Hiroshige Dan
September 2004

Acknowledgment

I would like to start by extending my heartfelt gratitude to Professor Masao Fukushima of Kyoto University. He gave me the invaluable opportunity to enter the Doctoral course of Kyoto University despite my full time work in Tokyo. This has been one of the greatest opportunities in my life. Without his continuous instruction and encouraging support, it would not have been possible for me to write this thesis.

I am also grateful to Assistant Professor Nobuo Yamashita of Kyoto University. His comments are always highly suggestive and make my research more meaningful. I could not have imagined writing this thesis without his innovative thinking. Further, I would like to thank Professor Tetsuya Takine of Osaka University, former Associate Professor of Kyoto University. Although I have not had many opportunities to take on his instruction during my research, he provided me with incredible encouragement and support.

I have worked at Mathematical Systems, Inc. since April 2001. President Hiroshi Yamashita and Division Manager Takahito Tanabe of Mathematical Systems, Inc. provided me an excellent environment for my research. I would like to express my appreciation for their consideration.

I would also like to thank everyone who has continued to support me, including all members of Fukushima Laboratory of Kyoto University and Numerical Optimization Division of Mathematical Systems, Inc.

Finally, I would like to dedicate this thesis to my family members for their unconditional support and understanding.

Contents

List of Tables

Chapter 1

Introduction

The optimization problem, the system of equations and the complementarity problem have various applications in the real world, especially, engineering, finance and so on. In this thesis, we propose algorithms for solving these problems, which get rid of difficulties in solving nonlinear and large-scale problems, such as slow convergence, shortage of memory size, and so on.

In this chapter, we give an overview of problems and methods which we will consider in this thesis. We also outline the contents of the thesis.

1.1 Overview of problems

1.1.1 Nonlinear optimization problem

In the thesis, we consider the following nonlinear optimization problem (NLP) with equality and inequality constraints:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) = 0 \ (j \in J_E), \\ & g_j(x) \geq 0 \ (j \in J_I), \end{array} \tag{1.1.1}$$

where $f : \Re^n \to \Re^1$ and $g_j : \Re^n \to \Re^1$ $(j \in J_E \cup J_I)$ with J_E and J_I being finite index sets such that $J_E \cap J_I = \emptyset$. Throughout this thesis, we assume that f and g_j $(j \in J_E \cup J_I)$ are twice continuously differentiable. The Lagrangian of (1.1.1) is defined by

$$L(x, y) = f(x) - \sum_{j \in J_E} y_j g_j(x) - \sum_{j \in J_I} y_j g_j(x),$$

where $y_j (j \in J_E \cup J_I)$ are Lagrange multipliers associated with the constraints.

Various methods have been proposed thus far to solve (1.1.1), for example,

- the penalty function method [1, 28],
- the barrier function method [1, 28],
- the augmented Lagrangian method [1, 28],

- the sequential quadratic programming (SQP) method [1, 28],
- the interior point method [56].

In this thesis, we propose an SQP-based method to solve (1.1.1), and so we briefly summarize results of previous research on the SQP method in the following.

In the SQP method, we usually solve the following problem at a current point x^k as a subproblem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}d^T G_k d + \nabla f(x^k)^T d \\ \text{subject to} & g_j(x^k) + \nabla g_j(x^k)^T d = 0 \ (j \in J_E), \\ & g_j(x^k) + \nabla g_j(x^k)^T d \geq 0 \ (j \in J_I), \end{array} \tag{1.1.2}$$

where G_k is an approximate matrix of $\nabla_x^2 L(x^k, y^k)$. Here y^k is a vector with elements y_j^k ($j \in J_E \cup J_I$) which are appropriate Lagrange multipliers at x^k. (1.1.2) is an approximate problem of (1.1.1) at x^k by a quadratic programming problem (QP). We use a solution of (1.1.1) as a search direction to find the next iteration point.

Now it is very important how we construct G_k in (1.1.2). There are two major methods to construct G_k. One method is to use $\nabla_x^2 L(x^k, y^k)$ itself, and the other is to use a matrix which approximate to $\nabla_x^2 L(x^k, y^k)$ by quasi-Newton methods. When we use $\nabla_x^2 L(x^k, y^k)$ as G_k, the quadratic convergence is guaranteed under appropriate assumptions [3], but the global convergence is not guaranteed. Otherwise, when we use quasi-Newton methods to construct G_k, the global convergence is guaranteed because of positive definiteness of G_k. However, matrices constructed by quasi-Newton methods are usually dense, then they are not suitable for large-scale problems.

To resolve these problems, some SQP-based methods which have the global convergence and are suitable for large-scale problems have been proposed. In these methods, a trust region technique [10] has been used. However, we also have two serious difficulties in these methods. One difficulty is that subproblems are not convex quadratic programming problems, so it is difficult to solve them. The other is that subproblems may be infeasible when a radius of a trust region becomes too small. For the latter difficulty, a few methods have been proposed, for example, there exist methods in which parameters are introduced in subproblems [5, 6, 44, 52] and subproblems are solved in two steps [38, 40]. However, these methods are so complicated that they are not suitable for implementation.

In this thesis, we will propose an SQP-based method which has global convergence property. In our method, we solve two subproblems in each iteration. One subproblem is a convex QP, and the other is a system of linear equations, then both of them can be solved by traditional methods. Moreover, these subproblems have sparsity, so our method is suitable for large-scale problems.

1.1.2 System of nonlinear equations

The system of nonlinear equations (SNE) is very important because it has so many applications in almost all fields of engineering. Especially, it is strongly concerned with optimization. In this thesis, we consider the following SNE:

$$F(x) = 0, \tag{1.1.3}$$

where $F : \Re^n \to \Re^m$. The SNE (1.1.3) can be classified in view of smoothness of F. Generally speaking, applications where F is smooth seem to be major in the real world, but we also have many applications where F is nonsmooth. For example, when we solve the complementarity problem, we often consider a nonsmooth SNE which is equivalent to the complementarity problem. We will explain this in detail in the next subsection and Section 2.2.

Now we consider the following NLP to verify the relationship between SNE and NLP.

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) = 0 \ (j = 1, 2, \dots, r), \end{array} \tag{1.1.4}$$

where $f : \Re^n \to \Re^1$ and $g_j : \Re^n \to \Re^1$ $(j = 1, 2, \dots, r)$. We note that (1.1.4) is the case where we set $J_E = \{1, 2, \dots, r\}$ and $J_I = \emptyset$ in (1.1.1). The Karush-Kuhn-Tucker (KKT) condition, which is the first order necessary condition for optimality, for (1.1.4) is

$$\begin{pmatrix} \nabla_x L(x, y) \\ g_1(x) \\ \vdots \\ g_r(x) \end{pmatrix} = 0.$$

This equation is one of concrete examples of (1.1.3).

We can enumerate representative methods to solve the smooth SNE as follows:

- Newton's method,
- the quasi-Newton method,
- the Gauss-Newton method [31],
- the conjugate gradient method [34],
- the Levenberg-Marquardt method (LMM) [20, 31, 60].

Moreover, when we solve the nonsmooth SNE,

- the generalized Newton's method [12, 45],
- the generalized Levenberg-Marquardt method [20]

are often used.

In this thesis, we will propose a method for the smooth SNE. Our method is based on the LMM, so we shortly introduce it in the following.

In the LMM, we solve the following system of linear equations at the present iterative point x^k:

$$\left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right) d = -\nabla F(x^k)^T F(x^k), \tag{1.1.5}$$

where $\nabla F(x^k)$ is the Jacobian of F at x^k, μ_k is a positive parameter and I is the identity matrix. The LMM generates a sequence $\{x^k\}$ by $x^{k+1} = x^k + d^k$, where d^k denotes the solution of (1.1.5). We note that (1.1.5) has the unique solution because a coefficient matrix of (1.1.5) becomes positive definite, even if $\nabla F(x^k)^T \nabla F(x^k)$ is singular. The LMM has such a good property which Newton's and the Gauss-Newton method do not have.

In traditional convergence theories, the nonsingularity of the Jacobian at a solution guarantees that the LMM has a quadratic rate of convergence. Then, when $\nabla F(x^k)^T \nabla F(x^k)$ converges to a singular matrix, the LMM may fail to have a fast convergence property.

In the meanwhile, a local error bound condition [43] gets much attention nowadays, concerning the nonsingularity of the Jacobian. When a local error bound condition holds, we can locally estimate the distance between a point and a solution set. This property is milder than the nonsingularity of the Jacobian and it is very important when we solve optimization and related problems on a computer in practice. Especially, Yamashita and Fukushima showed that the LMM has a quadratic rate of convergence under a local error bound condition [60], which is milder than those of traditional convergence theories.

In this thesis, we will propose a new algorithm based on the LMM. Our algorithm is an extension of [60]. To be more precise, we will propose an inexact Levenberg-Marquardt method (ILMM), which allow an inexact solution in solving (1.1.5), and has a superlinear rate of convergence under a local error bound condition.

1.1.3 Nonlinear complementarity problem

In this subsection, we explain the nonlinear complementarity problem (NCP) [22]. We can deal with various equilibrium problems by NCP. NCP is defined as follows:

$$\mathrm{NCP}(F): \quad \text{Find} \quad \bar{x} \in \Re^n \quad \text{such that} \quad \bar{x}_i \geq 0, \;\; F_i(\bar{x}) \geq 0, \;\; \bar{x}_i F_i(\bar{x}) = 0, \quad i = 1, \ldots, n,$$

where F is a mapping from $\Re^n$ to $\Re^n$. Throughout this thesis, we assume that F is monotone and continuously differentiable. We will give the definition of monotonicity in Section 2.3.

From the definition of NCP(F), we can classify each index i into one of the following three sets at a solution $\bar{x}$:

$$\begin{aligned} P(\bar{x}) &:= \{i \mid \bar{x}_i > 0, F_i(\bar{x}) = 0\}, \\ N(\bar{x}) &:= \{i \mid \bar{x}_i = 0, F_i(\bar{x}) > 0\}, \\ C(\bar{x}) &:= \{i \mid \bar{x}_i = 0, F_i(\bar{x}) = 0\}. \end{aligned}$$

If $C(\bar{x}) = \emptyset$ holds, we call $\bar{x}$ a nondegenerate solution, and otherwise, a degenerate solution.

NCP is closely related to NLP. To confirm this, we consider the following NLP:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \geq 0, \end{array} \tag{1.1.6}$$

where $f : \Re^n \to \Re^1$. We note that (1.1.6) is the case where we set $J_E = \emptyset, J_I = \{1, 2, \ldots, n\}$ and $g_j(x) = x_j \ (j \in J_I)$ in (1.1.1). The KKT conditions for (1.1.6) are

$$\nabla f(x) = y \geq 0, \; x \geq 0, \; y^T x = 0, \tag{1.1.7}$$

where y is a vector of Lagrange multipliers associated with $x \geq 0$. We can easily find that (1.1.7) is equivalent to NCP(∇f). Especially, ∇f is monotone when f is convex [47]. In this way, NCP has a strong connection with NLP. In addition, the optimality conditions for NLP are generally reduced to the variational inequality problem (VIP) [22]. NCP is a special case of VIP.

Most methods for NCP(F) use the system of equations, $H_F(x) = 0$, which is equivalent to NCP(F). We will give the concrete formulations of H_F in Section 2.2, and we note that H_F is nonsmooth as we explain in the previous section. We can enumerate methods for NCP(F) which use H_F as follows:

- the generalized Newton method [12, 45],
- the smoothing method [7, 8, 9, 29],
- the regularization method [21], and
- the proximal point algorithm (PPA) [59, 61].

Out of these methods, the generalized Newton method, the smoothing method and the regularization method need a certain type of nonsingularity of the Jacobian of H_F at a solution to have a quadratic rate of convergence. However, the PPA has a superlinear rate of convergence when $\|H_F(x)\|$ provides a local error bound for a solution set of NCP(F) [59, 61]. If the Jacobian of H_F is nonsingular, then $\|H_F(x)\|$ provides a local error bound. Accordingly, the PPA converges superlinearly under milder conditions than the generalized Newton method, the smoothing method and the regularization method. Though, in the previous research about the PPA, it is assumed for solving subproblems efficiently that $C(\bar{x}) = \emptyset$ holds, where $\bar{x} = \lim_{k\to\infty} x^k$ and $\{x^k\}$ is a sequence generated by the PPA.

The essential property of NCP(F) is to classify each index into $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$. If we can identify $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$ in advance of convergence, we may construct a new algorithm which uses this information. An identification method has been proposed in [18], but this method needs to assume local uniqueness of a solution.

We will propose two algorithms for NCP in this thesis. One algorithm is to identify $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$ in advance of convergence when we use the PPA to solve NCP. The other is to generate a sequence converging to a solution set superlinearly without uniqueness and nondegeneracy of a solution. The latter algorithm is a hybrid method which combines the former algorithm and the ILMM.

1.2 Outline of the thesis

In this section, we explain the outline of this thesis.

In Chapter 2, we give some concepts and mathematical properties which are used in this thesis.

In Chapter 3, we propose an SQP-based method for NLP. Our method enjoys a global convergence property. In our method, there are two subproblems in each iteration. One subproblem is a QP with equality and inequality constraints, and the Hessian of its objective function is a positive definite diagonal matrix. The other is a QP with equality constraints, and the Hessian of its objective function is $\nabla_x^2 L(x,y)$. The former subproblem is a strictly convex QP, so we can use traditional methods to solve it. Moreover, the KKT condition of the latter subproblem is the system of linear equations, then it is relatively easy to solve this subproblem. In addition, the Hessians of objective functions in these subproblems can be expected to be sparse, so our method is suitable for the large-scale NLP.

In Chapter 4, we propose an iterative method for the smooth SNE. Our method is the ILMM, which is an extension of [60] and has a superlinear rate of convergence under local error bound conditions which are milder than the nonsingularity of the Jacobian. Moreover, our method is especially suitable for large-scale problems. In large-scale problems, it is difficult to find an exact solution of (1.1.5) because of numerical error, but our method allow residuals to a certain extent when we solve (1.1.5), so we can find an appropriate solution easily.

In Chapters 5 and 6, we consider the monotone NCP. In Chapter 5, we propose a method which identifies $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$ in advance of convergence when we use the PPA to solve the monotone NCP. Our method does not need to assume uniqueness and nondegeneracy of a solution. In Chapter 6, we propose a method which combines two methods proposed in Chapters 4 and 5 for the monotone NCP. To be more precise, we first identify the index sets $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$ by the method proposed in Chapter 5, and using this result, we construct an SNE which is equivalent to NCP(F) locally. Moreover, we solve this system by the ILMM proposed in Chapter 4. In the previous research on NCP, it is necessary for a superlinear or quadratic rate of convergence to assume a certain type of nonsingularity or degeneracy at a solution, but our method has a superlinear rate of convergence under milder conditions than those of previous methods.

Lastly, in Chapter 7, we make some concluding remarks.

The results proposed in Chapter 3 is based on [54], Chapter 4 on [14], Chapter 5 on [57] and Chapter 6 on [15].

Chapter 2

Preliminaries

In this chapter, we give some definitions and preliminary results which will be necessary in the following chapters.

First, we define some notations which are used in this thesis. $\|\cdot\|$ denotes the l_2-norm, i.e., $\|x\| := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$ $(x \in \Re^n)$, $[\cdot]_+$ denotes the orthogonal projection onto the nonnegative orthant $\Re^n_+$, $\mathrm{dist}\{x, X^*\}$ denotes the distance between $x \in \Re^n$ and a set $X^* \subseteq \Re^n$, i.e., $\mathrm{dist}\{x, X^*\} := \min\{\|x - \bar{x}\| \mid \bar{x} \in X^*\}$, and $B(x^*, b)$ denotes the ball with center x^* and radius b, i.e., $B(x^*, b) := \{x \in \Re^n \mid \|x - x^*\| \le b\}$. Moreover, the kth power of a scalar α is denoted by $(\alpha)^k$.

2.1 Unconstrained optimization problem

In this section, we review some basic properties of unconstrained optimization problems. Those properties will be used in Chapter 4.

Consider the following unconstrained optimization problem:

$$\text{minimize} \quad \phi(x), \tag{2.1.1}$$

where $\phi : \Re^n \to \Re^1$ is assumed to be continuously differentiable. We need to find a point x^* satisfying $\nabla\phi(x^*) = 0$, which is the first order necessary condition for optimality. Moreover, x^* is called a stationary point of (2.1.1).

The gradient method [1] is one of the most well-known methods for (2.1.1). In this method, we find a search direction d^k which satisfies

$$\nabla\phi(x^k)^T d^k < 0$$

at a current iterative point x^k, if $\nabla\phi(x^k) \neq 0$. Moreover, we find an appropriate stepsize $\alpha^k > 0$ and set a next iterative point as $x^{k+1} = x^k + \alpha^k d^k$.

There are some rules to choose a stepsize α^k, for instance, the minimization rule, the limited minimization rule, the Armijo's rule and so on. Now we explain the Armijo's rule, which will be used in Chapter 4. In the Armijo's rule, we find the first nonnegative integer $m = m_k$ which satisfies

$$\phi(x^k) - \phi(x^k + (\beta)^m d^k) \ge -s(\beta)^m \nabla\phi(x^k)^T d^k,$$

where $s \in (0,1)$ and $\beta \in (0,1)$ are some constants. Then, we set $\alpha^k = (\beta)^{m_k}$.

In connection with the direction sequence $\{d^k\}$, the following property is important.

Definition 2.1.1 *A sequence $\{d^k\}$ is said to be gradient related to $\{x^k\}$ if $\{d^k\}_{k\in K}$ is bounded and satisfies*

$$\limsup_{k\to\infty, k\in K} \nabla\phi(x^k)^T d^k < 0$$

for any subsequence $\{x^k\}_{k\in K}$ converging to a nonstationary point of f.

When $\{d^k\}$ is gradient related to $\{x^k\}$, the following theorem holds.

Theorem 2.1.1 *[1, Proposition 1.2.1] Let $\{x^k\}$ be a sequence generated by a gradient method $x^{k+1} = x^k + \alpha^k d^k$, and assume that $\{d^k\}$ is gradient related to $\{x^k\}$ and α^k is chosen by the minimization rule, or the limited minimization rule, or the Armijo's rule. Then every limit point of $\{x^k\}$ is a stationary point of (2.1.1).*

2.2 Reformulation of NCP

In this section, we introduce an SNE which is equivalent to NCP(F). This is one of the most important ideas to solve NCP, as we have mentioned in Subsection 1.1.3.

A function $\phi : \Re^2 \to \Re$ that satisfies the following relation is called an NCP function:

$$\phi(a,b) = 0 \iff a \geq 0,\ b \geq 0,\ ab = 0.$$

Two well known examples of the NCP function are the natural residual function ϕ_{NR} and the Fischer-Burmeister function ϕ_{FB} [25] defined by

$$\phi_{NR}(a,b) := \min\{a,b\}$$

and

$$\phi_{FB}(a,b) := \sqrt{a^2+b^2} - a - b,$$

respectively. It is not difficult to see that these two functions satisfy the inequalities [51]

$$(2-\sqrt{2})|\phi_{NR}(a,b)| \leq |\phi_{FB}(a,b)| \leq (2+\sqrt{2})|\phi_{NR}(a,b)| \quad \forall (a,b)^T \in \Re^2,$$

which imply $|\phi_{NR}(a,b)|$ and $|\phi_{FB}(a,b)|$ are equivalent in a certain sense.

Let $H_F : \Re^n \to \Re^n$ be defined by

$$H_F(x) := \begin{pmatrix} \phi(x_1, F_1(x)) \\ \vdots \\ \phi(x_n, F_n(x)) \end{pmatrix},$$

where ϕ is either ϕ_{NR} or ϕ_{FB} or any other Lipschitz continuous NCP function satisfying

$$\nu_1|\phi_{NR}(a,b)| \leq |\phi(a,b)| \leq \nu_2|\phi_{NR}(a,b)| \quad \forall (a,b)^T \in \Re^2 \tag{2.2.1}$$

for some constants $\nu_2 \geq \nu_1 > 0$. Then it is clear that the system of equations

$$H_F(x) = 0$$

is equivalent to NCP(F).

We note that ϕ_{NR} and ϕ_{FB} are not continuously differentiable at $\{ (a,b) \mid a = b \}$ and $\{ (a,b) \mid a = b = 0 \}$, respectively. Consequently, H_F is not continuously differentiable at some points, too. However, H_F is semismooth everywhere [23].

2.3 Local error bound in NCP

First, we recall the definition of the monotonicity and the strong monotonicity for the mapping F.

Definition 2.3.1 *The mapping $F : \Re^n \to \Re^n$ is said to be*

(i) monotone, if

$$(x-y)^T(F(x) - F(y)) \geq 0 \qquad \forall x, y \in \Re^n,$$

(ii) strongly monotone with modulus $\mu > 0$, if there exists $\mu > 0$ such that

$$(x-y)^T(F(x) - F(y)) \geq \mu\|x-y\|^2 \qquad \forall x, y \in \Re^n.$$

Note that, if F is monotone, then for any $c > 0$ and $d \in \Re^n$, the mapping $x \mapsto F(x) + cx$ is strongly monotone with modulus c.

The norm of the mapping H_F provides an error bound for NCP(F). Recall that $\|H_F(x)\|$ is said to provide a *local error bound* if, for any solution x^* of NCP(F), there exist positive constants b_1 and b_2 such that

$$\text{dist}\{x, X^*\} \leq b_2\|H_F(x)\| \qquad \forall x \in B(x^*, b_1), \tag{2.3.1}$$

where X^* denotes the solution set of NCP(F).

The following error bound results play an essential role in ensuring desirable convergence properties of iterative methods based on H_F for solving NCP(F).

Theorem 2.3.1 **(i)** *[42] Suppose that F is strongly monotone with modulus μ and Lipschitz continuous with constant L on a set $S \subseteq \Re^n$ containing the unique solution $\hat{x}$ of NCP(F). Then $\|H_F(x)\|$ provides an error bound for NCP(F) on S, that is,*

$$\|x - \hat{x}\| \leq \frac{K_1(L+1)}{\mu}\|H_F(x)\| \qquad \forall x \in S$$

holds, where K_1 is a constant independent of F.

(ii) *[39, 46] Suppose that F is affine and there exists a solution of NCP(F). Then $\|H_F(x)\|$ provides a local error bound for NCP(F).*

Note that Pang [42] shows (i) for the special case of $S = \Re^n$, but the proof in [42] can easily be extended to show the general case.

2.4 PPA for NCP

Among various algorithms for solving monotone NCP(F), the proximal point algorithm (PPA) [48] has nice convergence properties. Billups [2] and Yamashita and Fukushima [59] proposed the PPAs based on the mapping H_F. Billups' method regularizes the mapping H_F itself, whereas Yamashita and Fukushima's method regularizes the mapping F involved in NCP(F). In Chapters 5 and 6, we use the following PPA, which is a slight modification of Yamashita and Fukushima's method.

Algorithm PPA

Step 0: Choose parameters $\alpha \in (0,1), B \in (0,\infty)$ and an initial point $x^0 \in \Re^n$. Set $c_0 = 1$ and $k := 0$.

Step 1: If x^k satisfies a stopping criterion, then stop.

Step 2: Let $F^k : \Re^n \to \Re^n$ be given by

$$F^k(x) = F(x) + c_k(x - x^k).$$

Find an approximate solution $\tilde{x}^{k+1}$ of NCP(F^k) such that

$$\|H_{F^k}(\tilde{x}^{k+1})\| \leq (c_k)^4\|\tilde{x}^{k+1} - x^k\|, \tag{2.4.1}$$

$$\|\tilde{x}^{k+1} - [\tilde{x}^{k+1}]_+\| \leq B \tag{2.4.2}$$

and

$$\Psi^k([\tilde{x}^{k+1}]_+) \leq \frac{(c_k)^3}{4\max\{1, \|[\tilde{x}^{k+1}]_+\|^2\}} \tag{2.4.3}$$

where $[\cdot]_+$ denotes the projection onto the positive orthant $\Re^n_+$ and

$$\Psi^k(x) := \sum_{i=1}^{n}(|x_i F_i^k(x)| + |\min\{x_i, F_i^k(x)\}|).$$

Step 3: Set $x^{k+1} := [\tilde{x}^{k+1}]_+, c_{k+1} := \alpha c_k$, and $k := k+1$. Go to Step 2.

The PPA presented in [59] does not involve condition (2.4.2). This condition is included in the above algorithm to ensure the boundedness of $\{\tilde{x}^k\}$, which was tacitly assumed in the proof of [59, Theorem 3.3]. It can be shown that the presence of condition (2.4.2) does not affect the convergence properties of the PPA established in [59].

This algorithm enjoys nice convergence properties. Specifically, a sequence generated by Algorithm PPA converges to a solution of NCP(F) globally, and the distance between the generated sequence and the solution set of NCP(F) converges to 0 superlinearly under mild assumptions.

The following convergence theorem has been established in [59].

Theorem 2.4.1 *Suppose that F is monotone and locally Lipschitzian, and that NCP(F) has a solution. Then, the sequence $\{x^k\}$ generated by Algorithm PPA converges to a solution x^* of NCP(F). Moreover, if $\|H_F(x)\|$ provides a local error bound in a neighborhood of x^*, then $\{\mathrm{dist}(x^k, X^*)\}$ converges to 0 superlinearly, where X^* is the solution set of NCP(F).*

The most expensive task in Algorithm PPA is to solve NCP(F^k) in Step 2. In [59], it is proposed that NCP(F^k) be solved using the generalized Newton method (GNM) [12, 45]. Note that F^k is strongly monotone when F is monotone. Then, the GNM can find an approximate solution of subproblem NCP(F^k) rapidly when x^* is nondegenerate, as stated in the following theorem [59].

Theorem 2.4.2 *Suppose that x^* is a nondegenerate solution of NCP(F) and $\|H_F(x)\|$ provides a local error bound in a neighborhood of x^*. Assume that an iterate x^k is sufficiently close to x^*. Then a single iteration of the GNM for NCP(F^k) yields a point x^{k+1} that satisfies (2.4.1), (2.4.2) and (2.4.3).*

Chapter 3

An SQP Method for Large-Scale NLP

3.1 Introduction

The purpose of this chapter is to propose a method to solve the following nonlinear optimization problem (NLP) with equality and inequality constraints:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) = 0 \ (j \in J_E), \\ & g_j(x) \geq 0 \ (j \in J_I), \end{array} \tag{3.1.1}$$

where $f : \Re^n \to \Re^1$ and $g_j : \Re^n \to \Re^1$ $(j \in J_E \cup J_I)$ with J_E and J_I being finite index sets such that $J_E \cap J_I = \emptyset$. Throughout this chapter, we assume that f and g_j $(j \in J_E \cup J_I)$ are twice continuously differentiable.

In this chapter, we use the sequential quadratic programming (SQP) method as a basic framework to solve (3.1.1). Moreover, we construct an algorithm which is applicable to large-scale problems. The SQP method with quasi-Newton methods has been well known as one of effective methods for nonlinear optimization problems. However, it is not easy to apply quasi-Newton methods to large-scale optimization problems because the Hessian of the Lagrangian, which is sparse in many cases, is approximated by a dense matrix in quasi-Newton methods usually. On the contrary, the trust region SQP method is considered to be applicable to large-scale optimization problems because it uses the Hessian of the Lagrangian itself, so the sparsity of the Hessian is preserved.

In the ordinary trust region SQP method, the QP subproblem at a current point x^k is basically defined by

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} d^T G_k d + \nabla f(x^k)^T d \\ \text{subject to} & g_j(x^k) + \nabla g_j(x^k)^T d = 0 \ (j \in J_E), \\ & g_j(x^k) + \nabla g_j(x^k)^T d \geq 0 \ (j \in J_I), \\ & \|d\| \leq \delta_k, \end{array} \tag{3.1.2}$$

where G_k is the Hessian of the Lagrangian of (3.1.1) at x^k and δ_k is a radius of the trust region. The ordinary trust region SQP method has some difficulties in execution. One of such difficulties is that it is possible that (3.1.2) has no feasible region when δ_k is too small. In the algorithms which have been proposed so far, some parameters are introduced to make subproblems feasible [5, 6, 44, 52], or subproblems are solved in two steps [38, 40]. These modifications of algorithms are so complicated that it is not favorable for implementation and calculation time. Moreover, another difficulty is to solve QP subproblems which are not necessarily convex because G_k is not necessarily positive semidefinite.

In this chapter, we propose a new SQP method which eliminates these two difficulties. In our method, we solve two types of subproblem. One is a convex QP problem and the other is a system of linear equations. Our method is applicable to large-scale optimization problems. In addition, we note that our approach in this chapter resembles an approach of [56], which deals with the primal-dual interior point method.

This chapter is organized as follows: In Section 3.2, we introduce some basic concepts. In Section 3.3, we propose the new SQP method. In Section 3.4, we show the global convergence of the proposed method. In Section 3.5, we show some local property of the proposed method. In Section 3.6, we report numerical results by the proposed method. In Section 3.7, we make some concluding remarks and discuss future research topics.

3.2 Preliminaries

In this section, we introduce some basic concepts which are necessary in the following sections.

First, we prepare some notations. The Lagrangian of (3.1.1) is

$$L(x,y) = f(x) - \sum_{j \in J_E} y_j g_j(x) - \sum_{j \in J_I} y_j g_j(x),$$

where $y_j (j \in J_E \cup J_I)$ are Lagrange multipliers for $g_j(x) = 0$ $(j \in J_E)$ and $g_j(x) \geq 0$ $(j \in J_I)$. Together with this, the KKT conditions for (3.1.1) are

$$\begin{cases} \nabla_x L(x,y) = \nabla f(x) - \sum_{j \in J_E} y_j \nabla g_j(x) - \sum_{j \in J_I} y_j \nabla g_j(x) = 0, \\ g_j(x) = 0 \ (j \in J_E), \\ y_j g_j(x) = 0, \ y_j \geq 0, \ g_j(x) \geq 0 \ (j \in J_I). \end{cases} \tag{3.2.1}$$

The penalty function for (3.1.1) is defined by

$$F(x) = f(x) + \sum_{j \in J_E} \rho_j |g_j(x)| + \sum_{j \in J_I} \rho_j |\min\{0, g_j(x)\}|,$$

where ρ_j $(j \in J_E \cup J_I)$ is the penalty parameter which is sufficiently large. It is known that the local minimum of (3.1.1) is also that of $F(x)$ under appropriate conditions [28] .

The first-order approximation of F at x in the direction $d \in \Re^n$ is defined by

$$F_l(x;d) = f(x) + \nabla f(x)^T d + \sum_{j \in J_E} \rho_j |g_j(x) + \nabla g_j(x)^T d| + \sum_{j \in J_I} \rho_j |\min\{0, g_j(x) + \nabla g_j(x)^T d\}|.$$

In addition, we define the second-order approximation of F at x in the direction d by

$$F_q(x;d) = F_l(x;d) + \frac{1}{2} d^T G d,$$

where $G = \nabla_x^2 L(x, y)$. The difference of F, F_l and F_q in the direction d are

$$\begin{aligned}\Delta F(x;d) &= F(x+d) - F(x),\\ \Delta F_l(x;d) &= F_l(x;d) - F(x),\\ \Delta F_q(x;d) &= F_q(x;d) - F(x),\end{aligned}$$

respectively.

Next, we explain two types of QP subproblem which we solve at each iteration in our algorithm. In what follows, k denotes an iteration number of our algorithm.

One of the QP subproblems is

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}d^T D_k d + \nabla f(x^k)^T d\\ \text{subject to} & g_j(x^k) + \nabla g_j(x^k)^T d = 0 \ (j \in J_E),\\ & g_j(x^k) + \nabla g_j(x^k)^T d \geq 0 \ (j \in J_I),\end{array} \tag{3.2.2}$$

where D_k is a diagonal matrix whose diagonal elements are positive. d_{SD}^k and $y_{SD,j}^{k+1}$ $(j \in J_E \cup J_I)$ denote a solution and Lagrange multipliers of (3.2.2), respectively. As we will show in Lemma 3.4.1, d_{SD}^k is a descent direction of the penalty function F, because D_k is a positive definite matrix. This fact contributes greatly to the global convergence of our algorithm. Moreover, (3.2.2) is a convex QP problem, so we can use various methods to solve it . Here we assume that (3.2.2) is feasible. We consider the infeasible case in Remark 3.4.1.

The KKT conditions for (3.2.2) are

$$D_k d_{SD}^k + \nabla f(x^k) - \sum_{j \in J_E \cup J_I} y_{SD,j}^{k+1} \nabla g_j(x^k) = 0, \tag{3.2.3}$$

$$g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k = 0 \ (j \in J_E), \tag{3.2.4}$$

$$y_{SD,j}^{k+1}(g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k) = 0,\ y_{SD,j}^{k+1} \geq 0,\ g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k \geq 0 \ (j \in J_I). \tag{3.2.5}$$

Now, let J_{A_k} be the index set of active constraints of (3.2.2) at the solution d_{SD}^k, that is,

$$J_{A_k} := \{j \in J_E \cup J_I \mid g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k = 0\}.$$

As we will show in Lemma 3.5.1, when x^k is sufficiently close to a solution, J_{A_k} coincides with the index set of active constraints of (3.1.1) under appropriate assumptions.

The other QP subproblem is defined by

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}d^T G_k d + \nabla f(x^k)^T d\\ \text{subject to} & g_j(x^k) + \nabla g_j(x^k)^T d = 0 \ (j \in J_{A_k}),\end{array} \tag{3.2.6}$$

where $G_k = \nabla_x^2 L(x^k, y^k)$, d_N^k and $y_{N,j}^{k+1}$ $(j \in J_{A_k})$ denote a solution and Lagrange multipliers of (3.2.6), respectively. We note that (3.2.6) is feasible when (3.2.2) is feasible. The KKT conditions of (3.2.6) are

$$G_k d_N^k + \nabla f(x^k) - \sum_{j \in J_{A_k}} y_{N,j}^{k+1} \nabla g_j(x^k) = 0, \tag{3.2.7}$$

$$g_j(x^k) + \nabla g_j(x^k)^T d_N^k = 0 \ (j \in J_{A_k}). \tag{3.2.8}$$

As we mentioned, when x^k is sufficiently close to a solution, J_{A_k} coincides with the index set of active constraints at a solution of (3.1.1). Accordingly, in a neighborhood of a solution of (3.1.1), the conditions (3.2.7) and (3.2.8) are good approximation of the KKT conditions of (3.1.1), except for the condition of inactive constraints whose Lagrange multipliers' value are equal to 0. Thus we set $y_{N,j}^{k+1} = 0$ $(j \notin J_{A_k})$. So (d_N^k, y_N^{k+1}) is a favorable direction for fast convergence in a neighborhood of a solution.

We can rewrite (3.2.7) and (3.2.8) as

$$\begin{pmatrix} G_k & -\nabla g_J(x^k) \\ \nabla g_J(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} d_N^k \\ y_{N,J}^{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ -g_J(x^k) \end{pmatrix}, \tag{3.2.9}$$

where $g_J(x^k)$ and $y_{N,J}^{k+1}$ are vectors which are composed of $g_j(x^k)$ and $y_{N,j}^{k+1}$ $(j \in J_{A_k})$, respectively. We note that we can get a solution of (3.2.6) with less computational time than general QP problems, because the KKT conditions of (3.2.6) are equivalent to a system of linear equations (3.2.9). However, (3.2.9) may be ill-posed when, for example, G_k is singular and d_N^k is not bounded. We will consider this case in Remark 3.4.2.

3.3 Algorithm

The algorithm which we propose in this chapter is as follows.

Algorithm SQP-2SP

Step 0. Set an initial point $x^0 \in \Re^n$, a positive definite diagonal matrix $D_0 \in \Re^{n\times n}$, a symmetric matrix $G_0 \in \Re^{n\times n}$, and parameters $\delta_0 > 0$ and $M > 1$. Set $k = 0$.

Step 1. Compute (d_{SD}^k, y_{SD}^{k+1}) and (d_N^k, y_N^{k+1}) by solving (3.2.2) and (3.2.6), respectively. If

$$\|d_N^k\| \le M\|d_{SD}^k\| \tag{3.3.1}$$

is not satisfied, modify G_k to satisfy (3.3.1).

Step 2. If (x^k, y^{k+1}) satisfies (3.2.1), where

$$y^{k+1} := \begin{cases} y_N^{k+1}, & \text{if } y_{N,j}^{k+1} \ge 0 \ (\forall j \in J_{A_k} \cap J_I), \\ y_{SD}^{k+1}, & \text{otherwise}, \end{cases}$$

then stop.

Step 3. Find $s^k \in \Re^n$ which satisfies

$$\|s^k\| \le \delta_k, \tag{3.3.2}$$

$$\|s^k\| \le M\|d_{SD}^k\|, \tag{3.3.3}$$

$$\Delta F_q(x^k; s^k) \le \frac{1}{2}\Delta F_q(x^k; \alpha^k d_{SD}^k) \tag{3.3.4}$$

where

$$\alpha^k = \min\left\{1, \frac{\delta_k}{\|d_{SD}^k\|}, -\frac{\Delta F_l(x^k; d_{SD}^k)}{\max\{0, {d_{SD}^k}^T G_k d_{SD}^k\}}\right\}, \tag{3.3.5}$$

and the last term in the braces in the right hand side is assumed to give the value $+\infty$ if the value of the denominator is 0.

Step 4. Set δ_{k+1} as follows:

$$\begin{array}{rcl} \Delta F(x^k; s^k) > \frac{1}{4}\Delta F_q(x^k; s^k) & \Rightarrow & \delta_{k+1} = \frac{1}{2}\delta_k, \\ \Delta F(x^k; s^k) \leq \frac{3}{4}\Delta F_q(x^k; s^k) & \Rightarrow & \delta_{k+1} = 2\delta_k, \\ \text{otherwise} & \Rightarrow & \delta_{k+1} = \delta_k. \end{array}$$

Step 5. If $\Delta F(x^k; s^k) \leq 0$, set $x^{k+1} := x^k + s^k$, otherwise set $x^{k+1} := x^k$. Set $D_{k+1}, G_{k+1} := \nabla_x^2 L(x^{k+1}, y^{k+1})$ and $k := k + 1$. Go to Step 1.

In Remark 3.4.2, we will explain how G_k is modified when (3.3.1) is not satisfied in Step 1 of Algorithm SQP-2SP.

At this moment, we make a remark on a difference between the ordinary trust region SQP method and Algorithm SQP-2SP. In the ordinary trust region SQP method, we find a solution of a subproblem with a trust region constraint. Thus a solution satisfies a trust region constraint and a solution itself can be a candidate of the next iteration. On the other hand, in Algorithm SQP-2SP, we solve (3.2.2) and (3.2.6) as subproblems and a solution d_{SD}^k and d_N^k, respectively, does not necessarily satisfy a trust-region-like constraint (3.3.2). Thus, we have to make a candidate of the next iteration which satisfies (3.3.2). We will consider a concrete method for it in Remark 3.4.3.

3.4 Global convergence

In this section, we show that Algorithm SQP-2SP has the global convergence property.

First, we assume the following.

Assumption 3.4.1 **(1)** *The vectors* $\{y_{SD,j}^{k+1}\}$ *and* $\{y_{N,j}^{k+1}\}$ $(j \in J_E \cup J_I)$ *are bounded. In addition,* $\rho_j > |y_{SD,j}^{k+1}|$ $(j \in J_E \cup J_I)$ *holds for all* k.

(2) *The penalty function* $F(x)$ *is bounded below and its level set at the initial point* x^0*, that is,* $\{x \in \Re^n \mid F(x) \leq F(x^0)\}$*, is compact.*

(3) *The matrix* D_k *is uniformly positive definite and uniformly bounded. The matrix* G_k *is uniformly bounded.*

(4) *There exists a positive constant* $M > 0$ *which satisfies (3.3.1) for all* k.

Now we show the next lemma.

Lemma 3.4.1 *Suppose that Assumption 3.4.1 holds. Then* $\Delta F_l(x^k; d_{SD}^k) \leq 0$ *holds. If* $d_{SD}^k \neq 0$, $\Delta F_l(x^k; d_{SD}^k) < 0$ *holds.*

Proof: By the definition of ΔF_l, (3.2.3), (3.2.4) and (3.2.5), we have

$$\begin{aligned}
\Delta F_l(x^k; d_{SD}^k) &= \nabla f(x^k)^T d_{SD}^k + \sum_{j\in J_E} \rho_j^k (|g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k| - |g_j(x^k)|) \\
&\quad + \sum_{j\in J_I} \rho_j^k (|\min\{0, g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}|) \\
&= -{d_{SD}^k}^T D_k d_{SD}^k + \sum_{j\in J_E} y_{SD,j}^{k+1} \nabla g_j(x^k)^T d_{SD}^k + \sum_{j\in J_I} y_{SD,j}^{k+1} \nabla g_j(x^k)^T d_{SD}^k \\
&\quad - \Big[\sum_{j\in J_E} \rho_j^k |g_j(x^k)| + \sum_{j\in J_I} \rho_j^k |\min\{0, g_j(x^k)\}|\Big]. \qquad (3.4.1)
\end{aligned}$$

Moreover, from (3.2.4) and (3.2.5), we have

$$\sum_{j\in J_E} y_{SD,j}^{k+1} \nabla g_j(x^k)^T d_{SD}^k \leq \sum_{j\in J_E} \left|y_{SD,j}^{k+1}\right| \left|\nabla g_j(x^k)^T d_{SD}^k\right| = \sum_{j\in J_E} \left|y_{SD,j}^{k+1}\right| \left|g_j(x^k)\right|, \qquad (3.4.2)$$

$$\sum_{j\in J_I} y_{SD,j}^{k+1} \nabla g_j(x^k)^T d_{SD}^k = \sum_{j\in J_I} y_{SD,j}^{k+1}(-g_j(x^k)) \leq \sum_{j\in J_I} y_{SD,j}^{k+1} |\min\{0, g_j(x^k)\}|. \qquad (3.4.3)$$

It then follows from (3.4.1), (3.4.2) and (3.4.3) that

$$\begin{aligned}
\Delta F_l(x^k; d_{SD}^k) &\leq -{d_{SD}^k}^T D_k d_{SD}^k + \sum_{j\in J_E} (|y_{SD,j}^{k+1}| - \rho_j^k)|g_j(x^k)| + \sum_{j\in J_I} (y_{SD,j}^{k+1} - \rho_j^k)|\min\{0, g_j(x^k)\}| \\
&\leq -{d_{SD}^k}^T D_k d_{SD}^k,
\end{aligned}$$

where the last inequality follows from Assumption 3.4.1 (1). Therefore we obtain $\Delta F_l(x^k; d_{SD}^k) \leq 0$ from Assumption 3.4.1 (3). In particular, $\Delta F_l(x^k; d_{SD}^k) < 0$ holds when $d_{SD}^k \neq 0$. □

Lemma 3.4.1 shows that d_{SD}^k is a descent direction of the penalty function F. This property contributes greatly to the global convergence of Algorithm SQP-2SP.

Lemma 3.4.2 *Suppose that Assumption 3.4.1 holds. Then,*

$$\Delta F_l(x; \alpha d_{SD}^k) \leq \alpha \Delta F_l(x; d_{SD}^k), \quad \forall \alpha \in [0,1]$$

holds.

Proof: First, we consider the index $j \in J_E$. From (3.2.4), we have

$$\begin{aligned}
&|g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k| - |g_j(x^k)| \\
&= |(1-\alpha) g_j(x^k) + \alpha(g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k)| - |g_j(x^k)| \\
&= -\alpha |g_j(x^k)| \\
&= \alpha \left(|g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k| - |g_j(x^k)|\right) \quad (j \in J_E). \qquad (3.4.4)
\end{aligned}$$

Next, we consider the index $j \in J_I$ in two cases: (i) $g_j(x^k) \geq 0$ and (ii) $g_j(x^k) < 0$.

Case (i): In this case, $g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k \geq 0$ holds from (3.2.5) and $\alpha \in [0, 1]$. Thus we have

$$\begin{aligned} & |\min\{0, g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \\ = \;& 0 \\ = \;& \alpha \left(|\min\{0, g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \right). \end{aligned}$$

Case (ii): In this case, we consider two cases: (a) $g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k \geq 0$ and (b) $g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k < 0$.

Case (a): By (3.2.5) and $\alpha \in [0, 1]$, we have

$$\begin{aligned} & |\min\{0, g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \\ = \;& 0 - |\min\{0, g_j(x^k)\}| \\ \leq \;& -\alpha |\min\{0, g_j(x^k)\}| \\ = \;& \alpha \left(|\min\{0, g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \right). \end{aligned}$$

Case (b): In this case, $\nabla g_j(x^k)^T d_{SD}^k \geq -g_j(x^k) > 0$ holds from (3.2.5). Thus we have

$$\begin{aligned} & |\min\{0, g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \\ = \;& -(g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k) + g_j(x^k) \\ = \;& -\alpha \nabla g_j(x^k)^T d_{SD}^k \\ \leq \;& \alpha g_j(x^k) \\ = \;& -\alpha |\min\{0, g_j(x^k)\}| \\ = \;& \alpha \left(|\min\{0, g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \right). \end{aligned}$$

Now we summarize the results mentioned above. For $j \in J_I$, we obtain

$$\begin{aligned} & |\min\{0, g_j(x^k) + \alpha \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \\ \leq \;& \alpha \left(|\min\{0, g_j(x^k) + \nabla g_j(x^k)^T d_{SD}^k\}| - |\min\{0, g_j(x^k)\}| \right) \quad (j \in J_I). \end{aligned} \tag{3.4.5}$$

Then, from (3.4.4) and (3.4.5), we have

$$\begin{aligned} \Delta F_l(x^k; \alpha d_k) &= \alpha \nabla f(x^k)^T d_k + \sum_{j \in J_E} \rho_j^k (|g_j(x^k) + \alpha \nabla g_j(x^k)^T d_k| - |g_j(x^k)|) \\ &\quad + \sum_{j \in J_I} \rho_j^k (|\min\{0, g_j(x^k) + \alpha \nabla g_j(x^k)^T d_k\}| - |\min\{0, g_j(x^k)\}|) \\ &\leq \alpha \Big[\nabla f(x^k)^T d_k + \sum_{j \in J_E} \rho_j^k (|g_j(x^k) + \nabla g_j(x^k)^T d_k| - |g_j(x^k)|) \\ &\quad + \sum_{j \in J_I} \rho_j^k (|\min\{0, g_j(x^k) + \nabla g_j(x^k)^T d_k\}| - |\min\{0, g_j(x^k)\}|) \Big] \\ &= \alpha \Delta F_l(x^k; d_k). \end{aligned}$$

This completes the proof. □

Lemma 3.4.3 *Suppose that Assumption 3.4.1 holds. Then,*

$$\Delta F_q(x^k;\alpha^k d_{SD}^k) \le \frac{1}{2}\alpha^k \Delta F_l(x^k;d_{SD}^k).$$

Proof: From Lemma 3.4.2, we obtain

$$\begin{aligned}\Delta F_q(x^k;\alpha^k d_{SD}^k) &= \Delta F_l(x^k;\alpha^k d_{SD}^k) + \frac{1}{2}(\alpha^k)^2 {d_{SD}^k}^T G_k d_{SD}^k \\ &\le \alpha^k \Delta F_l(x^k;d_{SD}^k) + \frac{1}{2}(\alpha^k)^2 {d_{SD}^k}^T G_k d_{SD}^k. \end{aligned} \qquad (3.4.6)$$

Moreover, we have from Lemma 3.4.1 that $\Delta F_l(x^k;d_{SD}^k) \le 0$.

To show this lemma, we consider two cases: (i) ${d_{SD}^k}^T G_k d_{SD}^k > 0$ and (ii) ${d_{SD}^k}^T G_k d_{SD}^k \le 0$.

Case (i): It follows from (3.3.5) that $0 \le \alpha^k \le -\frac{\Delta F_l(x^k;d_{SD}^k)}{{d_{SD}^k}^T G_k d_{SD}^k}$. Thus we have from (3.4.6) that

$$\Delta F_q(x^k;\alpha^k d_{SD}^k) \le \frac{1}{2}\alpha^k \Delta F_l(x^k;d_{SD}^k).$$

Case (ii): From (3.4.6), we obtain

$$\begin{aligned}\Delta F_q(x^k;\alpha^k d_{SD}^k) &\le \alpha^k \Delta F_l(x^k;d_{SD}^k) \\ &\le \frac{1}{2}\alpha^k \Delta F_l(x^k;d_{SD}^k).\end{aligned}$$

This completes the proof. □

Lemma 3.4.4 *Suppose that Assumption 3.4.1 holds. Then,* $\liminf_{k\to\infty} \|d_{SD}^k\| = 0$ *holds.*

Proof: First, from Step 3 of Algorithm SQP-2SP and Lemmas 3.4.1 and 3.4.3, we obtain

$$\begin{aligned}\Delta F_q(x^k;s^k) &\le \frac{1}{2}\Delta F_q(x^k;\alpha^k d_{SD}^k) \\ &\le \frac{1}{4}\alpha^k \Delta F_l(x^k;d_{SD}^k) \\ &= \frac{1}{4}\min\left\{1, \frac{\delta_k}{\|d_{SD}^k\|}, -\frac{\Delta F_l(x^k;d_{SD}^k)}{\max\{0, {d_{SD}^k}^T G_k d_{SD}^k\}}\right\}\Delta F_l(x^k;d_{SD}^k) \le 0. \end{aligned} \qquad (3.4.7)$$

Next, we define the sets K_1 and K_2 by

$$\begin{aligned} K_1 &:= \{k \in \{0,1,...\} \mid \Delta F(x^k;s^k) > \frac{1}{4}\Delta F_q(x^k;s^k)\}, \qquad (3.4.8)\\ K_2 &:= \{k \in \{0,1,...\} \mid \Delta F(x^k;s^k) \le \frac{1}{4}\Delta F_q(x^k;s^k)\}. \end{aligned}$$

Now we assume that $\liminf_{k\to\infty}\|d_{SD}^k\| > 0$. Then, from Lemma 3.4.1 and Assumption 3.4.1 (3), there exists a positive constant ϵ_1 such that

$$\liminf_{k\to\infty} |\Delta F_l(x^k;d_{SD}^k)| > \epsilon_1. \qquad (3.4.9)$$

Moreover, from Assumption 3.4.1 (3), there exists a positive constant R_1 for all k such that

$$\|d_{SD}^k\| < R_1. \qquad (3.4.10)$$

In addition, from Assumption 3.4.1 (3) and (3.4.10), there exists a positive constant R_2 for all k such that

$$|{d^k_{SD}}^T G_k d^k_{SD}| < R_2. \tag{3.4.11}$$

Now we consider two cases: (i) $\limsup_{k\to\infty} \delta_k = 0$ and (ii) $\limsup_{k\to\infty} \delta_k > 0$.

Case (i): This case is equivalent to $\lim_{k\to\infty} \delta_k = 0$. Thus we have from Step 3 of Algorithm SQP-2SP that $\lim_{k\to\infty} \|s^k\| = 0$. Moreover, from Step 4 of Algorithm SQP-2SP, K_1 has an infinite number of elements. Now we have from the definition of ΔF and (3.4.8) that

$$\begin{aligned}\Delta F(x^k;s^k) &= \Delta F_l(x^k;s^k) + O(\|s^k\|^2) \\ &= \Delta F_q(x^k;s^k) + O(\|s^k\|^2) > \frac{1}{4}\Delta F_q(x^k;s^k) \quad (k \in K_1)\end{aligned}$$

when k is large enough, where $M = O(A)$ means that there exists a positive constant β such that $\|M\| \le \beta A$. From this fact and $\Delta F_q(x^k;s^k) < 0$, we have

$$|\Delta F_q(x^k;s^k)| = O(\|s^k\|^2) \quad (k \in K_1) \tag{3.4.12}$$

when k is large enough. On the other hand, we obtain from (3.4.9) and (3.4.11) that

$$\liminf_{k\to\infty}\left(-\frac{\Delta F_l(x^k;d^k_{SD})}{\max\{0, {d^k_{SD}}^T G_k d^k_{SD}\}}\right) > \frac{\epsilon_1}{R_2} > 0.$$

From this fact, we have that $\alpha^k = \delta_k/\|d^k_{SD}\|$ $(k \in K_1)$ when k is large enough. Therefore we obtain from (3.4.7) and (3.4.10) that

$$\begin{aligned}|\Delta F_q(x^k;s^k)| &\ge \frac{1}{4}\alpha^k|\Delta F_l(x^k;d^k_{SD})| \\ &\ge \frac{1}{4}\frac{\delta_k}{\|d^k_{SD}\|}\epsilon_1 \\ &\ge \frac{\epsilon_1}{4R_1}\|s^k\|.\end{aligned} \tag{3.4.13}$$

Thus we obtain from (3.4.10) that

$$|\Delta F_q(x^k;s^k)| = \Omega(\|s^k\|), \tag{3.4.14}$$

where $M = \Omega(A)$ means that there exists a positive constant $\beta > 0$ such that $\|M\| \ge \beta A$. However, (3.4.14) contradicts (3.4.12).

Case (ii): In this case, from Step 4 in Algorithm SQP-2SP, K_2 has an infinite number of elements. From (3.4.7), there exists α^*_k such that

$$\Delta F(x^k;s^k) \le \frac{1}{4}\Delta F_q(x^k;s^k) \le \frac{\alpha^k}{16}\Delta F_l(x^k;d^k_{SD}) < 0 \quad (k \in K_2).$$

Moreover, $\{F(x^k)\}$ is bounded below and nonincreasing because of Assumption 3.4.1 (2) and Step 5 of Algorithm SQP-2SP. Thus we have

$$F(x^{k+1}) - F(x^k) = \Delta F(x^k; s^k) \to 0 \quad (k \to \infty, k \in K_2).$$

Hence, from (3.4.9), we obtain that $\alpha^k \to 0$ $(k \in K_2)$. Now we consider two cases: (a) K_1 has finite elements and (b) K_1 has infinite elements.

Case (a): From Step 4 of Algorithm SQP-2SP, $\liminf_{k\to\infty, k\in K_2} \delta_k > 0$ holds. Therefore, from (3.4.11) and (3.3.5), we have $\Delta F_l(x_k; dx^k_{SD}) \to 0$ $(k \in K_2)$. However, this contradicts (3.4.9).

Case (b): Now we assume that $\limsup_{k\to\infty, k\in K_2} \delta_k = 0$. From this assumption and Step 4 of Algorithm SQP-2SP, we have

$$0 \le \limsup_{k\to\infty, k\in K_1} \delta_k \le 2 \cdot \limsup_{k\to\infty, k\in K_2} \delta_k = 0,$$

so we obtain that $\limsup_{k\to\infty, k\in K_1} \delta_k = 0$. Therefore we have $\limsup_{k\to\infty} \delta_k = 0$. However, this contradicts the assumption of Case (ii). Hence, we have $\limsup_{k\to\infty, k\in K_2} \delta_k > 0$. Therefore, from (3.4.11) and (3.3.5), we have $\liminf_{k\to\infty, k\in K_2} \Delta F_l(x_k; dx^k_{SD}) = 0$. However, this contradicts (3.4.9).

Therefore, we conclude $\liminf_{k\to\infty} \|d^k_{SD}\| = 0$. □

Theorem 3.4.1 *Suppose that Assumption 3.4.1 holds. Let $\{x^k\}$ be a sequence generated by Algorithm SQP-2SP. Then, $\{x^k\}$ has an accumulation point. Moreover, there exists an accumulation point x^* of $\{x^k\}$, which has an appropriate Lagrange multiplier y^* such that (x^*, y^*) satisfies the KKT conditions for (3.1.1).*

Proof: First, from Step 5 of Algorithm SQP-2SP, $\{F(x^k)\}$ is nonincreasing. Then, from Assumption 3.4.1 (2), $\{x^k\}$ is in a compact set, thus $\{x^k\}$ has an accumulation point x^*. Now, without loss of generality, we assume that a subsequence $\{x^k\}_{k\in K_3}$ converges to x^* and $\lim_{k\to\infty, k\in K_3} \|d^k_{SD}\| = 0$ from Lemma 3.4.4. Moreover, since the number of elements of $J_E \cup J_I$ is finite, then possible combinations of elements which are included in J_{A_k} are finite. Therefore there exists an infinite set $K_4 \subseteq K_3$ such that $J_{A_k} = J_{\bar{A}}$ $(k \in K_4)$ for a particular index set $J_{\bar{A}}$. Thus we have $J_{A_k} = J_{\bar{A}}$ $(k \in K_4)$, $\lim_{k\to\infty, k\in K_4} x^k = x^*$ and $\lim_{k\to\infty, k\in K_4} \|d^k_{SD}\| = 0$. Moreover, from (3.3.1), we have $\lim_{k\to\infty, k\in K_4} \|d^k_N\| = 0$.

To show this theorem, we consider two cases: (i) there exists k_0 such that $y^{k+1}_{N,j} \ge 0$ $(j \in J_{A_k} \cap J_I, k \ge k_0, k \in K_4)$ and (ii) the case except (i).

Case (i): We note that $y^{k+1}_{N,j} = 0$ $(j \notin J_{A_k}, k \ge k_0, k \in K_4)$. For $k \ge k_0$, we have from (3.2.5), (3.2.7), (3.2.8) and the definition of J_{A_k} that

$$\begin{aligned}
&G_k d^k_N + \nabla f(x^k) - \textstyle\sum_{j\in J_E} y^{k+1}_{N,j} \nabla g_j(x^k) - \sum_{j\in J_I} y^{k+1}_{N,j} \nabla g_j(x^k) = 0,\\
&g_j(x^k) + \nabla g_j(x^k)^T d^k_N = 0 \ (j \in J_E),\\
&y^{k+1}_{N,j}(g_j(x^k) + \nabla g_j(x^k)^T d^k_N) = 0,\ y^{k+1}_{N,j} \ge 0,\ g_j(x^k) + \nabla g_j(x^k)^T d^k_N = 0 \ (j \in J_{\bar{A}} \cap J_I),\\
&y^{k+1}_{N,j}(g_j(x^k) + \nabla g_j(x^k)^T d^k_{SD}) = 0,\ y^{k+1}_{N,j} = 0,\ g_j(x^k) + \nabla g_j(x^k)^T d^k_{SD} > 0 \ (j \notin J_{\bar{A}}).
\end{aligned} \tag{3.4.15}$$

Now, from Assumption 3.4.1 (1), $\{y_N^{k+1}\}_{k\in K_4}$ has an accumulation point y_N^*. Thus, without loss of generality, we assume that $\{y_N^{k+1}\} \to y_N^*$ $(k \to \infty, k \in K_4)$. Therefore, from $\lim_{k\to\infty,k\in K_4} \|d_{SD}^k\| = \lim_{k\to\infty,k\in K_4} \|d_N^k\| = 0$, we obtain that (x^*, y_N^*) satisfies (3.2.1) when $k \to \infty$ $(k \in K_4)$ in (3.4.15).

Case (ii): In this case, there exists an infinite number of k such that $y^{k+1} = y_{SD}^{k+1}$. Thus, without loss of generality, $y^{k+1} = y_{SD}^{k+1}$ holds when k is sufficiently large. So we have (3.2.3), (3.2.4) and (3.2.5) when k is sufficiently large. Moreover, from Assumption 3.4.1 (1), $\{y_{SD}^{k+1}\}_{k\in K_4}$ has an accumulation point y_{SD}^*. Now we assume without loss of generality that $\{y_{SD}^{k+1}\} \to y_{SD}^*$ $(k \to \infty)$. Therefore, from $\lim_{k\to\infty,k\in K_4} \|d_{SD}^k\| = 0$, we obtain that (x^*, y_{SD}^*) satisfies (3.2.1) when $k \to \infty$ $(k \in K_4)$ in (3.2.3), (3.2.4) and (3.2.5).

This completes the proof. □

Remark 3.4.1 *Problem (3.2.2) may be infeasible even if the original problem (3.1.1) is feasible. In such a case, we solve the following problem instead of (3.2.2):*

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}d^T D_k d + \nabla f(x^k)^T d + \sum_{j\in J_E} \rho_j(\xi_j^+ + \xi_j^-) + \sum_{j\in J_I} \rho_j \eta_j \\ \text{subject to} & g_j(x^k) + \nabla g_j(x^k)^T d + \xi_j^+ - \xi_j^- = 0 \ (j \in J_E), \\ & g_j(x^k) + \nabla g_j(x^k)^T d + \eta_j \geq 0 \ (j \in J_I), \\ & \xi_j^+ \geq 0, \ \xi_j^- \geq 0 \ (j \in J_E), \\ & \eta_j \geq 0, \ (j \in J_I), \end{array} \tag{3.4.16}$$

where ξ_j^+, ξ_j^- $(j \in J_E)$ and η_j $(j \in J_I)$ are elastic variables. Moreover, active constraints and active elastic variables in (3.4.16) are treated as counterparts of constraints composing J_{A_k}, and we construct a subproblem which corresponds to (3.2.6). To show the validity of considering (3.4.16), we confirm the equivalence of following problems at first:

$$\text{minimize} \ |h(x)| \quad \Leftrightarrow \quad \begin{array}{ll} \text{minimize} & \xi^+ + \xi^- \\ \text{subject to} & h(x) + \xi^+ - \xi^- = 0, \\ & \xi^+ \geq 0, \ \xi^- \geq 0, \end{array}$$

$$\text{minimize} \ |\min\{0, h(x)\}| \quad \Leftrightarrow \quad \begin{array}{ll} \text{minimize} & \eta \\ \text{subject to} & h(x) + \eta = 0, \\ & \eta \geq 0, \end{array}$$

where $h : \Re^n \to \Re^1$. From these properties, we can easily reformulate (3.4.16) as

$$\text{minimize} \quad \bar{F}_l(x^k; d) = F_l(x^k; d) + \tfrac{1}{2}d^T D_k d. \tag{3.4.17}$$

In what follows, $\bar{d}_{SD}^k$ denotes a solution of (3.4.17). It follows from $\bar{F}_l(x^k; \bar{d}_{SD}^k) \leq \bar{F}_l(x^k; 0)$ that

$$F_l(x^k; \bar{d}_{SD}^k) + \frac{1}{2}(\bar{d}_{SD}^k)^T D_k \bar{d}_{SD}^k \leq F_l(x^k; 0) = F(x^k),$$

then we have

$$\Delta F_l(x^k; \bar{d}_{SD}^k) \leq -\frac{1}{2}(\bar{d}_{SD}^k)^T D_k \bar{d}_{SD}^k \leq 0.$$

This result is a counterpart of Lemma 3.4.1. Moreover, in the same way of this section, we can show the same property as Theorem 3.4.1 when we solve (3.4.16) instead of (3.2.2). However, a complete proof will be complicated a little bit, so it is eliminated here.

Remark 3.4.2 *When G_k is singular, we may fail to find a solution of (3.2.9). Moreover, when (3.2.9) is ill-posed, $\|d_N^k\|$ would be so large that (3.3.1) may be violated. In such cases, G_k is replaced by $\bar{G}_k$ which is defined by*

$$\bar{G}_k := G_k + \mu_k I,$$

where μ_k is a positive parameter. When μ_k is sufficiently large, $\bar{G}_k$ is nonsingular and (3.2.9) is well-posed, so (3.3.1) is satisfied. Practically, we choose a tiny positive value on μ_k at first. If (3.3.1) is not satisfied, we increase the value of μ_k gradually.

Remark 3.4.3 *In Step 3 of Algorithm SQP-2SP, we can choose any s^k which satisfies (3.3.2), (3.3.3) and (3.3.4). Now we consider an algorithm to construct s^k which is valid.*

For a convex combination of d_{SD}^k and d_N^k, that is,

$$\bar{d}^k(\nu_k) := \nu_k d_{SD}^k + (1-\nu_k) d_N^k, \quad \nu_k \in [0,1],$$

$\bar{\alpha}^k(\nu_k)$ is defined by

$$\bar{\alpha}^k(\nu_k) := \min\left\{1, \frac{\delta_k}{\|\bar{d}^k(\nu_k)\|}, -\frac{\Delta F_l(x^k; \bar{d}^k(\nu_k))}{\max\{0, \bar{d}^k(\nu_k)^T G_k \bar{d}^k(\nu_k)\}}\right\},$$

where the last term in the braces is assumed to give the value $+\infty$ if the value of the denominator is 0. Now we consider $\bar{s}_k(\nu_k) := \bar{\alpha}^k(\nu_k)\bar{d}^k(\nu_k)$. It is easy to show that $\bar{s}_k(1) = \bar{\alpha}^k(1)\bar{d}^k(1)$ satisfies (3.3.2), (3.3.3) and (3.3.4). Moreover, as we will see in Section 3.5, $\bar{d}^k(0)$ is equivalent to the search direction of the ordinary SQP method under some assumptions, so $\bar{d}^k(0)$ is a good direction for fast convergence practically. Therefore, in view of efficient implementation, we set 0 on ν_k at first, and repeatedly increase the value of ν_k by 0.1 until $\bar{s}^k(\nu_k)$ satisfies (3.3.2), (3.3.3) and (3.3.4).

Remark 3.4.4 *In this paper, we assume that the penalty parameter ρ_j $(j \in J_E \cup J_I)$ satisfies Assumption 3.4.1 (1). This assumption is tactically prepared for the proofs in this section, so we should update the value of ρ_j $(j \in J_E \cup J_I)$ at each iteration with an appropriate rule when we implement Algorithm SQP-2SP. For example, we can adopt the following update rule:*

- *When $k = 0$, we initialize $\rho_{k,j} := \max\{L\,|y_{SD,j}^k|,\ \rho^{min}\}$ $(j \in J_E \cup J_I)$,*
- *When $k > 0$, we set $\rho_{k,j} := \max\{L\,|y_{SD,j}^k|,\ \rho_{k-1,j}\}$ $(j \in J_E \cup J_I)$,*

where $\rho_{k,j}$ $(j \in J_E \cup J_I)$ denotes the penalty parameter at the iteration k, L and ρ^{min} are constants such that $L > 1$ and $\rho^{min} > 0$. The adoption of this update rule does not affect the proofs in this section.

3.5 Local convergence

In this section, we consider some property of Algorithm SQP-2SP in a neighborhood of a solution.

First, we assume the following.

Assumption 3.5.1 **(1)** *The sequence* $\{(x^k, y^{k+1})\}$ *which is generated by Algorithm SQP-2SP converges to* (x^*, y^*)*, and* (x^*, y^*) *satisfies (3.2.1). Moreover,* $\lim_{k\to\infty} \|d^k_{SD}\| = 0$ *holds and* $\{y^{k+1}_{SD}\}$ *converges to* y^*_{SD}*.*

(2) *The linear independence constraint qualification holds at* (x^*, y^*)*. Moreover, the strict complementarity in (3.2.1), that is,*

$$y_j g_j(x) = 0,\ y_j \geq 0,\ g_j(x) \geq 0,\ y_j + g_j(x) > 0 \quad (j \in J_I)$$

is satisfied at (x^*, y^*)*.*

From the investigation in Section 3.4, Assumption 3.5.1 (1) is not unrealistic. Moreover, Assumption 3.5.1 (2) is not so restrictive.

Let J_{A_*} be the index set of active constraints at x^*, that is,

$$J_{A_*} := \{j \in J_E \cup J_I \mid g_j(x^*) = 0\}.$$

We show that the next lemma holds under Assumptions 3.4.1 and 3.5.1.

Lemma 3.5.1 *Suppose that Assumptions 3.4.1 and 3.5.1 hold. Then,* $J_{A_k} = J_{A_*}$ *holds for sufficiently large* k*.*

Proof: From Assumption 3.5.1 (2), Lagrange multiplier which satisfies (3.2.1) at x^* is unique, thus $y^*_{SD} = y^*$ holds from Assumption 3.5.1 (1). If we assume that $J_{A_k} = J_{A_*}$ does not hold for sufficiently large k, one of the following cases holds:

(i) $y^{k+1}_{SD,j} = 0$ for any j which satisfies $y^*_{SD,j} > 0$,

(ii) $y^{k+1}_{SD,j} > 0$ for any j which satisfies $y^*_{SD,j} = 0$.

First, the case (i) does not occur because of $\lim_{k\to\infty} y^{k+1}_{SD,j} \neq y^*_{SD,j}$. Now we consider the case (ii). We have $g_j(x^k) + \nabla g_j(x^k)^T d^k_{SD} = 0$ for sufficiently large k. Thus $g_j(x^*) = 0$ holds from Assumption 3.5.1 (1). However, this contradicts Assumption 3.5.1 (2) because of $y^*_{SD,j} = 0$ in this case. This completes the proof. □

Let us consider the ordinary SQP method (without trust region constraints). The subproblem of this method is as follows:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} d^T G_k d + \nabla f(x^k)^T d,\ d \in \Re^n \\ \text{subject to} & g_j(x^k) + \nabla g_j(x^k)^T d = 0\ (j \in J_E), \\ & g_j(x^k) + \nabla g_j(x^k)^T d \geq 0\ (j \in J_I). \end{array} \qquad (3.5.1)$$

We can easily show from Lemma 3.5.1 that the KKT conditions of (3.5.1) coincide with (3.2.7) and (3.2.8), which are the KKT conditions of (3.2.6) for sufficiently large k. Thus we obtain $d_N^k = d_{SQP}^k$, where d_{SQP}^k is a solution of (3.5.1). It is known that the SQP method has fast local convergence under certain conditions. Consequently, we can expect that d_N^k is a good direction for fast convergence in a neighborhood of a solution.

3.6 Numerical results

In this section, we present numerical results of Algorithm SQP-2SP.

3.6.1 Implementation and parameters

We coded Algorithm SQP-2SP in C++ and tested this implementation on various problems from Hock and Schittkowski's book [35] and CUTE/CUTEr problem archive [4, 33].

In this experiment, we employ the methods which are explained in Remarks 3.4.1, 3.4.2, 3.4.3 and 3.4.4. Especially, in Remark 3.4.2, we set $\mu_k = 10^{-10}$ at first and replace μ_k with $\mu_k := 2\mu_k$ until (3.3.1) is satisfied, and in Remark 3.4.4, we set $L = 1.2$ and $\rho^{min} = 10^{-6}$. Moreover, we employ the Gill–Murray method [30] and the supernodal right-looking method [49] to solve (3.2.2) and (3.2.9) in Step 1 of Algorithm SQP-2SP, respectively.

All experiments were executed with following parameters and initial settings: $G_0 := \nabla^2 f(x)$, $(D_k)_{ii} := \max\{|(G_k)_{ii}|,\ 10^{-3}\}$ $(i = 1, 2, \ldots, n)$, $\delta_0 = \max\{\|d_{SD}^0\|, \|d_N^0\|\} \times 10^2$ and $M = 10^5$.

Moreover, we stopped Algorithm SQP-2SP in Step 2 successfully when the following conditions are satisfied:

$$R = \max\{R_1, R_2, R_3, R_4, R_5\} \leq \epsilon, \qquad (3.6.1)$$

where

$$\begin{aligned} R_1 &= \|\nabla_x L(x,y)\|_1 \ / \ \max\{1, n\|\nabla f(x)\|\}, \\ R_2 &= \sum_{j \in J_E} |g_j(x)| \ / \ |J_E|, \\ R_3 &= \sum_{j \in J_I} |y_j g_j(x)| \ / \ |J_I|, \\ R_4 &= \sum_{j \in J_I} |\min\{0, y_j\}|, \\ R_5 &= \sum_{j \in J_I} |\min\{0, g_j(x)\}| \end{aligned}$$

and $\epsilon = \sqrt{2} \times 10^{-6}$. We can consider R as an indicator of optimality and feasibility. Moreover, when (3.6.1) is not satisfied until Algorithm SQP-2SP is iterated 150 times, we stop the calculation abnormally.

3.6.2 Hock & Schittkowski's problems

Hock & Schittkowski's book [35] contains 115 mathematical programming problems. All problems are small (1 – 16 variables and 0 – 21 constraints), and all problems has nonlinearity. We try to solve these problems by applying Algorithm SQP-2SP.

Table 3.1: Results for Hock & Schittkowski's problems

Number of solvable problems	113
Number of failed problems	2

Table 3.1 shows the result for Hock & Schittkowski's problems. Algorithm SQP-2SP solves all problems except Problems No.13 and No.87. With regard to Problem No.13, none of the known constraint qualifications hold at a solution, so it seems that numerical difficulties have occurred. Moreover, an objective function of Problem No.87 is not continuously differentiable.

3.6.3 CUTE/CUTEr problems

CUTE/CUTEr [4, 33] is the biggest archive of mathematical programming problems, as far as we know. We try to solve 55 problems which are treated as typical problems in [4, Figures 1, 2].

Table 3.2 shows the result for problems whose optimal solution is calculated successfully by Algorithm SQP-2SP. In Table 3.2, "# vars.", "# cons.", "val. obj.", "# iter.", "val. R" and "time(sec.)" mean the number of variables, the number of constraints, the value of objective function at an optimal solution, the number of Algorithm SQP-2SP's iterations, the value of R which is defined by (3.6.1) and the calculation time(sec.), respectively.

We can find from Table 3.2 that Algorithm SQP-2SP works well. Especially, Algorithm SQP-2SP can solve problems which have some thousands variables and constraints.

Table 3.3 shows the problems whose optimal solution can not be calculated by Algorithm SQP-2SP.

Table 3.2: Results for CUTE/CUTEr problems

Problem name	# vars.	# cons.	val. obj.	# iter.	val. R	time(sec.)
AIRCRFTA	8	5	0.000000e+00	2	1.22e-06	0.00
BDVALUE	5002	5000	0.000000e+00	1	4.16e-08	0.49
BIGBANK	2230	1113	0.000000e+00	1	2.03e-16	2.63
BRATU2D	3200	2888	0.000000e+00	4	4.33e-12	43.69
BRATU2D	5184	4900	0.000000e+00	16	1.82e-09	893.99
BRIDGEND	2734	2727	4.056176e+01	1	2.68e-15	1.43
CHANDHEQ	100	100	0.000000e+00	9	9.54e-07	0.82
CHEMRCTA	5000	5000	0.000000e+00	70	6.95e-11	119.09
CLPLATEA	5041	0	-1.259209e-02	7	1.49e-06	351.64
DALLASL	906	667	-2.026039e+05	13	2.88e-07	1.39
EXPFITC	5	502	2.330257e-02	118	5.66e-10	0.74
GAUSSELM	506	1135	-1.000000e+00	21	1.56e-17	3.66
GRIDNETA	7564	3844	4.779795e+02	2	1.81e-16	28.37
HAGER4	5001	2500	2.794102e+00	3	8.27e-07	70.45
HIMMELBI	100	12	-1.755000e+03	2	3.74e-10	0.08
HIMMELBJ	45	14	-1.903851e+03	7	6.53e-18	0.12
HIMMELBK	24	13	5.000000e-02	37	3.41e-20	0.51
HS106	8	6	7.049248e+03	8	3.43e-10	0.01
HS107	9	6	5.055012e+03	6	1.26e-07	0.00
HS114	10	11	-1.768807e+03	6	1.09e-07	0.00
HS116	13	14	9.759103e+01	6	2.33e-13	0.01
HS68	4	2	-9.204250e-01	13	5.46e-08	0.01
HS73	4	3	2.989438e+01	3	1.33e-07	0.00
HS88	2	1	1.362657e+00	16	2.92e-10	0.04
HS93	6	2	1.350760e+02	5	9.10e-07	0.00
HYDCAR20	99	99	0.000000e+00	8	3.89e-07	0.07
HYDROELL	1009	1008	-3.585547e+06	2	3.51e-15	0.58
KOWOSB	4	0	3.078009e-04	16	3.18e-07	0.01
LCH	3000	1	-3.098858e+00	2	4.34e-14	30.90
MSQRTA	1024	1024	0.000000e+00	4	2.73e-09	122.53
NLMSURF	5625	0	3.894898e+01	39	3.20e-07	1927.46
ORTHREGA	8197	4096	2.665546e+04	9	7.20e-07	406.38
PALMER1C	8	0	9.760505e-02	2	6.98e-08	0.00
PENTAGON	6	15	1.365217e-04	10	1.56e-08	0.01
PRODPL1	60	29	3.573897e+01	7	3.55e-10	0.02
ROSENBR	2	0	2.232923e-16	21	8.90e-07	0.01
SSEBNLN	194	96	1.617060e+07	1	1.33e-10	0.01
STEENBRE	540	126	0.000000e+00	1	8.22e-14	0.92
TRIGGER	7	6	0.000000e+00	17	5.23e-07	0.01
VAREIGVL	50	0	1.256894e-11	17	7.89e-07	0.09

Table 3.3: CUTE/CUTEr problems which can not be solved by Algorithm SQP-2SP

Problem name	# vars.	# cons.
BRATU1D	5003	0
BRATU3D	4913	3375
GOULDQP2	19999	10000
HS87	6	4
JNLBRNGA	10000	1
LEWISPOL	6	9
MANNE	6000	4000
MEYER3	3	0
NYSTROM5	18	20
OBSTCLBU	10000	1
RAYBENDS	2050	0
READING3	4002	2001
SEMICON2	5002	5000
SVANBERG	5000	5000
TORSION1	5476	0

3.7 Concluding remarks

The trust region SQP methods which have been proposed had two difficulties: one difficulty is the QP subproblem with the trust region constraint may not be feasible in considerable cases, and the other difficulty is the QP subproblem are not convex necessarily. In this chapter, we propose a new SQP method which resolves these two difficulties. We can expect that our method needs less calculated amount than before.

Future work concerning this chapter is to improve the method which we propose for fast convergence. It is well-known that the SQP method may fail to have fast convergence because of Maratos effect [28]. However, the SQP method with nonmonotone line search which has the superlinear convergence property has been proposed [55]. We want to apply this strategy to the method which we propose in this chapter.

Chapter 4

Inexact Levenberg-Marquardt Method under Local Error Bound Conditions

4.1 Introduction

In this chapter, we consider solving the system of nonlinear equations (SNE)

$$F(x) = 0, \tag{4.1.1}$$

where $F : \Re^n \to \Re^m$ is a continuously differentiable function. To solve (4.1.1) is one of the most fundamental themes in engineering, economics and so on.

When $m = n$, we can use Newton's method to solve (4.1.1). As a search direction at a current point x^k, Newton's method uses a solution d^k of the system of linear equations

$$\nabla F(x^k)d = -F(x^k), \tag{4.1.2}$$

where $\nabla F(x^k)$ is the Jacobian of F at x^k. Newton's method has a rapid convergence property, and hence it is used extensively in many applications. But, to obtain a solution efficiently by Newton's method, an initial point has to be sufficiently close to a solution x^* and the Jacobian of F at the solution x^* has to be nonsingular. Besides these drawbacks, (4.1.2) may have no solution.

On the other hand, even if $m \neq n$ or there is no solution of (4.1.2), the system of linear equations

$$\nabla F(x^k)^T \nabla F(x^k)d = -\nabla F(x^k)^T F(x^k) \tag{4.1.3}$$

always has a solution $\bar{d}^k$. The Gauss-Newton method uses the solution $\bar{d}^k$ as a search direction. However, like Newton's method, the Gauss-Newton method [31] requires an initial point to be sufficiently close to a solution in order to ensure convergence to a solution. Note that we can use Moore-Penrose's generalized inverse [32] to compute a solution of (4.1.3) when $\nabla F(x^k)^T \nabla F(x^k)$ is singular. However, methods based on Moore-Penrose's inverse are, in general, more expensive than Levenberg-Marquardt-based methods which will be considered in the following.

The Levenberg-Marquardt method (LMM) [1, 20, 28, 31, 60] is a modified Gauss-Newton method that is designed to overcome the above mentioned drawbacks of the Gauss-Newton method. The LMM uses a solution of the system of linear equations

$$\left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right) d = -\nabla F(x^k)^T F(x^k) \tag{4.1.4}$$

as a search direction $\hat{d}^k$, where μ_k is a positive parameter and I is the identity matrix. Since $\nabla F(x^k)^T \nabla F(x^k) + \mu_k I$ is always positive definite, (4.1.4) has a unique solution. Moreover, $\hat{d}^k$ is a descent direction of ϕ at x^k, where $\phi : \Re^n \to \Re$ is a merit function defined by

$$\phi(x) = \frac{1}{2}\|F(x)\|^2. \tag{4.1.5}$$

In fact, if $\nabla \phi(x^k) \neq 0$, we obtain

$$\nabla \phi(x^k)^T \hat{d}^k = -\left(\nabla F(x^k)^T F(x^k)\right)^T \left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right)^{-1} \left(\nabla F(x^k)^T F(x^k)\right) < 0.$$

Therefore, the LMM with Armijo's stepsize rule enjoys global convergence to a stationary point of ϕ. Especially, when $m = n$ and $\nabla F(x^*)$ is nonsingular at a stationary point x^*, x^* is a solution of (4.1.1) because

$$0 = \nabla \phi\left(x^*\right) = \nabla F\left(x^*\right)^T F\left(x^*\right).$$

Recently, Yamashita and Fukushima [60] showed that the LMM has a quadratic rate of convergence under the assumption that $\|F(x)\|$ provides a local error bound for (4.1.1), instead of the nonsingularity of $\nabla F(x)$ at a solution. Recall that $\|F(x)\|$ is said to provide a local error bound on a neighborhood N of a solution of (4.1.1) if there exists a positive constant c such that

$$c \operatorname{dist}(x, X^*) \leq \|F(x)\| \qquad \forall x \in N, \tag{4.1.6}$$

where X^* is the solution set of (4.1.1) and $\operatorname{dist}(x, X^*)$ denotes the distance from point x to the set X^*. Concrete examples of local error bounds can be found in [43]. When $m = n$ and $\nabla F(x^*)$ is nonsingular at a solution x^*, x^* is a locally unique solution of (4.1.1) and $\|F(x)\|$ provides a local error bound for (4.1.1) in a neighborhood of x^* [13]. Moreover, $\|F(x)\|$ may provide a local error bound even if $\nabla F(x)$ is singular at a solution. For example, let us consider the mapping $F : \Re^2 \to \Re^2$ defined by

$$F(x_1, x_2) = \left(e^{x_1 - x_2} - 1, (x_1 - x_2)(x_1 - x_2 - 2)\right)^T.$$

The solution set of (4.1.1) is given by $X^* = \left\{x \in \Re^2 \mid x_1 - x_2 = 0\right\}$, and hence $\operatorname{dist}(x, X^*) = \frac{\sqrt{2}}{2}|x_1 - x_2|$. Moreover, since

$$\nabla F(x_1, x_2) = \begin{pmatrix} e^{x_1 - x_2} & -e^{x_1 - x_2} \\ 2(x_1 - x_2 - 1) & -2(x_1 - x_2 - 1) \end{pmatrix},$$

$\nabla F(x_1, x_2)$ is singular everywhere. On the other hand, we obtain

$$\begin{aligned} \|F(x)\| &= \sqrt{(e^{x_1 - x_2} - 1)^2 + (x_1 - x_2)^2 (x_1 - x_2 - 2)^2} \\ &\geq |x_1 - x_2|\,|x_1 - x_2 - 2|. \end{aligned}$$

Consequently, it is easy to see that, when N is chosen as $N = \left\{ x \in \Re^2 \;\middle|\; |x_1 - x_2| \le 2 - \frac{\sqrt{2}}{2} \right\}$, condition (4.1.6) holds with $c = 1$. Therefore, the condition that $\|F(x)\|$ provides a local error bound in a neighborhood of x^* is milder than the condition that $\nabla F(x^*)$ is nonsingular.

In the LMM considered in [60], it is assumed that (4.1.4) is solved exactly at every iteration. For large-scale problems, however, it is expensive to solve (4.1.4) exactly, and hence it is often effective to use inexact methods that find an approximate solution satisfying some appropriate conditions. Therefore, in this chapter, we consider the inexact Levenberg-Marquardt method (ILMM) that uses an approximate solution d^k of (4.1.4). Let the vector r^k be defined by

$$r^k := \left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right) d^k + \nabla F(x^k)^T F(x^k). \tag{4.1.7}$$

The vector r^k is a residual vector associated with an approximate solution d^k. Facchinei and Kanzow [20] showed that the ILMM converges superlinearly under the assumption that $\|r^k\|$ is sufficiently small and $\nabla F(x^k)^T \nabla F(x^k)$ is uniformly nonsingular.

In this chapter, using techniques similar to [60], we show that the distance between the solution set and a sequence generated by the ILMM converges to 0 superlinearly under the assumption that $\|F(x)\|$ provides a local error bound in a neighborhood of a solution. Moreover, we propose the ILMM with Armijo's stepsize rule and show that the proposed algorithm enjoys global convergence.

This chapter is organized as follows: In Section 4.2, we establish local convergence of the ILMM with unit step size under the local error bound assumption. In Section 4.3, we propose the ILMM with Armijo's stepsize rule and show that the algorithm has global convergence. In Section 4.4, we report numerical results. In Section 4.5, we make some concluding remarks.

4.2 Local convergence

In this section, we discuss local convergence properties of the ILMM. Yamashita and Fukushima [60] consider a minimization problem equivalent to (4.1.4) and analyze local convergence of the LMM by using the properties of the minimization problem. In this section, we use similar techniques to analyze the rate of convergence of the ILMM.

For each k, we define $\theta^k : \Re^n \to \Re$ by

$$\theta^k(d) = \|\nabla F(x^k)d + F(x^k)\|^2 + \mu_k \|d\|^2, \tag{4.2.1}$$

and consider the minimization problem

$$\min_{d \in \Re^n} \; \theta^k(d). \tag{4.2.2}$$

Since the first order optimality condition for (4.2.2) is given by (4.1.4) and θ^k is a strictly convex quadratic function, (4.1.4) is in fact equivalent to (4.2.2).

First, we make the following assumption on F, under which we will show a superlinear rate of convergence of the ILMM.

Assumption 4.2.1

(i) There exists a solution x^* of (4.1.1).

(ii) There exist constants $b_1 \in (0,1)$ and $c_1 \in (0,\infty)$ such that

$$\|\nabla F(y)(x-y) - (F(x) - F(y))\| \le c_1 \|x-y\|^2 \quad \forall x, y \in B(x^*, b_1).$$

(iii) $\|F(x)\|$ provides an error bound for (4.1.1) on $B(x^*, b_1)$, i.e., there exists a constant $c_2 > 0$ such that

$$c_2 \operatorname{dist}(x, X^*) \le \|F(x)\| \quad \forall x \in B(x^*, b_1),$$

where X^* is the solution set of (4.1.1).

In this chapter, Assumption 4.2.1 (iii) plays the most important role instead of the nonsingularity of Jacobian. Note that Assumption 4.2.1 (ii) is satisfied if ∇F is Lipschitzian [41, Theorem 3.2.12]. Moreover, since F is continuously differentiable, $\|\nabla F(y)\|$ is bounded on $B(x^*, b_1)$ and F is Lipschitzian on $B(x^*, b_1)$, i.e., there exists a constant $L > 0$ such that

$$\|F(x) - F(y)\| \le L\|x-y\| \quad \forall x, y \in B(x^*, b_1). \tag{4.2.3}$$

In what follows, $\bar{x}^k$ denotes an arbitrary vector such that

$$\|\bar{x}^k - x^k\| = \operatorname{dist}(x^k, X^*) \text{ and } \bar{x}^k \in X^*.$$

Note that such $\bar{x}^k$ always exists even though the set X^* need not be convex.

The following assumption is concerned with the choice of the parameters μ_k used in the ILMM.

Assumption 4.2.2 *For each k, the parameter μ_k is chosen to satisfy*

$$\mu_k = \|F(x^k)\|^\delta, \tag{4.2.4}$$

where δ is a constant such that $0 < \delta \le 2$.

Throughout this section, we suppose that the ILMM generates a sequence $\{x^k\}$ by

$$x^{k+1} := x^k + d^k, \tag{4.2.5}$$

where d^k is an approximate solution of (4.1.4).

Now we show the next lemma.

Lemma 4.2.1 *Suppose that Assumptions 4.2.1 and 4.2.2 hold. If $x^k \in B(x^*, \frac{b_1}{2})$ and r^k is the residual vector given by (4.1.7), then*

$$\|d^k\| \le c_3 \operatorname{dist}(x^k, X^*) + \frac{\|r^k\|}{\mu_k}, \tag{4.2.6}$$

$$\|\nabla F(x^k)d^k + F(x^k)\| \le c_4 \operatorname{dist}(x^k, X^*)^{1+\frac{\delta}{2}} + \|\nabla F(x^k)\| \frac{\|r^k\|}{\mu_k}, \tag{4.2.7}$$

where $c_3 = \sqrt{\frac{c_1^2}{c_2^\delta}\left(\frac{b_1}{2}\right)^{2-\delta} + 1}$ and $c_4 = \sqrt{c_1^2\left(\frac{b_1}{2}\right)^{2-\delta} + L^\delta}$.

Proof: Let $\hat{d}^k$ be the exact solution of (4.1.4). Because (4.1.4) is equivalent to (4.2.2), we have

$$\theta^k(\hat{d}^k) \le \theta^k(\bar{x}^k - x^k). \tag{4.2.8}$$

Moreover, since $x^k \in B(x^*, \frac{b_1}{2})$, we have

$$\|\bar{x}^k - x^k\| \le \|x^* - x^k\| \le \frac{b_1}{2}, \tag{4.2.9}$$

and

$$\|\bar{x}^k - x^*\| \le \|\bar{x}^k - x^k\| + \|x^* - x^k\| \le \|x^* - x^k\| + \|x^* - x^k\| \le b_1,$$

and hence $\bar{x}^k \in B(x^*, b_1)$. It then follows from Assumptions 4.2.1 and 4.2.2 together with condition (4.2.3) that

$$\mu_k \quad = \quad \|F(x^k)\|^\delta \ge c_2^\delta \|\bar{x}^k - x^k\|^\delta \tag{4.2.10}$$

and

$$\mu_k \quad = \quad \|F(x^k)\|^\delta = \|F(\bar{x}^k) - F(x^k)\|^\delta \le L^\delta \, \|\bar{x}^k - x^k\|^\delta. \tag{4.2.11}$$

By the definition (4.2.1) of θ^k, we have

$$\begin{aligned}
\|\hat{d}^k\|^2 &\le \frac{1}{\mu_k}\theta^k(\hat{d}^k) \\
&\le \frac{1}{\mu_k}\theta^k(\bar{x}^k - x^k) \\
&= \frac{1}{\mu_k}\left(\|\nabla F(x^k)(\bar{x}^k - x^k) + F(x^k)\|^2 + \mu_k\|\bar{x}^k - x^k\|^2\right) \\
&\le \frac{1}{\mu_k}\left(c_1^2\|\bar{x}^k - x^k\|^4 + \mu_k\|\bar{x}^k - x^k\|^2\right) \\
&= \left(\frac{c_1^2\|\bar{x}^k - x^k\|^2}{\mu_k} + 1\right)\|\bar{x}^k - x^k\|^2 \\
&\le \left(\frac{c_1^2}{c_2^\delta}\|\bar{x}^k - x^k\|^{2-\delta} + 1\right)\|\bar{x}^k - x^k\|^2 \\
&\le \left\{\frac{c_1^2}{c_2^\delta}\left(\frac{b_1}{2}\right)^{2-\delta} + 1\right\}\|\bar{x}^k - x^k\|^2,
\end{aligned}$$

where the second inequality follows from (4.2.8), the third inequality follows from Assumption 4.2.1 (ii), the fourth inequality follows from (4.2.10), and the last inequality follows from (4.2.9). So we obtain

$$\|\hat{d}^k\| \le \sqrt{\frac{c_1^2}{c_2^\delta}\left(\frac{b_1}{2}\right)^{2-\delta} + 1}\; \|\bar{x}^k - x^k\|. \tag{4.2.12}$$

Moreover, from (4.1.7), we have

$$d^k = \hat{d}^k + \left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right)^{-1} r^k.$$

It then follows that

$$\begin{aligned}\left\|d^k\right\| &\leq \|\hat{d}^k\| + \left\|\left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right)^{-1}\right\| \|r^k\| \\ &\leq \|\hat{d}^k\| + \frac{\|r^k\|}{\mu_k}. \end{aligned} \tag{4.2.13}$$

Consequently, we obtain (4.2.6) with $c_3 = \sqrt{\frac{c_1^2}{c_2^\delta}\left(\frac{b_1}{2}\right)^{2-\delta} + 1}$ from (4.2.12) and (4.2.13).

Next we show (4.2.7). The left-hand side of (4.2.7) can be estimated as

$$\begin{aligned}\|\nabla F(x^k) d^k + F(x^k)\| &= \left\|\nabla F(x^k)\left(\hat{d}^k + \left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right)^{-1} r^k\right) + F(x^k)\right\| \\ &\leq \|\nabla F(x^k)\hat{d}^k + F(x^k)\| + \|\nabla F(x^k)\| \left\|\left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right)^{-1}\right\| \|r^k\| \\ &\leq \|\nabla F(x^k)\hat{d}^k + F(x^k)\| + \|\nabla F(x^k)\| \frac{\|r^k\|}{\mu_k}. \end{aligned} \tag{4.2.14}$$

It then follows from (4.2.1), (4.2.8), Assumption 4.2.1 (ii), (4.2.11), and (4.2.9) that

$$\begin{aligned}\|\nabla F(x^k)\hat{d}^k + F(x^k)\|^2 &\leq \theta^k(\hat{d}^k) \leq \theta^k(\bar{x}^k - x^k) \\ &\leq c_1^2 \|\bar{x}^k - x^k\|^4 + \mu_k \|\bar{x}^k - x^k\|^2 \\ &\leq c_1^2 \left(\frac{b_1}{2}\right)^{2-\delta} \|\bar{x}^k - x^k\|^{2+\delta} + L^\delta \|\bar{x}^k - x^k\|^{2+\delta} \\ &= \left\{c_1^2\left(\frac{b_1}{2}\right)^{2-\delta} + L^\delta\right\} \|\bar{x}^k - x^k\|^{2+\delta}. \end{aligned}$$

Therefore, the first term of (4.2.14) can be estimated as

$$\|\nabla F(x^k)\hat{d}^k + F(x^k)\| \leq \sqrt{c_1^2\left(\frac{b_1}{2}\right)^{2-\delta} + L^\delta}\, \|\bar{x}^k - x^k\|^{1+\frac{\delta}{2}},$$

which yields the desired inequality (4.2.7). □

Now we give a condition on $\|r^k\|$ for superlinear convergence of the ILMM.

Assumption 4.2.3 *The residual vector r^k given by (4.1.7) satisfies*

$$\frac{\|r^k\|}{\mu_k} = o\left(\mathrm{dist}(x^k, X^*)\right),$$

where $o(\cdot)$ means $\lim_{t\to 0} \frac{o(t)}{t} = 0$.

Suppose that Assumption 4.2.3 holds. Then there exists a constant $b_2 > 0$ such that

$$\mathrm{dist}(x^k, X^*) \leq b_2 \implies \frac{\|r^k\|}{\mu_k} \leq \mathrm{dist}(x^k, X^*). \tag{4.2.15}$$

By using Lemma 4.2.1, we show a key lemma of our analysis.

Lemma 4.2.2 *Suppose that Assumptions 4.2.1, 4.2.2 and 4.2.3 hold. Let b_2 be the constant given in (4.2.15) and $\hat{b}$ be the constant defined by $\hat{b} := \min\{b_1/(c_3+2), b_2\}$. If $x^k \in B(x^*, \hat{b})$, then*

$$\operatorname{dist}(x^{k+1}, X^*) = o\left(\operatorname{dist}(x^k, X^*)\right).$$

In particular, there exists a constant $b_3 > 0$ such that

$$\operatorname{dist}(x^k, X^*) \le b_3 \implies \operatorname{dist}(x^{k+1}, X^*) \le \frac{1}{2} \operatorname{dist}(x^k, X^*).$$

Proof: Since $\hat{b} \le \frac{b_1}{2}$ and $\hat{b} \le b_2$, we have from Lemma 4.2.1 and (4.2.15) that

$$\|d^k\| \le (c_3+1) \operatorname{dist}(x^k, X^*). \tag{4.2.16}$$

It follows from Lemma 4.2.1, Assumption 4.2.3 and the boundedness of $\|\nabla F(x)\|$ on $B(x^*, \hat{b})$ that

$$\begin{aligned} \left\|\nabla F(x^k)d^k + F(x^k)\right\| &\le c_4 \operatorname{dist}(x^k, X^*)^{1+\frac{\delta}{2}} + \|\nabla F(x^k)\| \cdot o\left(\operatorname{dist}(x^k, X^*)\right) \qquad (4.2.17)\\ &= o\left(\operatorname{dist}(x^k, X^*)\right). \end{aligned}$$

Moreover, from (4.2.16), we have

$$\begin{aligned} \|x^k + d^k - x^*\| &\le \|x^k - x^*\| + \|d^k\| \\ &\le \|x^k - x^*\| + (c_3+1)\operatorname{dist}(x^k, X^*) \\ &\le (c_3+2)\|x^k - x^*\| \le b_1, \end{aligned} \tag{4.2.18}$$

where the last inequality follows from $\|x^k - x^*\| \le \hat{b}$ and the definition of $\hat{b}$. Then we obtain

$$\begin{aligned} \operatorname{dist}(x^{k+1}, X^*) &\le \frac{1}{c_2}\|F(x^k + d^k)\| \\ &\le \frac{1}{c_2}\|\nabla F(x^k)d^k + F(x^k)\| + \frac{c_1}{c_2}\|d^k\|^2 \\ &\le o\left(\operatorname{dist}(x^k, X^*)\right) + \frac{c_1(c_3+1)^2}{c_2} \operatorname{dist}(x^k, X^*)^2 \\ &= o\left(\operatorname{dist}(x^k, X^*)\right), \end{aligned}$$

where the first inequality follows from (4.2.18), Assumption 4.2.1 (iii) and (4.2.5), the second inequality follows from Assumption 4.2.1 (ii), and the last inequality follows from (4.2.16) and (4.2.17). This completes the proof. □

Lemma 4.2.2 shows that $\{\operatorname{dist}(x^k, X^*)\}$ is convergent to 0 superlinearly if $x^k \in B(x^*, \hat{b})$ for all k. Now we give a sufficient condition for $x^k \in B(x^*, \hat{b})$ for all k.

Lemma 4.2.3 *Suppose that Assumptions 4.2.1, 4.2.2 and 4.2.3 hold. Let $\bar{b} := \min\{\hat{b}, b_3\}$ and $e := \frac{\bar{b}}{3+2c_3}$, where $\hat{b}, b_3$ and c_3 are given in Lemmas 4.2.1 and 4.2.2. If $x^0 \in B(x^*, e)$, then $x^k \in B(x^*, \bar{b}) \subseteq B(x^*, \frac{b_1}{2})$ for all k.*

Proof: Since $\|r^0\|/\mu_0 \le \operatorname{dist}(x^0, X^*)$ holds from (4.2.15), this lemma can be proven in a way similar to [60, Lemma 2.3]. The complete proof of this lemma is given in Appendix A. □

From these lemmas, we can show the next theorem, which is the main result in this chapter.

Theorem 4.2.1 *Suppose that Assumptions 4.2.1, 4.2.2 and 4.2.3 hold. Let* e *and* $\bar{b}$ *be the constants given in Lemma 4.2.3, and* $\{x^k\}$ *be a sequence generated by the ILMM with* $x^0 \in B(x^*, e)$ *and (4.2.5). Then,* $\{\text{dist}(x^k, X^*)\}$ *converges to* 0 *superlinearly. Moreover,* $\{x^k\}$ *converges to a solution* $\hat{x} \in B(x^*, \bar{b})$.

Proof: We can prove this theorem in a way similar to [60, Theorem 2.1]. The complete proof of this lemma is given in Appendix B. □

Note that Theorem 4.2.1 does not say that a sequence $\{x^k\}$ generated by the ILMM converges to $\hat{x}$ superlinearly. Moreover, this theorem does not show the quadratic convergence, though the exact LMM converges quadratically under the same conditions and $r^k = 0$ for all k [60]. However, we can show that the ILMM has a quadratic rate of convergence if we assume a more restrictive condition than Assumption 4.2.3.

Theorem 4.2.2 *Suppose that Assumptions 4.2.1 and 4.2.2 hold. Suppose also that* $\delta = 2$ *and*

$$\frac{\|r^k\|}{\mu_k} = O\left(\text{dist}(x^k, X^*)^2\right) \tag{4.2.19}$$

holds, where $O(\cdot)$ *means* $\lim_{t \to 0} \frac{O(t)}{t} < \infty$. *Let* e *and* $\bar{b}$ *be the constants defined in Lemma 4.2.3 and* $\{x^k\}$ *be a sequence generated by the ILMM with* $x^0 \in B(x^*, e)$ *and (4.2.5). Then,* $\{\text{dist}(x^k, X^*)\}$ *is convergent to* 0 *quadratically. Moreover,* $\{x^k\}$ *converges to a solution* $\hat{x} \in B(x^*, \bar{b})$.

Proof: Using proof techniques similar to Lemma 4.2.2, we immediately obtain from (4.2.19)

$$\text{dist}(x^{k+1}, X^*) = O\left(\text{dist}(x^k, X^*)^2\right).$$

Then, in a way similar to Theorem 4.2.1, we can show this theorem. □

Remark 4.2.1 *Condition (4.2.19) with* $\mu_k = O\left(\|F(x^k)\|^2\right)$ *is satisfied if* $\|r^k\| = O\left(\|F(x^k)\|^4\right)$, *which cannot be weakened because of Assumption 4.2.1 (iii). On the other hand, when* $n = m$ *and* $\nabla F(x^*)$ *is nonsingular, the ILMM with* $\mu_k = O\left(\|F(x^k)\|\right)$ *and* $\|r^k\| = O\left(\|F(x^k)\|^2\right)$ *has a quadratic rate of convergence [20]. This fact indicates that, to establish quadratic convergence under the weaker error bound condition, the regularization parameter* μ_k *and the residual* r^k *have to be much smaller than in the case of nonsingular Jacobian. This may be regarded as the cost to pay for relaxing the regularity assumption.*

4.3 Global convergence

In the previous section, we considered the rate of convergence of the ILMM with unit stepsize. In this section, we propose the ILMM with Armijo's stepsize rule and show that it has global convergence.

We propose the following algorithm, which uses the merit function ϕ defined by (4.1.5).

Algorithm ILMM

Step 0: Choose parameters $\alpha \in (0,1), \beta \in (0,1), \gamma \in (0,1), \rho \in (0,1), \delta \in (0,2], p > 0$, and an initial point $x^0 \in \Re^n$. Set $\mu_0 = \|F(x^0)\|^\delta$ and $k := 0$.

Step 1: If x^k satisfies a stopping criterion, then stop.

Step 2: Find an approximate solution d^k of the system of linear equations

$$\left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right) d = -\nabla F(x^k)^T F(x^k). \tag{4.3.1}$$

If the condition

$$\|F(x^k + d^k)\| \leq \gamma \|F(x^k)\| \tag{4.3.2}$$

is satisfied, then set $x^{k+1} := x^k + d^k$ and go to Step 4.

Step 3: If

$$\nabla \phi(x^k)^T d^k \leq -\rho \|d^k\|^p \tag{4.3.3}$$

is not satisfied, set $d^k = -\nabla \phi(x^k)$. Find the smallest nonnegative integer m such that

$$\phi(x^k + (\beta)^m d^k) - \phi(x^k) \leq \alpha(\beta)^m \nabla \phi(x^k)^T d^k,$$

and set $x^{k+1} := x^k + (\beta)^m d^k$.

Step 4: Set $\mu_{k+1} = \|F(x^{k+1})\|^\delta$ and $k := k + 1$. Go to Step 1.

In Step 3 of Algorithm ILMM, we must check whether a search direction d^k satisfies (4.3.3) or not. This is because d^k is not an exact solution of (4.3.1) in general and hence d^k may not be a good descent direction of the merit function ϕ. If d^k is not a good search direction, we reset d^k to be the steepest descent direction of ϕ. Consequently, d^k is always a sufficient descent direction of the merit function, and hence the line search in Step 3 is well-defined.

Now, we can prove the following global convergence theorem for Algorithm ILMM in a way similar to [20].

Theorem 4.3.1 *Let $\{x^k\}$ be a sequence generated by Algorithm ILMM. If the residual vector r^k given by (4.1.7) satisfies the condition*

$$\|r^k\| \leq \min\left\{\eta \|\nabla F(x^k)^T F(x^k)\|,\ \nu_k \|\nabla F(x^k)^T F(x^k)\|^\delta\right\}, \tag{4.3.4}$$

where $\eta \in (0,1)$ and $\nu_k = o\left(\mathrm{dist}(x^k, X^*)\right)$, then any accumulation point of $\{x^k\}$ is a stationary point of ϕ. Moreover, if an accumulation point x^* of $\{x^k\}$ is a solution of (4.1.1) that satisfies Assumption 4.2.1, then $\{\mathrm{dist}(x^k, X^*)\}$ converges to 0 superlinearly.

Proof: First we show that any accumulation point of $\{x^k\}$ is a stationary point of ϕ. Let $K_1 := \left\{k \;\middle|\; \|F(x^k+d^k)\| \le \gamma\|F(x^k)\|\right\}$. If K_1 is an infinite set, then we have $\|F(x^k)\| \to 0$ as $k \to \infty$ because $\left\{\|F(x^k)\|\right\}$ is a monotonically decreasing sequence. This shows that any accumulation point of $\{x^k\}$ is a stationary point of ϕ. If K_1 is finite, then, without loss of generality, we can assume that $\|F(x^k+d^k)\| > \gamma\|F(x^k)\|$ for all k. We will show that $\{d^k\}$ is gradient related to $\{x^k\}$, i.e., for any subsequence $\{x^k\}_{k\in K}$ converging to a nonstationary point of ϕ, $\{d^k\}_{k\in K}$ satisfies

$$\limsup_{k\to\infty,\, k\in K} \|d^k\| < \infty, \tag{4.3.5}$$

$$\liminf_{k\to\infty,\, k\in K} \nabla\phi(x^k)^T d^k < 0. \tag{4.3.6}$$

Then we can conclude that any accumulation point of $\{x^k\}$ is a stationary point of the merit function ϕ from Theorem 2.1.1. Let $\{x^k\}_{k\in K}$ be a subsequence converging to a nonstationary point of ϕ, and let $K_2 := \left\{k \in K \;\middle|\; d^k = -\nabla\phi(x^k)\right\}$. If K_2 is an infinite set, $\{d^k\}$ is obviously gradient related, and hence $\{x^k\}$ must converge to a stationary point of the merit function ϕ. If K_2 is a finite set, then we may assume, without loss of generality, that d^k are approximate solutions of (4.1.4) with residuals r^k satisfying (4.1.7) for all k.

First we show that (4.3.5) holds. To this end, we consider two cases for the sequence $\{\mu_k\}$: (i) $\liminf_{k\to\infty,k\in K}\mu_k = 0$ and (ii) $\liminf_{k\to\infty,k\in K}\mu_k > 0$.

Case (i): In this case, without loss of generality, we assume that $\mu_k \to 0$ $(k \in K)$. Then, we have $\|F(x^k)\| \to 0$ $(k \in K)$. Since $\left\{\|\nabla F(x^k)\|\right\}$ is bounded on any convergent subsequence $\{x^k\}_{k\in K}$, we obtain that $\nabla\phi(x^k) = \nabla F(x^k)^T F(x^k) \to 0$ $(k \in K)$. This contradicts the assumption that $\{x^k\}_{k\in K}$ converges to a nonstationary point. Therefore this case does not occur.

Case (ii): In this case, there exists a constant ξ such that $\mu_k \ge \xi > 0$ for all k. Since $\left\{\|\nabla F(x^k)^T F(x^k)\|\right\}$ is bounded on any convergent subsequence $\{x^k\}_{k\in K}$, it follows from (4.1.7) and (4.3.4) that

$$\begin{aligned}
\|d^k\| &\le \left\|\left(\nabla F(x^k)^T\nabla F(x^k) + \mu_k I\right)^{-1}\right\| \left(\|\nabla F(x^k)^T F(x^k)\| + \|r^k\|\right) \\
&\le \frac{1+\eta}{\xi}\|\nabla F(x^k)^T F(x^k)\| < \infty.
\end{aligned}$$

Therefore (4.3.5) holds.

Next, we show that (4.3.6) holds. Suppose to the contrary that (4.3.6) does not hold, i.e.,

$$\liminf_{k\to\infty,k\in K} \nabla\phi(x^k)^T d^k = 0.$$

It then follows from (4.3.3) that there exists an infinite set $K_3 \subset K$ such that

$$\lim_{k\to\infty,\, k\in K_3} d^k = 0.$$

On the other hand, we have from (4.1.7)

$$\begin{aligned}
\|\nabla\phi(x^k) - r^k\| &= \|\nabla F(x^k)^T F(x^k) - r^k\| \\
&\le \left\|\left(\nabla F(x^k)^T\nabla F(x^k) + \mu_k I\right) d^k\right\| \\
&\le \|\nabla F(x^k)^T\nabla F(x^k) + \mu_k I\|\|d^k\|.
\end{aligned}$$

Since $\limsup_{k\to\infty,k\in K_3} \|\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\| < \infty$ and $\lim_{k\to\infty,\, k\in K_3} d^k = 0$, we have

$$\lim_{k\to\infty,\, k\in K_3} \|\nabla\phi(x^k) - r^k\| = 0. \tag{4.3.7}$$

Moreover, from (4.3.4), we have $\|r^k\| \le \eta\|\nabla F(x^k)^T F(x^k)\| = \eta\|\nabla\phi(x^k)\|$. Then we can deduce from (4.3.7) that

$$\lim_{k\to\infty,\, k\in K_3} \|\nabla\phi(x^k)\| = \lim_{k\to\infty,\, k\in K_3} \|r^k\| = 0.$$

This contradicts the assumption that the limit point of $\{x^k\}_{k\in K}$ is not a stationary point. Therefore, we have (4.3.6), and hence $\{d^k\}$ is gradient related to $\{x^k\}$. Consequently we conclude that any accumulation point of $\{x^k\}$ is a stationary point of the merit function ϕ.

Now, we proceed to showing the latter half of the theorem. From Theorem 4.2.1, it is sufficient to show that Assumption 4.2.3 is satisfied and that $x^{k+1} = x^k + d^k$ with d^k being determined from (4.3.1) for all k large enough. Since $\|r^k\|$ satisfies (4.3.4), we have for any convergent subsequence $\{x^k\}_{k\in K}$

$$\begin{aligned}\frac{\|r^k\|}{\mu_k} &\le \frac{\nu_k\|\nabla F(x^k)^T F(x^k)\|^\delta}{\|F(x^k)\|^\delta}\\ &\le \nu_k\|\nabla F(x^k)\|^\delta = o\left(\operatorname{dist}(x^k, X^*)\right),\end{aligned}$$

where the last equality follows from the boundedness of $\left\{\|\nabla F(x^k)\|\right\}$ and the definition of $\{\nu_k\}$. This means that Assumption 4.2.3 holds. Let x^* be an accumulation point of $\{x^k\}$, and $\{x^k\}_{k\in K}$ be a subsequence such that $\lim_{k\in K} x^k = x^*$. Then there exists $\bar{k} \in K$ such that

$$\|x^k - x^*\| \le e,\ \forall k \ge \bar{k}, k \in K,$$

where e is given in Lemma 4.2.3. Moreover, by choosing larger $\bar{k}$ if necessary, we can show from Lemma 4.2.2 that a sequence $\{y^l\}$ generated by the ILMM with $y^0 = x^{\bar{k}}$ and the unit step size satisfies

$$\|\bar{y}^{l+1} - y^{l+1}\| \le \frac{c_2\gamma}{L}\|\bar{y}^l - y^l\| \quad \forall l, \tag{4.3.8}$$

where $\bar{y}^l$ denotes one of the nearest solutions from y^l, that is, $\bar{y}^l \in X^*$ and $\|\bar{y}^l - y^l\| = \operatorname{dist}(y^l, X^*)$. (Note that there may be more than one nearest solutions to y^l, since the solution set X^* need not be convex.) It then follows that

$$\begin{aligned}\|F(y^{l+1})\| &= \|F(\bar{y}^{l+1}) - F(y^{l+1})\|\\ &\le L\|\bar{y}^{l+1} - y^{l+1}\|\\ &\le c_2\gamma\|\bar{y}^l - y^l\|\\ &\le \gamma\|F(y^l)\|,\end{aligned}$$

where the first inequality follows from (4.2.3), the second inequality follows from (4.3.8), and the last inequality follows from Assumption 4.2.1 (iii). Hence, (4.3.2) is satisfied for $k \ge \bar{k}$, and we obtain $x^{k+1} = x^k + d^k$ for $k \ge \bar{k}$, where d^k is determined from (4.3.1). This completes the proof. □

Remark 4.3.1 *As an updating rule of ν_k that satisfies the assumption in Theorem 4.3.1, we may employ the rule*

$$\nu_k = \|F(x^k)\|^\tau,$$

where τ is a constant such that $\tau > 1$. In this case, it follows from (4.2.3) that

$$\begin{aligned}\nu_k &= \|F(x^k)\|^\tau = \|F(\bar{x}^k) - F(x^k)\|^\tau \\ &\leq L^\tau \|\bar{x}^k - x^k\|^\tau = o\left(\operatorname{dist}(x^k, X^*)\right)\end{aligned}$$

when k is sufficiently large.

In Theorem 4.3.1, we have established the superlinear convergence of Algorithm ILMM by using Theorem 4.2.1. Using Theorem 4.2.2, we can also give conditions for a quadratic convergence of Algorithm ILMM.

Theorem 4.3.2 *Let $\{x^k\}$ be a sequence generated by Algorithm ILMM with $\delta = 2$. If the residual vector r^k given by (4.1.7) satisfies*

$$\|r^k\| \leq \min\left\{\eta\|\nabla F(x^k)^T F(x^k)\|,\ \nu_k\|\nabla F(x^k)^T F(x^k)\|^\delta\right\}, \tag{4.3.9}$$

where $\eta \in (0,1)$ and $\nu_k = O\left(\operatorname{dist}(x^k, X^*)^2\right)$, then any accumulation point of $\{x^k\}$ is a stationary point of ϕ. Moreover, if an accumulation point x^* of $\{x^k\}$ is a solution of (4.1.1) that satisfies Assumption 4.2.1, then $\{\operatorname{dist}(x^k, X^*)\}$ converges to 0 quadratically.

Proof: It is easy to verify that the assumptions of Theorem 4.3.1 hold, so any accumulation point of $\{x^k\}$ is a stationary point of ϕ by Theorem 4.3.1. Moreover, since $\|r^k\|$ satisfies (4.3.9), we have

$$\frac{\|r^k\|}{\mu_k} \leq \frac{\nu_k\|\nabla F(x^k)^T F(x^k)\|^\delta}{\|F(x^k)\|^\delta} \leq \nu_k\|\nabla F(x^k)\|^\delta = O\left(\operatorname{dist}(x^k, X^*)^2\right),$$

and hence (4.2.19) is satisfied. Then, in a way similar to Theorem 4.3.1, we can show the quadratic convergence of $\{\operatorname{dist}(x^k, X^*)\}$. □

4.4 Numerical results

In this section, we discuss implementation issues of Algorithm ILMM proposed in Section 4.3 and report numerical results for a number of test problems where x^* is not a locally unique solution but $\|F(x)\|$ provides a local error bound in a neighborhood of x^*.

In implementing Algorithm ILMM, the most expensive task is to compute the search direction d^k by solving (4.3.1). Since the coefficient matrix $\nabla F(x^k)^T \nabla F(x^k) + \mu_k I$ of the linear equation (4.3.1) is always positive definite, we could find the exact solution of (4.3.1) by means of Cholesky factorization. However, our algorithm does not require the exact solution of (4.3.1), that is, the search direction d^k has only to satisfy the approximate condition (4.3.4). In our numerical experiments, we employ the conjugate gradient method (CGM) [1, 34] to find d^k satisfying the approximate condition (4.3.4) from the following two reasons. First, the CGM can find an approximate solution of (4.3.1) with any accuracy. So, when the approximate condition (4.3.4) is mild, the CGM may find d^k in a small number of iterations. Second, the CGM is suitable for large-scale problems. At each iteration of the CGM for (4.3.1), we need to calculate

$$-\left(\nabla F(x^k)^T \nabla F(x^k) + \mu_k I\right) \bar{d}^{j-1} - \nabla F(x^k)^T F(x^k), \tag{4.4.1}$$

where $\bar{d}^{j-1}$ is the search direction used in the previous iteration. This is done by calculating $\bar{v}^j := \nabla F(x^k)\bar{d}^{j-1}$ first, and then $\nabla F(x^k)^T\bar{v}^j + \mu_k \bar{d}^{j-1}$. Thus, the calculation of (4.4.1) is inexpensive if $\nabla F(x^k)$ is sparse.

Now, we state some practical modifications of Algorithm ILMM. If an iterative point x^k is far from the solution set, the value of parameter μ_k determined by the rule (4.2.4) may become exceedingly large. In this case, a search direction d^k obtained from (4.3.1) is close to the steepest descent direction for the function ϕ, and hence it is likely that the algorithm converges slowly. To prevent this difficulty, we modify the updating rule for μ_k as

$$\mu_k = \min\left\{\|F(x^k)\|^\delta, \zeta\right\}, \tag{4.4.2}$$

where $\zeta > 0$ is an appropriate constant. We can expect that, even if $\|F(x^k)\|$ is large, the rule (4.4.2) enables us to find a good approximation to the Gauss-Newton direction. Next we consider the criterion for approximate solution of the linear equation (4.3.1). If an iterative point is far from the solution set, then the approximate condition (4.3.4) for r^k need not be tight, and hence a computed search direction may also approximate the steepest descent direction, because we use the CGM to solve (4.3.1). In view of this fact, we modify the approximate condition (4.3.4) as

$$\|r^k\| \le \min\left\{\eta\|\nabla F(x^k)^T F(x^k)\|,\ \nu_k\|\nabla F(x^k)^T F(x^k)\|^\delta, \kappa\sqrt{n}\right\}, \tag{4.4.3}$$

where $\kappa > 0$ is an appropriate constant. This condition ensures that, even if an iterative point x^k is far from the solution set, $\|r^k\|$ is smaller than $\kappa\sqrt{n}$, and hence, especially in the early stage of the iterations, we can expect that a search direction becomes a good approximation to the Gauss-Newton direction.

We have solved the following problems with $n = 1000$.

$$\text{Problem 4.1}: \quad F:\Re^n \to \Re^n, \quad F_i(x) = \begin{cases} \sqrt{i}\exp((x_i + x_{i+1})/n) - \sqrt{i}, & \mathrm{mod}(i,2) = 1, \\ \sqrt{i}(x_{i-1} + x_i)(x_{i-1} + x_i - 1), & \mathrm{mod}(i,2) = 0. \end{cases}$$

$$\text{Problem 4.2}: \quad F:\Re^n \to \Re^n, \quad F_i(x) = \begin{cases} \sqrt{i}\exp((\sum_{j=i}^{i+3} x_j)/n) - \sqrt{i}, & \mathrm{mod}(i,4) = 1, \\ \sqrt{i}\sin((\sum_{j=i-1}^{i+2} x_j)/n), & \mathrm{mod}(i,4) = 2, \\ \sqrt{i}(\sum_{j=i-2}^{i+1} x_j)(\sum_{j=i-2}^{i+1} x_j - 1), & \mathrm{mod}(i,4) = 3, \\ \sqrt{i}(\sum_{j=i-3}^{i} x_j), & \mathrm{mod}(i,4) = 0. \end{cases}$$

$$\text{Problem 4.3}: \quad F:\Re^{2n} \to \Re^n, \quad F_i(x) = (x_i + x_{n+i})(x_i + x_{n+i} - \sqrt{i}), \quad i = 1,\dots,n.$$

$$\text{Problem 4.4}: \quad F:\Re^{2n} \to \Re^n, \quad F_i(x) = x_i x_{n+i} - \sqrt{i}, \quad i = 1,\dots,n.$$

Note that these problems have a continuum of solutions. Moreover, $\|F(x)\|$ provides a local error bound in a neighborhood of the solution set.

Table 4.1: Results for Problems 4.1–4.4

Problem	i.p.	# iter.	# iter. (CGM)	time(sec.)	$\|F(x^k)\|$
Problem 4.1	$x^{0,1}$	F			
	$x^{0,2}$	F			
	$x^{0,3}$	16	2128	12.9	2.1e-03, 4.1e-07, 4.7e-14
	$x^{0,4}$	17	2309	13.9	2.1e-03, 4.1e-07, 4.7e-14
Problem 4.2	$x^{0,1}$	17	1651	14.3	2.8e-01, 8.2e-04, 1.2e-08
	$x^{0,2}$	19	1808	15.8	9.4e-03, 2.1e-06, 2.3e-13
	$x^{0,3}$	16	1632	13.8	7.1e-03, 6.2e-07, 4.4e-15
	$x^{0,4}$	17	1645	14.3	1.2e-02, 2.9e-06, 2.6e-13
Problem 4.3	$x^{0,1}$	15	737	13.6	1.9e-02, 3.0e-04, 1.3e-07
	$x^{0,2}$	16	740	14.2	1.9e-02, 3.0e-04, 1.3e-07
	$x^{0,3}$	15	730	13.6	1.9e-02, 2.8e-04, 1.1e-07
	$x^{0,4}$	16	734	14.2	1.9e-02, 2.8e-04, 1.1e-07
Problem 4.4	$x^{0,1}$	14	244	10.4	1.3e-03, 1.0e-06, 7.8e-13
	$x^{0,2}$	15	247	11.0	1.3e-03, 1.0e-06, 7.8e-13
	$x^{0,3}$	14	242	10.4	1.2e-03, 7.8e-07, 4.5e-13
	$x^{0,4}$	15	245	11.0	1.2e-03, 7.8e-07, 4.5e-13

For each problem, we choose an initial point as $x^{0,1} = \{\frac{n}{2}, \dots, \frac{n}{2}\}^T$, $x^{0,2} = \{n, \dots, n\}^T$, $x^{0,3} = \{-\frac{n}{2}, \dots, -\frac{n}{2}\}^T$, and $x^{0,4} = \{-n, \dots, -n\}^T$. Moreover, we update ν_k in the approximate condition (4.3.4) as described in Remark 4.3.1. We set the default values of parameters as $\alpha = 0.6, \beta = 0.7, \gamma = 0.8, \delta = 1.0, \eta = 0.8, \rho = 0.5, \tau = 2.0, p = 2.0$ and $\zeta = 0.001$. As to parameter κ, we always set $\kappa = 0.001$ except Problem 4.2 with initial points $x^{0,2}$ and $x^{0,4}$, for which we set $\kappa = 0.01$. We use the stopping criterion $\|F(x^k)\| < 10^{-8}\sqrt{n}$ for each experiment. The algorithm was coded in C and run on a Sun Ultra 60 workstation.

Table 4.1 shows the computational results for each problem, with the following items: The initial point (i.p.), the number of iterations of Algorithm ILMM (# iter.), the cumulative number of iterations of the CGM (# iter. (CGM)), the CPU time in second (time(sec.)), and the values of $\|F(x^k)\|$ at last three iterations of Algorithm ILMM ($\|F(x^k)\|$). The symbol "F" in the column "# iter." means that Algorithm ILMM fails to solve the problem. Note that, from Assumption 4.2.1 (iii), we have $\|F(x^k)\| = O\left(\mathrm{dist}(x^k, X^*)\right)$. For each problem except Problem 4.1 with initial points $x^{0,1}$ and $x^{0,2}$, Algorithm ILMM always stopped successfully and the generated sequence converged to the solution set superlinearly. The reason why Algorithm ILMM fails to solve Problem 4.1 with initial points $x^{0,1}$ and $x^{0,2}$ is that the generated sequences converge to stationary points of the merit function that do not solve Problem 4.1.

4.5 Concluding remarks

In this chapter, we have discussed the convergence properties of the ILMM under a local error bound condition on F. Using an approach similar to [60], we have showed that the ILMM converges to the solution set superlinearly under appropriate conditions on the approximate solution of the system of linear equations solved at each iteration. For large scale problems, this property is very useful. On the other hand, it was shown in [60] that the LMM converges quadratically under the assumption that $\mu_k = \|F(x^k)\|^2$. In that case, if μ_k is very small and $\nabla F(x^*)^T \nabla F(x^*)$ is singular, the system of linear equations tends to be unstable numerically. Since the ILMM converges superlinearly even if $\mu_k = \|F(x^k)\|^\delta, 0 < \delta \leq 2$, we can expect the numerical robustness of the ILMM when it is implemented with $0 < \delta < 2$.

Chapter 5

Identification of Degenerate Indices in Monotone NCP

5.1 Introduction

The nonlinear complementarity problem [22] NCP(F) is to find a vector $x \in \Re^n$ such that

$$x_i \geq 0,\ F_i(x) \geq 0,\ x_i F_i(x) = 0 \ \text{ for all } i \in J,$$

where F is a mapping from $\Re^n$ to $\Re^n$ and $J := \{1, 2, \ldots, n\}$. When F is affine, the problem is called the linear complementarity problem (LCP) [11]. Throughout this chapter, we assume that F is monotone and continuously differentiable, and that NCP(F) has a nonempty solution set, which we denote X^*.

The purpose of this chapter is to propose a method of identifying three index sets $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$ for a solution $\bar{x}$ of NCP(F), which are defined by

$$\begin{aligned} P(\bar{x}) &:= \{i \in J \mid \bar{x}_i > 0, F_i(\bar{x}) = 0\}, \\ N(\bar{x}) &:= \{i \in J \mid \bar{x}_i = 0, F_i(\bar{x}) > 0\}, \\ C(\bar{x}) &:= \{i \in J \mid \bar{x}_i = 0, F_i(\bar{x}) = 0\}, \end{aligned}$$

respectively. (The symbols P, N and C stand for "positive", "naught" and "complementarity", respectively. The first two are related to the status of $\bar{x}_i$.) When the point $\bar{x}$ under consideration is clear from the context, we shall denote these sets simply by P, N and C. By definition it is easy to verify that $J = P \cup N \cup C$, $P \cap N = \emptyset$, $N \cap C = \emptyset$, and $P \cap C = \emptyset$. Therefore, these sets characterize the solution $\bar{x}$. In particular, we call $\bar{x}$ a degenerate solution if $C \neq \emptyset$, otherwise we call it a nondegenerate solution. Moreover, it is important to identify these sets from a practical point of view. If we can identify the index sets P, N and C before we know $\bar{x}$ exactly, then the original NCP(F) can be reduced to the following system:

$$F_P(x) = 0,\ x_P \geq 0,\ F_N(x) \geq 0,\ x_N = 0,\ F_C(x) = x_C = 0,$$

where, for a vector $z \in \Re^n$ and an index set K, z_K denotes the $|K|$-dimensional vector with components $z_i, i \in K$. This system is more tractable than the original NCP(F).

Several methods have been proposed for identifying the index sets P, N and C at a solution $\bar{x}$ of NCP(F). The case where F is continuous, C is empty and there exists a sequence $\{x^k\}$ converging to $\bar{x}$ is not difficult. In such a case, let index sets P^k and N^k be defined by $P^k := \{i \mid x_i^k > F_i(x^k)\}$ and $N^k := \{i \mid x_i^k < F_i(x^k)\}$, respectively. Then, because of the continuity of F, we have for all k sufficiently large, $P(\bar{x}) = P^k$ and $N(\bar{x}) = N^k$. However, this approach cannot be applied when $\bar{x}$ is degenerate. El-Bakry, Tapia and Zhang [16, 17], Stoer, Wechs and Mizuno [50] and Facchinei, Fischer and Kanzow [19] proposed identification techniques based on the interior point method. The methods do not require the local uniqueness of the solution $\bar{x}$. However the methods only deal with LCP, and they cannot be directly applied to NCP(F). Recently, Facchinei, Fischer and Kanzow [18] presented a technique based on the error bound results. The method is originally developed for the identification of the active set of an inequality constrained minimization problem, but it is also applicable to NCP(F). However, the technique proposed in [18] requires the local uniqueness of the solution $\bar{x}$. Until now, to the authors' knowledge, no technique has been proposed which can successfully identify the index sets for NCP(F) without assuming the local uniqueness of the solution $\bar{x}$.

In this chapter we propose a new identification technique based on Algorithm PPA introduced in Section 2.4. The proposed technique generates a sequence of index sets $\{P^k, N^k, C^k\}$ using the information on $\{x^k\}$, and eventually identifies $P(x^*), N(x^*)$ and $C(x^*)$, where x^* is a limit point of the sequence generated by Algorithm PPA. Moreover, a solution of the equality constrained minimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x - x^k\|^2 \\ \text{subject to} & F_{P^k}(x) = 0,\ x_{N^k} = 0, \\ & F_{C^k}(x) = 0,\ x_{C^k} = 0 \end{array}$$

eventually becomes a solution of NCP(F). When F is affine, we can compute a solution of the above problem by solving a system of linear equations derived from its KKT conditions. Therefore, for the monotone LCP, we can construct the PPA with the finite termination property.

The chapter is organized as follows. In Section 5.2 we present a new identification technique based on Algorithm PPA, and show that, under mild assumptions, it eventually identifies the sets P, N and C at x^*. In Section 5.3 we report some numerical results which show that the proposed technique can identify the index sets successfully. In Section 5.4, by using this technique, we construct a minimization problem with equality constraints whose solution is a solution of the original NCP(F).

5.2 Identification of degenerate indices

The purpose of this section is to present a new technique for identifying the index sets

$$\begin{aligned} P(x^*) &:= \{i \mid x_i^* > 0, F_i(x^*) = 0\}, \\ N(x^*) &:= \{i \mid x_i^* = 0, F_i(x^*) > 0\}, \\ C(x^*) &:= \{i \mid x_i^* = 0, F_i(x^*) = 0\}, \end{aligned}$$

where x^* is the limit point of a sequence generated by Algorithm PPA.

To begin with, we explain the basic idea of our approach. Suppose that there exists a sequence $\{x^k\}$ converging to a solution x^* of NCP(F). Suppose also that there exists a positive sequence $\{a_k\}$ such that

$$\max\{\|x^k - x^*\|, \|F(x^k) - F(x^*)\|\} \le a_k, \quad \lim_{k\to\infty} a_k = 0. \tag{5.2.1}$$

Using $\{x^k\}$ and $\{a_k\}$, let index sets P^k, N^k and C^k be defined by

$$\begin{aligned} P^k &:= \{i \mid x_i^k > a_k, F_i(x^k) \le a_k\}, \\ N^k &:= \{i \mid x_i^k \le a_k, F_i(x^k) > a_k\}, \\ C^k &:= \{i \mid x_i^k \le a_k, F_i(x^k) \le a_k\}, \end{aligned}$$

respectively. Then, it is not difficult to show that, for all k sufficiently large

$$P^k = P^* := P(x^*), \quad N^k = N^* := N(x^*), \quad C^k = C^* := C(x^*).$$

For example, we can show that $C^k = C^*$ as follows. For each $i \in C^*$, we have

$$x_i^k \le |x_i^k - x_i^*| \le \|x^k - x^*\| \le a_k,$$

and similarly

$$F_i(x^k) \le a_k.$$

Therefore, we have $C^* \subseteq C^k$. Moreover, noting that $a_k \to 0$ and $x^k \to x^*$, we see that $C^k \subseteq C^*$ for large k. Hence C^k must coincide with C^* for all k sufficiently large. In this manner, the idea underlying the approach is easy to understand, but it is not trivial to find an adequate sequence $\{a_k\}$ which satisfies (5.2.1).

Facchinei, Fischer and Kanzow [18] proposed to use $\|H_F(x^k)\|^{\frac{1}{2}}$ as $\{a_k\}$. In [18], a function $f : \Re^n \to \Re$ satisfying

$$\max\{\|x^k - x^*\|, \|F(x^k) - F(x^*)\|\} \le f(x^k), \quad \lim_{k\to\infty} f(x^k) = 0$$

is called *an identification function.* If x^* is a locally unique solution of NCP(F) and $\|H_F(x)\|$ provides a local error bound, then it is easy to verify that $\|H_F(\cdot)\|^{\frac{1}{2}}$ can serve as an identification function. Here, the local uniqueness of x^* seems crucial, because this local error bound only measures the distance between a point x and the nearest solution of NCP(F) to x, rather than a specific solution. To deal with the case where the local uniqueness assumption fails to hold, we focus on a specific sequence that is generated by Algorithm PPA. (Note that in [18] a sequence $\{x^k\}$ need not be generated by a particular algorithm.) The fact that $\{x^k\}$ converges superlinearly to the solution set of NCP(F) will play an essential role in our approach.

Before we present a new technique, we introduce the concept of *an identification function sequence.*

Definition 5.2.1 *Suppose that a sequence* $\{x^k\}$ *converges to a solution* x^* *of NCP*(F)*. Suppose also that* P^k, N^k *and* C^k *are defined by*

$$\begin{aligned} P^k &:= \{i \mid x_i^k > f^k(x^k), F_i(x^k) \le f^k(x^k)\}, \\ N^k &:= \{i \mid x_i^k \le f^k(x^k), F_i(x^k) > f^k(x^k)\}, \\ C^k &:= \{i \mid x_i^k \le f^k(x^k), F_i(x^k) \le f^k(x^k)\}, \end{aligned} \tag{5.2.2}$$

where $\{f^k\}$ *is a sequence of functions from* $\Re^n$ *to* $\Re$*. Then* $\{f^k\}$ *is called an identification function sequence for* $\{x^k\}$*, if*

$$P^k = P(x^*), \quad N^k = N(x^*), \quad C^k = C(x^*)$$

hold for all k *sufficiently large.*

Note that if f is an identification function, then the sequence $\{f^k\}$ with $f^k \equiv f$ is an identification function sequence. Moreover, if a sequence $\{a_k\}$ satisfies (5.2.1), then any sequence $\{f^k\}$ such that $f^k(x^k) = a_k$ can be used as an identification function sequence.

The main objective of this chapter is to show that the function sequence $\{\rho^k\}$ given by

$$\rho^k(x) := \eta \left(\frac{\|H_F(x)\|}{c_k} \right)^{\beta} \tag{5.2.3}$$

with constants $\eta > 0$ and $\beta \in (0,1)$ can serve as an identification function sequence for the sequence $\{x^k\}$ generated by Algorithm PPA with parameters $\{c_k\}$. To this end, we need the following assumptions.

Assumption 5.2.1

(i) The solution set X^* *of NCP*(F) *is nonempty.*

(ii) $\|H_F(x)\|$ *provides a local error bound for NCP*(F)*, cf.* (2.3.1).

The following are sufficient conditions for Assumption 5.2.1 (ii).

(a) The mapping F is affine, that is, NCP(F) is LCP.

(b) The mapping F is strongly monotone.

(c) The mapping F is given by

$$F(x) := \begin{pmatrix} G(y) \\ Uy + Vz \end{pmatrix},$$

where $x = (y, z)$, G is a strongly monotone mapping, and U, V are matrices of appropriate dimensions.

(d) The mapping F is derived from the Karush-Kuhn-Tucker conditions of the nonlinear program

$$\begin{array}{ll} \text{minimize} & f(y) \\ \text{subject to} & g(y) \le 0, \quad y \ge 0, \end{array} \tag{5.2.4}$$

and the following statements are satisfied at each solution of (5.2.4).

- The functions f and g are twice continuously differentiable;
- The Mangasarian-Fromovitz constraint qualifications hold;
- The second-order sufficient conditions hold.

Note that F is given by

$$F(x) := \begin{pmatrix} \nabla f(y) + \nabla g(y)\lambda \\ -g(y) \end{pmatrix}$$

with $x := (y, \lambda)$.

Since the proof of the sufficiency of conditions (c) and (d) is somewhat lengthy, it is shown in Appendix C.

We remark that the following statements hold under Assumption 5.2.1.

Remark 5.2.1

(i) By Theorem 2.4.1 and Assumption 5.2.1 (ii), $\{\mathrm{dist}\{x^k, X^*\}\}$ *converges superlinearly to* 0, *that is,*

$$\lim_{k\to\infty} \frac{\mathrm{dist}\{x^{k+1}, X^*\}}{\mathrm{dist}\{x^k, X^*\}} = 0. \tag{5.2.5}$$

(ii) Since $\{x^k\}$ *is bounded,* $\{\tilde{x}^{k+1}\}$ *is also bounded by condition* (2.4.2) *in Step 2 of Algorithm PPA. Furthermore, since* F *is continuously differentiable,* F^k *is Lipschitz continuous with a constant independent of* k *on a bounded set* $S \subseteq \Re^n$ *containing* $\{x^k\}$, $\{P_k(x^k)\}$ *and* $\{\tilde{x}^k\}$, *where* $P_k(x^k)$ *is the unique solution of NCP(*F^k*). It then follows from the strong monotonicity of* F^k *with modulus* $c_k = (\alpha)^k$ *and Theorem 2.3.1 (i) that* $\|H_{F^k}(x)\|/(\alpha)^k$ *provides an error bound for NCP(*F^k*) on* S, *that is, there exists a positive constant* b_3 *independent of* k *such that*

$$\|x - P_k(x^k)\| \leq b_3 \frac{\|H_{F^k}(x)\|}{(\alpha)^k} \qquad \forall x \in S. \tag{5.2.6}$$

Now we state the main result.

Theorem 5.2.1 *Let Assumption 5.2.1 be satisfied. Then the function sequence* $\{\rho^k\}$ *defined by* (5.2.3) *is an identification function sequence for* $\{x^k\}$ *generated by Algorithm PPA.*

The proof of this theorem will be given after showing some lemmas.

Lemma 5.2.1 *Let Assumption 5.2.1 be satisfied. Then there exist a positive integer* k_1 *and a positive constant* B_1 *such that*

$$\|x^{k+1} - x^k\| \leq B_1 \frac{\mathrm{dist}\{x^k, X^*\}}{(\alpha)^k} \qquad \forall k \geq k_1. \tag{5.2.7}$$

Proof: Since $x^{k+1} = [\tilde{x}^{k+1}]_+$ and $x^k = [x^k]_+$, we have for each k

$$\|x^{k+1} - x^k\| \le \|\tilde{x}^{k+1} - x^k\|. \tag{5.2.8}$$

Moreover, we have

$$\|\tilde{x}^{k+1} - x^k\| \le \|\tilde{x}^{k+1} - P_k(x^k)\| + \|P_k(x^k) - x^k\|, \tag{5.2.9}$$

where $P_k(x^k)$ is the unique solution of NCP(F^k). Since $\|H_{F^k}(x)\|/(\alpha)^k$ provides an error bound for NCP(F^k) on a bounded set $S \subseteq \Re^n$ containing $\{x^k\}$, $\{\tilde{x}^k\}$ and $\{P_k(x^k)\}$ as mentioned in Remark 5.2.1 (ii), we have

$$\|\tilde{x}^{k+1} - P_k(x^k)\| \le b_3 \frac{\|H_{F^k}(\tilde{x}^{k+1})\|}{(\alpha)^k},$$

where b_3 is a positive constant independent of k. It then follows from Step 2 in Algorithm PPA that

$$\|\tilde{x}^{k+1} - P_k(x^k)\| \le b_3 \frac{((\alpha)^k)^4}{(\alpha)^k} \|\tilde{x}^{k+1} - x^k\|. \tag{5.2.10}$$

By the error bound property of $\|H_{F^k}(x)\|/(\alpha)^k$, we also have

$$\|P_k(x^k) - x^k\| \le b_3 \frac{\|H_{F^k}(x^k)\|}{(\alpha)^k},$$

which together with (5.2.9) and (5.2.10) implies

$$\|\tilde{x}^{k+1} - x^k\| \le b_3(\alpha)^{3k} \|\tilde{x}^{k+1} - x^k\| + b_3 \frac{\|H_{F^k}(x^k)\|}{(\alpha)^k}.$$

Since $\alpha \in (0,1)$, there exists an integer $k_1 > 0$ such that $b_3(\alpha)^{3k} < \frac{1}{2}$ for all $k \ge k_1$. It then follows from $\|H_{F^k}(x^k)\| = \|H_F(x^k)\|$ that the above inequality can be rewritten as

$$\|\tilde{x}^{k+1} - x^k\| \le \frac{b_3}{1 - b_3(\alpha)^{3k}} \frac{\|H_F(x^k)\|}{(\alpha)^k} \le 2b_3 \frac{\|H_F(x^k)\|}{(\alpha)^k} \qquad \forall k \ge k_1. \tag{5.2.11}$$

Since F is continuously differentiable, F is Lipschitz continuous on the bounded set S. Therefore, H_F is also Lipschitz continuous, that is, there exists a constant $L > 0$ such that, for all k,

$$\|H_F(x^k)\| = \|H_F(x^k) - H_F(\bar{x}^k)\| \le L\|x^k - \bar{x}^k\| = L\text{dist}\{x^k, X^*\}, \tag{5.2.12}$$

where $\bar{x}^k$ is the nearest solution of NCP(F) to x^k. It follows from (5.2.8) together with (5.2.11) and (5.2.12) that

$$\|x^{k+1} - x^k\| \le 2Lb_3 \frac{\text{dist}\{x^k, X^*\}}{(\alpha)^k} \qquad \forall k \ge k_1.$$

Therefore, setting $B_1 := 2Lb_3$ yields the desired inequality (5.2.7). □

From Lemma 5.2.1, we can deduce the key property for our main theorem.

Lemma 5.2.2 *Let Assumption 5.2.1 be satisfied. Then there exist a positive integer* k_2 *and a positive constant* B_2 *such that*

$$\|x^k - x^*\| \leq B_2 \frac{\|H_F(x^k)\|}{(\alpha)^k} \qquad \forall k \geq k_2. \tag{5.2.13}$$

Proof: Since $x^k \to x^*$, we have from the triangle inequality

$$\|x^k - x^*\| \leq \sum_{i=0}^{\infty} \|x^{k+i+1} - x^{k+i}\|.$$

It then follows from Lemma 5.2.1 that for $k \geq k_1$

$$\|x^k - x^*\| \leq B_1 \left\{ \frac{\text{dist}\{x^k, X^*\}}{(\alpha)^k} + \frac{\text{dist}\{x^{k+1}, X^*\}}{(\alpha)^{k+1}} + \cdots \right\}.$$

Since $\{\text{dist}\{x^k, X^*\}\}$ converges superlinearly to 0 as mentioned in Remark 5.2.1 (i), there exists an integer $k_3 > 0$ such that

$$\text{dist}\{x^{k+1}, X^*\} \leq \frac{\alpha}{2} \text{dist}\{x^k, X^*\} \qquad k \geq k_3.$$

Hence, we have for $k \geq \max\{k_1, k_3\}$

$$\begin{aligned} \|x^k - x^*\| &\leq B_1 \left\{ \frac{\text{dist}\{x^k, X^*\}}{(\alpha)^k} + \frac{1}{2} \frac{\text{dist}\{x^k, X^*\}}{(\alpha)^k} + \frac{1}{2^2} \frac{\text{dist}\{x^k, X^*\}}{(\alpha)^k} + \cdots \right\} \\ &\leq 2B_1 \frac{\text{dist}\{x^k, X^*\}}{(\alpha)^k}. \end{aligned} \tag{5.2.14}$$

Since $\|H_F(x)\|$ provides a local error bound by Assumption 5.2.1 (ii) and $x^k \to x^* \in X^*$, there exists a positive integer k_4 such that

$$\text{dist}\{x^k, X^*\} \leq b_2 \|H_F(x^k)\| \qquad \forall k \geq k_4.$$

Let $k_2 := \max\{k_1, k_3, k_4\}$. Then, we have from (5.2.14)

$$\|x^k - x^*\| \leq 2B_1 b_2 \frac{\|H_F(x^k)\|}{(\alpha)^k} \qquad \forall k \geq k_2,$$

and hence setting $B_2 := 2B_1 b_2$ yields (5.2.13). □

Lemma 5.2.3 *Let Assumption 5.2.1 be satisfied. Then the sequence* $\{\|H_F(x^k)\|/(\alpha)^k\}$ *converges to* 0. *Moreover, there exists a positive integer* k_5 *such that*

$$\|x^k - x^*\| \leq \rho^k(x^k), \;\; \|F(x^k) - F(x^*)\| \leq \rho^k(x^k) \qquad \forall k \geq k_5.$$

Proof: By the Lipschitz continuity of H_F on S, where S is given in the proof of Lemma 5.2.1, we have $\|H_F(x^{k+1})\| \leq \hat{L}\text{dist}\{x^{k+1}, X^*\}$ for some $\hat{L} > 0$. Since $x^k \to x^* \in X^*$, Assumption 5.2.1 (ii) implies $\text{dist}\{x^k, X^*\} \leq b_2 \|H_F(x^k)\|$ for all k sufficiently large. Let $\gamma < \alpha/(\hat{L}b_2)$. Then by (5.2.5) there exists an integer $k_6 > 0$ such that for all $k > k_6$

$$\frac{\|H_F(x^{k+1})\|}{(\alpha)^{k+1}} \leq \frac{\hat{L}\text{dist}\{x^{k+1}, X^*\}}{(\alpha)^{k+1}} \leq \frac{\gamma \hat{L}\text{dist}\{x^k, X^*\}}{(\alpha)^{k+1}} \leq \frac{\gamma \hat{L} b_2}{\alpha} \frac{\|H_F(x^k)\|}{(\alpha)^k}.$$

This shows that $\{\|H_F(x^k)\|/(\alpha)^k\}$ converges to 0.

Next we show the last part of the lemma. From Lemma 5.2.2 and the Lipschitz continuity of F on S, we have for $k \geq k_2$

$$\begin{aligned}\max\{\|x^k - x^*\|, \|F(x^k) - F(x^*)\|\} &\leq \max\{1, L\}\|x^k - x^*\| \\ &\leq \max\{1, L\} B_2 \frac{\|H_F(x^k)\|}{(\alpha)^k} \\ &= \frac{\max\{1, L\} B_2}{\eta} \left(\frac{\|H_F(x^k)\|}{(\alpha)^k}\right)^{1-\beta} \rho^k(x^k).\end{aligned}$$

From the first part of the lemma, there exists an integer $k_7 > 0$ such that

$$\frac{\max\{1, L\} B_2}{\eta} \left(\frac{\|H_F(x^k)\|}{(\alpha)^k}\right)^{1-\beta} \leq 1$$

for all $k \geq k_7$. Hence setting $k_5 := \max\{k_2, k_7\}$ yields the desired inequality. □

Now we stand at the position to prove our main theorem.

Proof of Theorem 5.2.1. It suffices to show that there exists a positive integer $\bar{k}$ such that for all $k \geq \bar{k}$

$$x_i^* > 0 \iff x_i^k > \rho^k(x^k), \tag{5.2.15}$$

$$x_i^* = 0 \iff x_i^k \leq \rho^k(x^k), \tag{5.2.16}$$

$$F_i(x^*) > 0 \iff F_i(x^k) > \rho^k(x^k), \tag{5.2.17}$$

$$F_i(x^*) = 0 \iff F_i(x^k) \leq \rho^k(x^k). \tag{5.2.18}$$

Notice that $\{\|H_F(x^k)\|/(\alpha)^k\}$ converges to 0 by Lemma 5.2.3, and hence $\{\rho^k(x^k)\}$ converges to 0. It then follows from $x^k \to x^*$ and the continuity of F that there exists a positive integer k_8 such that

$$\begin{aligned} x_i^* > 0 &\Rightarrow x_i^k > \rho^k(x^k), \\ x_i^* = 0 &\Leftarrow x_i^k \leq \rho^k(x^k), \\ F_i(x^*) > 0 &\Rightarrow F_i(x^k) > \rho^k(x^k), \\ F_i(x^*) = 0 &\Leftarrow F_i(x^k) \leq \rho^k(x^k), \end{aligned}$$

for $k \geq k_8$. Next, we show that the converse of each relation also holds for $k \geq k_5$, where k_5 is the positive integer given in Lemma 5.2.3. First we show $x_i^* = 0 \Rightarrow x_i^k \leq \rho^k(x^k)$ for $k \geq k_5$. Suppose that $x_i^* = 0$. Then we have from Lemma 5.2.3

$$x_i^k = x_i^k - x_i^* \leq \|x^k - x^*\| \leq \rho^k(x^k) \qquad k \geq k_5.$$

In a similar way, we can show $F_i(x^*) = 0 \Rightarrow F_i(x^k) \leq \rho^k(x^k)$. Next we show $x_i^k > \rho^k(x^k) \Rightarrow x_i^* > 0$ for all $k \geq k_5$. Suppose the contrary, that is, there exists $k \geq k_5$ such that $x_i^k > \rho^k(x^k)$ and $x_i^* = 0$. This immediately leads to a contradiction, because $x_i^* = 0$ implies $x_i^k \leq \rho^k(x^k)$ for $k \geq k_5$ as shown above. In a similar way, we can show $F_i(x^k) > \rho^k(x^k) \Rightarrow F_i(x^*) > 0$. Consequently we have (5.2.15)–(5.2.18) for all $k \geq \bar{k} := \max\{k_5, k_8\}$. □

5.3 Numerical results

In this section, we present some numerical results for the identification technique proposed in the previous section. We employed the Fischer-Burmeister function ϕ_{FB} to define the identification function sequence $\{\rho^k\}$ and solved the following test problems:

- **Problems 5.1–5.8** (LCPs derived from quadratic programming problems): The Karush-Kuhn-Tucker conditions of the quadratic programming problem (QP)

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}y^TQy + c^Ty, \\ &\text{subject to} \quad Ay \le b,\ y \ge 0 \end{aligned} \tag{5.3.1}$$

can be written as the linear complementarity problem

$$F(x) := \begin{pmatrix} Q & A^T \\ -A & 0 \end{pmatrix} \begin{pmatrix} y \\ \lambda \end{pmatrix} + \begin{pmatrix} c \\ b \end{pmatrix} \ge 0, \quad x \ge 0, \quad F(x)^Tx \ge 0, \tag{5.3.2}$$

where λ is a Lagrange multiplier vector associated with the constraint $Ay \le b$ and $x := (y, \lambda)^T$. It is well-known that the QP (5.3.1) is equivalent to the LCP (5.3.2) when Q is a positive semidefinite matrix. The specific matrices Q, A and vectors b, c involved in Problems 5.1–5.8 are as follows:

Problem 5.1 (LCP derived from [35, Problem 35]):

$$Q := \begin{pmatrix} 4 & 2 & 2 \\ 2 & 4 & 0 \\ 2 & 0 & 2 \end{pmatrix}, \ c := \begin{pmatrix} -8 \\ -6 \\ -4 \end{pmatrix}, \ A := (1, 1, 2),\ b := 3.$$

The LCP (5.3.2) with these Q, A, b and c is monotone, since Q is positive definite. The problem has the unique solution $\bar{x} = \left(\frac{4}{3}, \frac{7}{9}, \frac{4}{9}, \frac{2}{9}\right)^T$ with $F(\bar{x}) = (0, 0, 0, 0)^T$. Clearly the solution $\bar{x}$ is nondegenerate.

Problem 5.2 (LCP derived from [35, Problem 76]):

$$Q := \begin{pmatrix} 2 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \ c := \begin{pmatrix} -1 \\ -3 \\ 1 \\ -1 \end{pmatrix}, \ A := \begin{pmatrix} 1 & 2 & 1 & 1 \\ 3 & 1 & 2 & -1 \\ 0 & -1 & -4 & 0 \end{pmatrix}, \ b := \begin{pmatrix} 5 \\ 4 \\ -3/2 \end{pmatrix}.$$

The LCP (5.3.2) with these Q, A, b and c is monotone, since Q is positive definite. The problem has the unique solution $\bar{x} = \left(\frac{3}{11}, \frac{23}{11}, 0, \frac{6}{11}, \frac{5}{11}, 0, 0\right)^T$.

Since $F(\bar{x}) = \left(0, 0, \frac{19}{11}, 0, 0, \frac{18}{11}, \frac{13}{22}\right)^T$, the solution $\bar{x}$ is nondegenerate.

Problems 5.3, 5.4 (LCP): Problems 5.3 and 5.4 are derived from convex QPs `MOSARQP1` and `MOSARQP2` in [4], respectively. `MOSARQP1` has 2500 variables and 700 constraints, and `MOSARQP2` has 900 variables and 600 constraints. Problems 5.3 and 5.4 have a unique solution because Q is positive definite in these problems. Moreover, in view of the results obtained by Algorithm PPA, the solutions of Problems 5.3 and 5.4 are nondegenerate.

Problem 5.5 (LCP): This problem has the same Q and A as Problem 5.1, but c and b are replaced by

$$c := \begin{pmatrix} -8/3 \\ -10/3 \\ -4/3 \end{pmatrix}, \; b := 5/3.$$

This problem is still monotone and has the unique solution $\bar{x} = \left(0, \frac{7}{9}, \frac{4}{9}, \frac{2}{9}\right)^T$. Since $F(\bar{x}) = (0,0,0,0)^T$, the solution is degenerate.

Problem 5.6 (LCP): This problem has the same Q and A as Problem 5.2, but c and b are replaced by

$$c := \begin{pmatrix} -5/11 \\ -3 \\ -1 \\ -1 \end{pmatrix}, \; b := \begin{pmatrix} 52/11 \\ 4 \\ -3/2 \end{pmatrix}.$$

This problem is still monotone and has the unique solution $\bar{x} = \left(0, \frac{23}{11}, 0, \frac{6}{11}, \frac{5}{11}, 0, 0\right)^T$. Since $F(\bar{x}) = (0,0,0,0,0,\frac{27}{11},\frac{13}{22})^T$, the solution is degenerate.

Problems 5.7, 5.8 (LCP): Problems 5.7 and 5.8 are constructed from Problems 5.3 and 5.4, respectively. Problem 5.7 has the same Q and A as Problem 5.3, but $(c,b)^T$ is replaced by $(c,b)^T - d^T$, where d is determined as follows: Let $\tilde{N}(\bar{x})$ be an index set consisting of 10 elements chosen from $N(\bar{x})$, where $\bar{x}$ is a solution of Problem 5.3. Then we set $d_i := F_i(\bar{x})$ if $i \in \tilde{N}(\bar{x})$, and $d_i := 0$ otherwise. This problem has a degenerate solution. Moreover, it is unique because Q is positive definite. Problem 5.8 is constructed from Problem 5.4 in a similar manner.

- **Problems 5.9–5.13:** Problems 5.9–5.13 are NCPs, not LCPs. In addition, Problems 5.9–5.12 are monotone, and hence it is guaranteed that Algorithm PPA finds a solution and the index sets are identified correctly. However, Problem 5.13 is not monotone, thus Algorithm PPA may fail to find a solution.

Problem 5.9 (NCP(F)):

$$F(x) := \begin{pmatrix} x_1 - 2 \\ x_2^3 + x_2 - x_3 + 3 \\ x_2 + 2x_3^3 + x_3 - 3 \end{pmatrix}$$

The unique solution of this problem is $\bar{x} = (2,0,1)^T$ and $F(\bar{x}) = (0,2,0)^T$. The solution is nondegenerate.

Problem 5.10 (NCP(F)):

$$F(x) := \begin{pmatrix} x_1 - 2 \\ x_2^3 + x_2 - x_3 + 1 \\ x_2 + 2x_3^3 + x_3 - 3 \end{pmatrix}$$

The unique solution of this problem is $\bar{x} = (2,0,1)^T$ and $F(\bar{x}) = (0,0,0)^T$. The solution is degenerate.

Problem 5.11 (NCP(F)):

$$F(x) := \begin{pmatrix} 0 \\ x_2^3 + x_2 - x_3 + 3 \\ x_2 + 2x_3^3 + x_3 - 3 \end{pmatrix}$$

The solution set of this problem is given by $X^* = \{(a,0,1)^T \mid a \geq 0\}$, and hence the solution is not locally unique. Moreover, since $F(x) = (0,2,0)^T$ for all solutions, the particular solution $(0,0,1)^T$ is degenerate.

Problem 5.12 (NCP(F) [58]):

$$F(x) := \begin{pmatrix} x_1^3 - 8 \\ x_2 + x_2^3 - x_3 + 3 \\ x_2 + x_3 + 2x_3^3 - 3 \\ x_4 + 2x_4^3 \end{pmatrix}$$

The unique solution of this problem is $\bar{x} = (2,0,1,0)^T$ and $F(\bar{x}) = (0,2,0,0)^T$. The solution is degenerate.

Problem 5.13 (NCP(F) [37]):

$$F(x) := \begin{pmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_2^2 + x_1 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}$$

The problem has two solutions $\bar{x}^1 = \left(\frac{\sqrt{6}}{2}, 0, 0, \frac{1}{2}\right)^T$ and $\bar{x}^2 = (1,0,3,0)^T$. Moreover, since $F(\bar{x}^1) = \left(0, \frac{\sqrt{6}}{2} + 2, 0, 0\right)^T$, the solution $\bar{x}^1$ is degenerate. On the other hand, $\bar{x}^2$ is nondegenerate as $F(\bar{x}^2) = (0,31,0,4)^T$.

- **Problem 5.14** (LCP): Problem 5.14 is randomly generated as follows: Let n be an even positive integer. Firstly, we construct a matrix $Q \in \Re^{n \times \frac{n}{2}}$ whose elements Q_{ij} are given by

$$Q_{ij} := \begin{cases} -1 & \text{if } \tau_{ij} \in [0, 1/3] \\ 0 & \text{if } \tau_{ij} \in (1/3, 2/3] \\ 1 & \text{otherwise,} \end{cases}$$

with a random number τ_{ij} uniformly distributed in the interval $[0,1]$. Secondly, we let $M \in \Re^{n \times n}$ and $q \in \Re^n$ be

$$M := QQ^T, \; q := -Ma + b,$$

where $a := (0,1,0,1,...)^T \in \Re^n$ and $b := (1,0,0,0,1,0,0,0,1,...)^T \in \Re^n$. Note that M is singular. Lastly, we define $F : \Re^n \to \Re^n$ by $F(x) := Mx + q$. This problem is an LCP with the following properties:

Table 5.1: Number of iterations versus index sets identification for Problems 5.1–5.8

Problem	5.1		5.2		5.3		5.4		5.5		5.6		5.7		5.8	
Initial point	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$
$x^{0,1}$	7	5	7	4	11	7	10	7	7	4	9	6	11	7	10	7
$x^{0,2}$	7	4	7	4	11	7	10	7	7	4	9	6	11	7	10	7
$x^{0,3}$	8	5	8	5	14	13	13	12	7	5	9	6	14	13	13	12
$x^{0,4}$	8	5	8	5	14	13	14	13	8	5	9	6	13	13	14	13

– It is monotone because M is positive semidefinite.

– It has a degenerate solution $\bar{x} = a$.

– $\|H_F(x)\|$ provides a local error bound since F is affine.

In the experiments, we used the following slight modification of the identification function:

$$\rho^k(x) := \min\left\{10^{-3}, \eta\left(\frac{\|H_F(x)\|}{c_k}\right)^{\beta}\right\}.$$

Clearly this modification does not affect the theoretical results established in the previous section. Moreover, we have slightly modified the algorithm when we solved LCPs (Problems 5.1–5.8 and 5.14). In fact, when F is affine and monotone, $\|H_{F^k}(x)\|$ provides a global error bound for NCP(F^k), i.e., there exists a constant δ independent of k such that

$$\|x^{k+1} - \bar{x}^{k+1}\| \leq \frac{\delta}{c_k}\|H_{F^k}(x^{k+1})\|,$$

where $\bar{x}^{k+1}$ is the unique solution of NCP(F^k). This can be verified from Theorem 2.3.1 (i) by observing the fact that F^k is globally Lipschitz and strongly monotone with modulus c_k. Therefore, we may employ the criterion

$$\|H_{F^k}(\tilde{x}^{k+1})\| \leq (c_k)^{\frac{3}{2}} \min\{1, \|\tilde{x}^{k+1} - x^k\|\} \tag{5.3.3}$$

instead of (2.4.1) and (2.4.3). This modification does not affect the convergence properties shown in Theorem 2.4.1. Moreover it is expected to make the algorithm simpler and robuster than the original Algorithm PPA, because the original criterion (2.4.1) has the fourth power of α, and hence it may cause a numerical difficulties when the number of iterations becomes large. We have used the criterion (5.3.3) for Problems 5.1–5.8 and 5.14, whereas the original criteria (2.4.1) and (2.4.3) for Problems 5.9–5.13.

Firstly, we mention the results for Problems 5.1–5.8. We used four initial points $x^{0,1} = (0, 0, ..., 0)^T$, $x^{0,2} = (1, 1, ..., 1)^T$, $x^{0,3} = (\frac{n}{2}, \frac{n}{2}, ..., \frac{n}{2})^T$ and $x^{0,4} = (n, n, ..., n)^T$, and set parameters as $\alpha = 0.5$, $B = 10^5$, $\eta = 10^{-1}$ (for Problems 5.1, 5.2, 5.5 and 5.6), $\eta = 10^{-3}$ (for Problems 5.3, 5.4, 5.7 and 5.8), and $\beta = 0.9$. We employed $\|H_F(x^k)\| < 10^{-8}$ as the termination criterion of Algorithm PPA. Table 5.1 shows two numbers $\hat{k}$ and $\bar{k}$ for Problems 5.1–5.8, where $\hat{k}$ is the iteration number at which the termination criterion was satisfied for the first time, and $\bar{k}$ is the iteration number at which the index sets P^*, N^* and C^* were identified correctly for the first time.

Table 5.2: Results for Problem 5.2 with initial point $x^{0,1}$

Index	1	2	3	4	5	6	7	$\|H_F(x^k)\|$
$k = 0$	C	C	N	C	N	N	C	7.3e+00
1	P	P	N	P	N	N	N	2.0e+00
2	-	P	N	P	P	N	N	2.9e−01
3	-	-	N	-	P	N	N	2.7e−02
4	P	P	N	P	P	N	N	9.9e−04
5	P	P	N	P	P	N	N	2.4e−05
6	P	P	N	P	P	N	N	4.4e−07
7	P	P	N	P	P	N	N	4.9e−09

Table 5.3: Results for Problem 5.6 with initial point $x^{0,2}$

Index	1	2	3	4	5	6	7	$\|H_F(x^k)\|$
$k = 0$	-	P	P	-	P	P	-	2.0e+00
1	-	P	-	-	-	-	-	7.1e−01
2	-	P	-	-	-	N	N	2.3e−01
3	-	P	-	P	P	N	N	4.6e−02
4	-	P	-	P	P	N	N	1.3e−02
5	P	P	-	P	P	N	N	2.6e−03
6	C	P	C	P	P	N	N	2.8e−04
7	C	P	C	P	P	N	N	1.6e−05
8	C	P	C	P	P	N	N	4.9e−07
9	C	P	C	P	P	N	N	7.4e−09

Table 5.4: Number of iterations versus index sets identification for Problems 5.9–5.13

Problem	5.9		5.10		5.11		5.12		5.13	
Initial point	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$	$\hat{k}$	$\bar{k}$
$x^{0,1}$	12	1	12	3	8	1	8	1	12	1
$x^{0,2}$	12	1	12	6	6	1	11	8	10	9
$x^{0,3}$	12	4	12	7	7	5	12	8	12	6
$x^{0,4}$	12	7	12	7	8	5	12	8	11	9

Table 5.5: Results for Problem 5.11 with initial point $x^{0,3}$

Index	1	2	3	$\|H_F(x^k)\|$
$k = 0$	P	-	-	1.9e+00
1	P	N	-	3.0e−01
2	P	N	-	2.7e−02
3	P	N	-	1.9e−03
4	P	N	-	1.0e−04
5	P	N	P	4.6e−06
6	P	N	P	1.6e−07
7	P	N	P	4.8e−09

The generated sequence converged to the solution $x^* = (1.5, 0, 1)^T$.

The numbers $\hat{k}$ and $\bar{k}$ in Table 5.1 show that the index sets were successfully identified for Problems 5.1–5.8 as guaranteed by Theorem 5.2.1. Especially, the results for Problems 5.7 and 5.8 indicate that the proposed identification method can be applied to degenerate large scale LCPs effectively. Tables 5.2 and 5.3 show how the index sets are identified as the iteration proceeds on some LCPs. In these tables, "P", "N" and "C" mean that index i belongs to P^k, N^k and C^k, respectively, at the kth iteration, while "-" means that index i belong to none of P^k, N^k and C^k.

Secondly, we discuss Problems 5.9–5.13. We set parameters as $\alpha = 0.8$, $B = 10^5$, $\eta = 10^{-1}$, and $\beta = 0.9$. We employed the same termination criterion $\|H_F(x)\| < 10^{-8}$ as before. For these problems, α is chosen relatively large to avoid numerical difficulties that may occur in (2.4.1) and (2.4.3). Table 5.4 shows that the index sets were identified correctly for these problems. Especially, we were able to identify the index sets successfully for Problem 5.13, which is not a monotone NCP. Tables 5.5 and 5.6 show how the index sets are identified as the iteration proceeds on some NCPs. In these tables, "P", "N", "C" and "-" have the same meaning as in Tables 5.2 and 5.3.

Finally we give the results for Problem 5.14. We generated 100 problems with $n = 100$, and solved them. We used four initial points $x^{0,1}, x^{0,2}, x^{0,3}$ and $x^{0,4}$, and set parameters as $\alpha = 0.1$ and 0.5, $B = 10^5$, $\eta = 10^{-1}$, and $\beta = 0.9$. We employed the same termination criterion $\|H_F(x)\| < 10^{-8}$ as before. The averages of $\hat{k}$ and $\bar{k}$ for 100 problems are shown in Tables 5.7 and 5.8. A closer look at the solutions obtained for those 100 problems has revealed that the number of indices belonging to C^* are 9.60 on average.

Table 5.6: Results for Problem 5.13 with initial point $x^{0,4}$

Index	1	2	3	4	$\|H_F(x^k)\|$
$k=0$	-	-	-	-	7.8e+00
1	-	N	N	-	1.3e+00
2	-	N	N	-	2.7e−01
3	P	N	P	-	1.1e−01
4	-	N	N	-	6.8e−02
5	P	N	N	-	6.8e−03
6	P	N	N	-	8.9e−04
7	P	N	N	P	9.7e−05
8	P	N	N	P	8.8e−06
9	P	N	C	P	6.4e−07
10	P	N	C	P	3.8e−08
11	P	N	C	P	1.8e−09

The generated sequence converged to the solution $x^* = \left(\frac{\sqrt{6}}{2}, 0, 0, \frac{1}{2}\right)^T$.

Table 5.7: Average of $\hat{k}$ and $\bar{k}$ for Problem 5.14 with $\alpha = 0.1$

Initial point	average of $\hat{k}$	average of $\bar{k}$
$x^{0,1}$	5.95	4.75
$x^{0,2}$	5.95	4.38
$x^{0,3}$	7.04	4.97
$x^{0,4}$	7.18	4.82

Table 5.8: Average of $\hat{k}$ and $\bar{k}$ for Problem 5.14 with $\alpha = 0.5$

Initial point	average of $\hat{k}$	average of $\bar{k}$
$x^{0,1}$	8.78	6.78
$x^{0,2}$	8.18	5.72
$x^{0,3}$	12.76	11.12
$x^{0,4}$	13.16	11.86

5.4 Application

In this section, we give an application of the identification technique presented in Section 5.2. As mentioned in Section 5.1, if we can identify the sets $P(x^*), N(x^*)$ and $C(x^*)$, we may obtain x^* by solving a system of nonlinear equations and inequalities. Here we will further reduce the system into a problem that involves no inequalities. More specifically, we show that, by using Theorem 5.2.1, a solution of the equality constrained minimization problem

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\|x - x^k\|^2 \\ &\text{subject to} \quad F_{P^k}(x) = 0,\ x_{N^k} = 0, \\ &\qquad\qquad\quad F_{C^k}(x) = 0,\ x_{C^k} = 0 \end{aligned} \tag{5.4.1}$$

eventually becomes a solution of the original NCP(F). Note that, if F is affine, then problem (5.4.1) is an equality constrained quadratic programming problem which can be solved by way of the system of linear equations [1] derived from its KKT conditions.

Theorem 5.4.1 *Let Assumption 5.2.1 be satisfied. Let $\{x^k\}$ be generated by Algorithm PPA and $\{\rho^k\}$ be defined by* (5.2.3). *If the index sets P^k, N^k and C^k are determined by* (5.2.2) *with the identification function sequence $\{\rho^k\}$, then, for all k sufficiently large, a solution of the minimization problem* (5.4.1) *is a solution of NCP(F).*

Proof: Let S denote the feasible region of (5.4.1). Since $x^k \to x^*$ and $P^k = P(x^*)$, $N^k = N(x^*)$, $C^k = C(x^*)$ for all k sufficiently large by Theorem 5.2.1, we have $F_{P^k}(x^*) = 0, x^*_{N^k} = 0, F_{C^k}(x^*) = 0, x^*_{C^k} = 0$. Therefore $x^* \in S$, and hence $S \neq \emptyset$. Since the objective function of (5.4.1) is strongly convex and S is closed by the continuity of F, the minimization problem (5.4.1) has a solution for all k sufficiently large.

Let $\bar{x}$ be a solution of (5.4.1). Since $\bar{x} \in S$, $\bar{x}$ is a solution of NCP(F) if $\bar{x}_{P^k} > 0$ and $F_{N^k}(\bar{x}) > 0$. Since $x^*_{P^k} > 0, F_{N^k}(x^*) > 0$ for all k sufficiently large and F is continuous, there exists $\epsilon > 0$ such that

$$\|x - x^*\| < \epsilon \implies x_{P^k} > 0, F_{N^k}(x) > 0.$$

Let k be large enough to satisfy $\|x^* - x^k\| < \frac{\epsilon}{2}$. Since $x^* \in S$ and $\bar{x}$ is a solution of (5.4.1), we have

$$\|\bar{x} - x^k\| \le \|x^* - x^k\| < \frac{\epsilon}{2}.$$

Hence, we obtain

$$\|\bar{x} - x^*\| \le \|\bar{x} - x^k\| + \|x^k - x^*\| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon,$$

which means that $\bar{x}_{P^k} > 0$ and $F_{N^k}(\bar{x}) > 0$. Consequently $\bar{x}$ is a solution of NCP(F). □

This theorem enables us to modify Algorithm PPA to construct a method with a finite termination property, which attempts to find a solution of NCP(F) by solving the minimization problem (5.4.1) with the index sets determined by the identification function sequence $\{\rho^k\}$. The prototype of the method is described as follows.

[1] The system becomes singular when C^k is not empty.

Algorithm PPA-MIN

Step 0: Choose parameters $\alpha \in (0,1), B \in (0,\infty)$ and an initial point $x^0 \in \Re^n$. Set $c_0 := 1$ and $k := 0$.

Step 1: If $\|H_F(x^k)\|$ is sufficiently small, then go to Step 2. Otherwise go to Step 3.

Step 2: Solve the equality constrained minimization problem (5.4.1). If it is solvable and the solution satisfies $x_{P^k} \geq 0$ and $F_{N^k}(x) \geq 0$, then stop.

Step 3: Let $F^k : \Re^n \to \Re^n$ be defined by

$$F^k(x) := F(x) + c_k(x - x^k).$$

Obtain an approximate solution $\tilde{x}^{k+1}$ of NCP(F^k) which satisfies

$$\|H_{F^k}(\tilde{x}^{k+1})\| \leq c_k^4 \|\tilde{x}^{k+1} - x^k\|,$$

$$\|\tilde{x}^{k+1} - [\tilde{x}^{k+1}]_+\| \leq B$$

and

$$\Psi^k([\tilde{x}^{k+1}]_+) \leq \frac{c_k^3}{4\max\{1, \|[\tilde{x}^{k+1}]_+\|\}^2}$$

where

$$\Psi^k(x) := \sum_{i=1}^n (|x_i F_i^k(x)| + |\min\{x_i, F_i^k(x)\}|).$$

Step 4: Set $x^{k+1} := [\tilde{x}^{k+1}]_+$, $c_{k+1} := \alpha c_k$, and $k := k+1$. Go to Step 1.

Since F^k is strongly monotone, an approximate solution $\tilde{x}^{k+1}$ can be obtained in Step 3 by a suitable solution method such as the generalized Newton method [12, 45].

Problem (5.4.1) solved in Step 2 is a nonlinear programming problem, which in general may not be solved easily. However, when we deal with LCP, the minimization problem (5.4.1) is reduced to a system of linear equations as mentioned earlier. Therefore, Algorithm PPA-MIN can find a solution of the monotone LCP in a finite number of arithmetic operations.

5.5 Concluding remarks

In this chapter, we have proposed a technique which identifies degenerate indices at a solution of the monotone NCP. Traditional identification techniques in previous research require the local uniqueness of a solution, but our technique can be performed without assuming it. However, our technique deeply depends on the sequence generated by Algorithm PPA, so we cannot apply it directly to another methods for the monotone NCP. Then we want to develop another identification technique which can be applied to any method for the monotone NCP.

Chapter 6

A Superlinearly Convergent Algorithm for Monotone NCP

6.1 Introduction

The nonlinear complementarity problem (NCP) [22] is to find a vector $\bar{x} \in \Re^n$ such that

$$\mathrm{NCP}(F): \quad \bar{x}_i \geq 0, \;\; F_i(\bar{x}) \geq 0, \;\; \bar{x}_i F_i(\bar{x}) = 0, \quad i = 1, \ldots, n,$$

where F is a mapping from $\Re^n$ to $\Re^n$. When F is affine, NCP(F) is called the linear complementarity problem (LCP) [11]. NCP can be found in various fields, e.g., operations research, engineering, finance and so on [24]. Throughout this chapter, F is assumed to be continuously differentiable and monotone.

For a solution $\bar{x}$ of NCP(F), let $P(\bar{x}), N(\bar{x})$ and $C(\bar{x})$ be the index sets defined by

$$\begin{aligned} P(\bar{x}) &:= \{i \mid \bar{x}_i > 0, F_i(\bar{x}) = 0\}, \\ N(\bar{x}) &:= \{i \mid \bar{x}_i = 0, F_i(\bar{x}) > 0\}, \\ C(\bar{x}) &:= \{i \mid \bar{x}_i = 0, F_i(\bar{x}) = 0\}, \end{aligned}$$

respectively. Note that $P(\bar{x}) \cup N(\bar{x}) \cup C(\bar{x}) = \{1, \ldots, n\}$ and these index sets are mutually disjoint. If $C(\bar{x}) \neq \emptyset$, we call $\bar{x}$ a degenerate solution, otherwise we call $\bar{x}$ a nondegenerate solution. In this chapter, we will focus on these index sets for a particular solution, and develop an algorithm that is designed to solve NCP(F) by estimating the correct index sets.

Various methods for solving NCP, such as the generalized Newton method (GNM) [12, 45], the smoothing method [7, 8, 9, 29] and the regularization method [21], have been proposed and shown to have nice convergence properties. However, those methods generally require the local uniqueness of a solution for a superlinear rate of convergence. Recently, Yamashita and Fukushima [59] proposed a method, which is introduced in Section 2.4 as Algorithm PPA, and showed that it has a superlinear rate of convergence without the local uniqueness of a solution. However, when a sequence $\{x^k\}$ generated by Algorithm PPA converges to a degenerate solution, subproblems may become computationally expensive. This difficulty comes from the fact that we do not know in advance whether or not $\{x^k\}$ converges to a degenerate solution. In Chapter 5, we proposed a technique that enables us to identify $P^* = P(x^*), N^* = N(x^*)$ and $C^* = C(x^*)$ when x^k enters a certain vicinity of x^*. Once we identify P^*, N^* and C^*, we can find a solution of NCP(F) by solving the system of nonlinear equations (SNE)

$$G_{x^*}(x) := \begin{pmatrix} F_{P^*}(x) \\ x_{N^*} \\ F_{C^*}(x) \\ x_{C^*} \end{pmatrix} = 0, \tag{6.1.1}$$

where $F_{P^*}(x)$ is the vector with components $F_i(x)$, $i \in P^*$, x_{N^*} is the vector with components x_i, $i \in N^*$, and so on. In fact, if a solution $\hat{x}$ of (6.1.1) is sufficiently near to x^*, then we have $\hat{x}_{P^*} > 0$ and $F_{N^*}(\hat{x}) > 0$, and hence $\hat{x}$ is also a solution of NCP(F). Moreover, since the mapping $G_{x^*} : \Re^n \to \Re^{n+|C^*|}$ is differentiable, we can use any Newton-type method to solve (6.1.1), and we can expect that such a method has a quadratic or superlinear rate of convergence no matter whether x^* is degenerate or nondegenerate.

In this chapter, we construct a differentiable SNE (6.1.1) by using the technique presented in Chapter 5. Moreover, we propose a hybrid algorithm that generates a sequence $\{x^k\}$ by Algorithm PPA primarily and also tries to find a solution of an SNE (6.1.1) by using the inexact Levenberg-Marquardt method (ILMM) proposed in Chapter 4, when an iterative point x^k is judged to lie sufficiently close to a solution. We show that the proposed method has a quadratic or superlinear convergence property if $\|G_{x^*}(x)\|$ provides a local error bound for (6.1.1), i.e., there exists constants $b_G > 0$ and $c_G > 0$ such that

$$c_G \mathrm{dist}(x, x^*_G) \leq \|G_{x^*}(x)\| \quad \forall x \in B(x^*, b_G), \tag{6.1.2}$$

where x^*_G is the solution set of (6.1.1). More specifically, we show that either a sequence $\{x^k\}$ generated by the proposed algorithm converges to the solution set of NCP quadratically or $\{\|G_{x^*}(x^k)\|\}$ converges to 0 superlinearly, without assuming the nondegeneracy of a solution.

This chapter is organized as follows: In Section 6.2, we review some results which are used in this chapter. Actually, these results are introduced in previous discussions of this thesis. In Section 6.3, we construct a differentiable SNE whose solution is a solution of NCP(F). Moreover, we propose the hybrid algorithm based on (6.1.1) and show its global and superlinear convergence. In Section 6.4, we make some concluding remarks and discuss future research topics.

6.2 Preliminaries

In this section, we review some results which are used in the subsequent discussion. These results were introduced in this thesis until now, and we summarize main results which will be used in this chapter.

6.2.1 Identification of the index sets

In this subsection, we recall the identification technique of the index sets, which is proposed in Chapter 5.

When x^* is a degenerate solution of NCP(F), Theorem 2.4.2 no longer guarantees that the GNM can find a solution of subproblem NCP(F^k) in a few iterations, and hence it may take much time to solve subproblems. To overcome this difficulty, we will use the technique proposed in Chapter 5 to identify the index sets $P(x^*), N(x^*)$ and $C(x^*)$. The following theorem has been established in Chapter 5.

Theorem 6.2.1 *Suppose that $\|H_F(x)\|$ provides a local error bound for NCP(F). Let a sequence $\{x^k\}$ be generated by Algorithm PPA and x^* be the limit point of $\{x^k\}$. Let a function sequence $\{\rho^k\}$ be defined by*

$$\rho^k(x) = \eta \left(\frac{\|H_F(x)\|}{c_k} \right)^{\beta},$$

where $\eta > 0$ and $\beta \in (0,1)$, and the index sets P^k, N^k and C^k be defined by

$$\begin{array}{rcl} P^k & := & \{i \mid x_i^k > \rho^k(x^k), F_i(x^k) \le \rho^k(x^k)\}, \\ N^k & := & \{i \mid x_i^k \le \rho^k(x^k), F_i(x^k) > \rho^k(x^k)\}, \\ C^k & := & \{i \mid x_i^k \le \rho^k(x^k), F_i(x^k) \le \rho^k(x^k)\}, \end{array} \tag{6.2.1}$$

respectively. Then, we have

$$P^k = P(x^*), \quad N^k = N(x^*), \quad C^k = C(x^*)$$

for sufficiently large k.

6.2.2 Inexact Levenberg-Marquardt method

In this subsection, we recall the inexact Levenberg-Marquardt method (ILMM) and summarize main results of Chapter 4.

The Levenberg-Marquardt method (LMM) [1, 28, 31, 60] is a method for solving the SNE

$$G(y) = 0, \tag{6.2.2}$$

where $G : \Re^n \to \Re^m$ is a continuously differentiable mapping. The LMM generates a sequence $\{y^l\}$ by $y^{l+1} := y^l + \hat{d}^l$, where $\hat{d}^l$ is a solution of the system of linear equations

$$\left(\nabla G(y^l)^T \nabla G(y^l) + \mu_l I \right) d = -\nabla G(y^l)^T G(y^l). \tag{6.2.3}$$

Here $\nabla G(y) \in \Re^{m \times n}$ is the Jacobian of G at y, μ_l is a positive parameter, and I is the identity matrix. Since $\nabla G(y^l)^T \nabla G(y^l) + \mu_l I$ is positive definite, (6.2.3) has a unique solution. However, it is expensive to find an exact solution of (6.2.3) when n is large. In that case, the inexact Levenberg-Marquardt method (ILMM) [20] is useful. The ILMM uses an approximate solution d^l of (6.2.3) as a search direction, and generates a sequence $\{y^l\}$ by $y^{l+1} := y^l + d^l$.

In Chapter 4, we show that the ILMM has a quadratic rate of convergence under a local error bound condition, which is milder than the nonsingularity condition at a solution. This result is stated in the following theorem.

Theorem 6.2.2 *Let $\{y^l\}$ be a sequence generated by the ILMM and y^* be a solution of (6.2.2). Suppose that the following two conditions hold:*
(i) *There exist constants $b_1 \in (0,1)$ and $\kappa_1 \in (0,\infty)$ such that*

$$\|\nabla G(x)(y-x) - (G(y) - G(x))\| \leq \kappa_1 \|y-x\|^2 \quad \forall x, y \in B(y^*, b_1).$$

(ii) *$\|G(y)\|$ provides an error bound for (6.2.2) on $B(y^*, b_1)$, i.e., there exists a constant $\kappa_2 > 0$ such that*

$$\kappa_2 \operatorname{dist}(y, Y^*) \leq \|G(y)\| \quad \forall y \in B(y^*, b_1),$$

where Y^* is the solution set of (6.2.2).
Let the parameters μ_l be chosen as $\mu_l = \|G(y^l)\|^2$ and the residual vectors r^l be defined by

$$r^l := \left(\nabla G(y^l)^T \nabla G(y^l) + \mu_l I\right) d^l + \nabla G(y^l)^T G(y^l). \tag{6.2.4}$$

If an initial point y^0 is sufficiently close to y^* and

$$\frac{\|r^l\|}{\mu_l} = O\left(\operatorname{dist}(y^l, Y^*)^2\right)$$

holds for all l, then there exists a constant $c > 0$ such that $\{y^l\} \subseteq B(y^*, c\|y^0 - y^*\|)$. Moreover, $\{\operatorname{dist}(y^l, Y^*)\}$ converges to 0 quadratically.

The assumption (i) of Theorem 6.2.2 holds when ∇G is locally Lipschitzian [41, Theorem 3.2.12].

6.3 Proposed algorithm and its convergence properties

As shown in Theorem 2.4.1, Algorithm PPA enjoys a nice convergence property for the monotone NCP. However, if x^* is a degenerate solution of NCP(F), it is not guaranteed theoretically that the sequence $\{x^k\}$ converges to x^* superlinearly in a genuine sense. In this section, we describe the proposed algorithm and show that it has at least a superlinear rate of convergence for NCP in a genuine sense, without assuming the local uniqueness and the nondegeneracy of a solution.

6.3.1 Differentiable nonlinear equations

First, we make the following assumption to guarantee that Algorithm PPA has a superlinear convergence and the technique to identify the index sets works well.

Assumption 6.3.1 (i) *∇F is locally Lipschitzian.*
(ii) *$\|H_F(x)\|$ provides a local error bound for NCP(F), i.e., there exist constants $b_H > 0$ and $c_H > 0$ such that*

$$\|H_F(x)\| \leq c_H \implies b_H \operatorname{dist}(x, x^*) \leq \|H_F(x)\|,$$

where x^* is the solution set of NCP(F).

Sufficient conditions under which $\|H_F(x)\|$ provides a local error bound for NCP(F) are given in Theorem 2.3.1 (i).

We consider the mapping $G_{x^*} : \Re^n \to \Re^{n+|C^*|}$ defined by (6.1.1), where x^* is a solution of NCP(F). Since $x_{P^*} > 0$ and $F_{N^*}(x) > 0$ in a sufficiently small neighborhood of x^*, any solution $\hat{x}$ of (6.1.1) also solves NCP(F) if $\hat{x}$ is sufficiently close to x^*. We note that G_{x^*} is differentiable and ∇G_{x^*} is locally Lipschitzian, and hence assumption (i) of Theorem 6.2.2 holds for G_{x^*}. So, the ILMM applied to the system of equations (6.1.1) has a quadratic rate of convergence if $\|G_{x^*}(x)\|$ provides a local error bound. Accordingly, we make the following assumption.

Assumption 6.3.2 *$\|G_{x^*}(x)\|$ provides a local error bound for (6.1.1) in a neighborhood of x^*, i.e., there exist positive constants b_G and c_G such that (6.1.2) holds.*

Assumption 6.3.2 holds when F_{P^*} and F_{C^*} are affine. Though this assumption does not necessarily hold when F is nonlinear, it does not seem very restrictive. Assumptions 6.3.1 and 6.3.2 are closely related. In fact, as shown in the next lemma, Assumption 6.3.2 is implied by Assumption 6.3.1 under some normal circumstances.

Lemma 6.3.1 *Suppose that Assumption 6.3.1 holds. If there exists a constant $r_1 > 0$ such that $x^* \cap B(x^*, r_1) = x^*_G \cap B(x^*, r_1)$, where x^*_G is the solution set of (6.1.1), then $\|G_{x^*}(x)\|$ provides a local error bound for (6.1.1) in a neighborhood of x^*, i.e., there exist constants b_G and c_G such that (6.1.2) holds. In particular, if x^* is a locally unique solution of NCP(F), then $\|G_{x^*}(x)\|$ provides a local error bound for (6.1.1) in a neighborhood of x^*.*

Proof: Choosing $r_2 > 0$ sufficiently small yields that, for any $x \in B(x^*, r_2)$,

$$x_i \geq F_i(x) \quad \forall i \in P^*,$$
$$x_i \leq F_i(x) \quad \forall i \in N^*.$$

It then follows that, for any $x \in B(x^*, r_2)$,

$$\begin{aligned} \|H_F(x)\| &= \sqrt{\sum_{i\in P^*\cup N^*\cup C^*} \phi^2(x_i, F_i(x))} \\ &\leq \nu_2 \sqrt{\sum_{i\in P^*\cup N^*\cup C^*} |\min\{x_i, F_i(x)\}|^2} \\ &\leq \nu_2 \sqrt{\sum_{i\in P^*} F_i^2(x) + \sum_{i\in N^*} x_i^2 + \sum_{i\in C^*} \left(x_i^2 + F_i^2(x)\right)} \\ &= \nu_2 \|G_{x^*}(x)\|, \end{aligned} \tag{6.3.1}$$

where ν_2 is the positive constant given in (2.2.1). Moreover, by choosing $r_3 > 0$ sufficiently small, we have

$$\|G_{x^*}(x)\| \leq c_H/\nu_2 \quad \forall x \in B(x^*, r_3). \tag{6.3.2}$$

Let $r := \min\{r_1, r_2, r_3\}$. It then follows from (6.3.1) and (6.3.2) that

$$\|H_F(x)\| \leq c_H \quad \forall x \in B(x^*, r).$$

Therefore, from Assumption 6.3.1 (ii) and (6.3.1), we get

$$b_H \mathrm{dist}(x, x^*) \leq \|H_F(x)\| \leq \nu_2 \|G_{x^*}(x)\| \quad \forall x \in B(x^*, r). \tag{6.3.3}$$

Let $x \in B\left(x^*, \frac{r}{2}\right)$, and let $\bar{x}_G$ and $\bar{x}$ be one of the nearest points from x in x_G^* and x^*, respectively. Since $x^* \in x_G^*$ and $x^* \in x^*$, we have, for any $x \in B\left(x^*, \frac{r}{2}\right)$,

$$\begin{aligned} \|\bar{x}_G - x\| &\leq \|x^* - x\| \leq \frac{r}{2}, \\ \|\bar{x} - x\| &\leq \|x^* - x\| \leq \frac{r}{2}, \end{aligned}$$

and hence

$$\begin{aligned} \|\bar{x}_G - x^*\| &\leq \|\bar{x}_G - x\| + \|x^* - x\| \leq r, \\ \|\bar{x} - x^*\| &\leq \|\bar{x} - x\| + \|x^* - x\| \leq r. \end{aligned}$$

Therefore we have $\bar{x}_G \in X_G^* \cap B(x^*, r)$ and $\bar{x} \in x^* \cap B(x^*, r)$. It then follows from the assumption $x^* \cap B(x^*, r) = x_G^* \cap B(x^*, r)$ that $\mathrm{dist}(x, x^*) = \mathrm{dist}(x, x_G^*)$ for any $x \in B\left(x^*, \frac{r}{2}\right)$. Consequently, by (6.3.3), we have

$$\nu_2^{-1} b_H \ \mathrm{dist}(x, x_G^*) \leq \|G_{x^*}(x)\| \quad \forall x \in B(x^*, \frac{r}{2}),$$

i.e., $\|G_{x^*}(x)\|$ provides a local error bound for (6.1.1) in a neighborhood of x^*. □

6.3.2 Algorithm

Let $\{x^k\}$ be a sequence generated by Algorithm PPA, x^* be the limit of $\{x^k\}$, and P^k, N^k, C^k be defined by (6.2.1). Then, as stated in Subsection 6.2.1, $P^k = P(x^*)$, $N^k = N(x^*)$ and $C^k = C(x^*)$ hold when k is sufficiently large, and hence, the SNE

$$G^k(x) := \begin{pmatrix} F_{P^k}(x) \\ x_{N^k} \\ F_{C^k}(x) \\ x_{C^k} \end{pmatrix} = 0 \tag{6.3.4}$$

coincides with the SNE (6.1.1). In this case, a solution $\hat{x}$ of (6.3.4) is a solution of NCP(F) if $\hat{x}$ is sufficiently close to x^*. Then we naturally come up with the following method: We generate a sequence $\{x^k\}$ by Algorithm PPA primarily, and if an iterative point is judged to be close to the solution x^*, then we solve (6.3.4) by the ILMM. Based on this idea, we propose the following hybrid algorithm.

Algorithm HYBRID

Step 0: Choose parameters $M_1 > 0, M_2 > 0, \alpha \in (0,1), \beta \in (0,1), \eta > 0, B \in (0, \infty), \gamma \in (0,1)$, and an initial point x^0. Set $P^0 = N^0 = C^0 = \emptyset$, $c_0 := 1$ and $k := 1$.

Step 1: Obtain x^k by applying a single iteration of Algorithm PPA. Determine the index sets P^k, N^k, C^k by (6.2.1).

Step 2: If $\|H_F(x^k)\| \leq M_1, P^k = P^{k-1}, N^k = N^{k-1}, C^k = C^{k-1}$ and $P^k \cup N^k \cup C^k = \{1, \ldots, n\}$, then go to Step 3. Otherwise, set $k := k+1$ and go to Step 1.

Step 3: [ILMM for (6.3.4)]

Step 3.0: Set $y^{k,0} := x^k$, $\mu_{k,0} := \|G^k(y^{k,0})\|^2$, and $l := 0$.

Step 3.1: If $y^{k,l}$ solves NCP(F), then terminate. Otherwise, find an approximate solution $d^{k,l}$ of the system of linear equations

$$\left(\nabla G^k(y^{k,l})^T \nabla G^k(y^{k,l}) + \mu_{k,l} I\right) d = -\nabla G^k(y^{k,l})^T G^k(y^{k,l}). \tag{6.3.5}$$

Step 3.2: If the conditions

$$(y^{k,l} + d^{k,l})_{P^k} > 0, \tag{6.3.6}$$

$$F_{N^k}(y^{k,l} + d^{k,l}) > 0, \tag{6.3.7}$$

$$\|G^k(y^{k,l} + d^{k,l})\| \leq (\gamma)^l \|G^k(y^{k,l})\| \tag{6.3.8}$$

$$\|x^k - (y^{k,l} + d^{k,l})\| \leq M_2 \tag{6.3.9}$$

are not satisfied, then set $k := k+1$ and go to Step 1.

Step 3.3: Set $y^{k,l+1} := y^{k,l} + d^{k,l}, \mu_{k,l+1} := \|G^k(y^{k,l+1})\|^2$ and $l := l+1$. Go to Step 3.1.

In Algorithm HYBRID, we try to identify the three index sets $P(x^*), N(x^*)$ and $C(x^*)$ in Step 2. The conditions in Step 2 will be satisfied when k is sufficiently large. Note, however, that those conditions do not guarantee that the index sets P^*, N^* and C^* are identified correctly. Therefore, we check conditions (6.3.6) – (6.3.9) in Step 3.2 to ensure that the sequence $\{y^{k,l}\}_{l=0,1,2,\ldots}$ is converging to a solution of NCP(F). Conditions (6.3.6) and (6.3.7) check whether P^k and N^k contain any wrong index. Condition (6.3.8) ensures that $\{\|G^k(y^{k,l})\|\}_{l=0,1,2,\ldots}$ is converging to 0 superlinearly, and condition (6.3.9) ensures that $\{y^{k,l}\}_{l=0,1,2,\ldots}$ is not diverging.

6.3.3 Convergence theorem

Let the residual vector $r^{k,l}$ associated with an approximate solution $d^{k,l}$ of the system of linear equations (6.3.5) be defined by

$$r^{k,l} := \left(\nabla G^k(y^{k,l})^T \nabla G^k(y^{k,l}) + \mu_{k,l} I\right) d^{k,l} + \nabla G^k(y^{k,l})^T G^k(y^{k,l}).$$

Now we show the following convergence theorem for Algorithm HYBRID.

Theorem 6.3.1 *Suppose that Assumptions 6.3.1 and 6.3.2 hold. Suppose also that, for any k, the sequence $\{y^{k,l}\}_{l=0,1,2,\ldots}$ generated in Step 3 of Algorithm HYBRID satisfies*

$$\frac{\|r^{k,l}\|}{\mu_{k,l}} = O\left(\mathrm{dist}(y^{k,l}, x^*_{G^k})^2\right), \tag{6.3.10}$$

where $x^*_{G^k}$ is the solution set of (6.3.4). Then, there exists a positive integer k for which either of the following statements is true:

(a): *Any accumulation point of the sequence* $\{y^{k,l}\}_{l=0,1,2,\ldots}$ *is a solution of NCP*(F)*, and* $\{\|G^k(y^{k,l})\|\}_{l=0,1,2,\ldots}$ *converges to* 0 *superlinearly.*

(b): $\{y^{k,l}\}_{l=0,1,2,\ldots}$ *converges to a solution of NCP*(F)*, and* $\{\mathrm{dist}(y^{k,l}, x_G^*)\}_{l=0,1,2,\ldots}$ *converges to* 0 *quadratically.*

Proof: First, we consider the case where the inner loop in Step 3 cycles infinitely for some k, that is, an infinite sequence $\{y^{k,l}\}_{l=0,1,2,\ldots}$ satisfies (6.3.6) – (6.3.9), no matter whether the index sets $P(x^*), N(x^*)$ and $C(x^*)$ have yet to be identified correctly. In this case, it follows from (6.3.9) that $\{y^{k,l}\}_{l=0,1,2,\ldots}$ has accumulation points, and from (6.3.8), $\{\|G^k(y^{k,l})\|\}_{l=0,1,2,\ldots}$ converges to 0 superlinearly. Moreover, from (6.3.6) and (6.3.7), any accumulation point of $\{y^{k,l}\}_{l=0,1,2,\ldots}$ is a solution of NCP(F). Therefore the statement (a) holds.

Next, we show that, even if (a) does not hold, statement (b) holds eventually. In fact, for sufficiently large k, $\|H_F(x^k)\|$ becomes sufficiently small by Theorem 2.4.1, and $P^k = P(x^*), N^k = N(x^*), C^k = C(x^*)$ hold by Theorem 6.2.1. Hence, the system (6.3.4) coincides with the system (6.1.1) for sufficiently large k. Note that, by Assumptions 6.3.1 and 6.3.2, G_{x^*} and x^* satisfy assumptions (i) and (ii) in Theorem 6.2.2, where G and y^* are regarded as G_{x^*} and x^*, respectively. In what follows, we consider a sequence $\{z^{k,l}\}_{l=0,1,2,\ldots}$ generated by the ILMM for the equation $G_{x^*}(x) = 0$ with an initial point $z^{k,0} := x^k$ for sufficiently large k, and show that $z^{k,l}$ satisfies the conditions (6.3.6)–(6.3.9) for all l. By Theorem 4.2.2 and (6.3.10), if k is sufficiently large and $z^{k,0} = x^k$ is sufficiently close to x^*, then we have $\|z^{k,l+1} - \bar{z}^{k,l+1}\| \leq c\,\|z^{k,l} - \bar{z}^{k,l}\|^2$ for all l, where c is a positive constant and $\bar{z}^{k,l}$ is one of the nearest points from $z^{k,l}$ in X_G^*. Then, from Assumption 6.3.2, we have

$$\begin{aligned}
\frac{\|G_{x^*}(z^{k,l+1})\|}{\|G_{x^*}(z^{k,l})\|^2} &= \frac{\|G_{x^*}(z^{k,l+1}) - G_{x^*}(\bar{z}^{k,l+1})\|}{\|G_{x^*}(z^{k,l})\|^2} \\
&\leq \frac{L_{x_*}\|z^{k,l+1} - \bar{z}^{k,l+1}\|}{c_G^2\|z^{k,l} - \bar{z}^{k,l}\|^2} \\
&\leq \frac{cL_{x^*}}{c_G^2} \qquad l = 0, 1, 2, \ldots,
\end{aligned}$$

where L_{x^*} is a Lipschitz constant of the mapping G_{x^*}. When k is sufficiently large, we have $\|G_{x^*}(z^{k,0})\| \leq (\gamma c_G^2)/(cL_{x^*})$, and it follows that for each $l = 0, 1, 2, \ldots$

$$\begin{aligned}
\|G_{x^*}(z^{k,l+1})\| &\leq \frac{cL_{x^*}}{c_G^2}\|G_{x^*}(z^{k,l})\|^2 \\
&\leq \left(\frac{cL_{x^*}}{c_G^2}\|G_{x^*}(z^{k,l-1})\|\right)^2 \|G_{x^*}(z^{k,l})\| \\
&\;\;\vdots \\
&\leq \left(\frac{cL_{x^*}}{c_G^2}\|G_{x^*}(z^{k,0})\|\right)^{2^l} \|G_{x^*}(z^{k,l})\|
\end{aligned}$$

$$\leq \quad (\gamma)^{2^l} \|G_{x^*}(z^{k,l})\|$$

$$\leq \quad (\gamma)^l \|G_{x^*}(z^{k,l})\|.$$

Therefore, (6.3.8) holds when k is sufficiently large. Let $\hat{r}^k$ be the distance between $y^{k,0}$ and x^*. When $\hat{r}^k$ is sufficiently small, Theorem 6.2.2 says that a generated sequence $\{z^{k,l}\}_{l=0,1,2,\ldots}$ satisfies $z^{k,l} \in B(x^*, c_3\hat{r}^k)$ for all l, where $c_3 > 0$ is a constant. Moreover, choosing sufficiently large k if necessary, $\hat{r}^k$ can be made arbitrarily small. Then, it follows from $x^*_{P^*} > 0$ and $F_{N^*}(x^*) > 0$ that (6.3.6), (6.3.7) and (6.3.9) hold for sufficiently large k. Consequently, (6.3.6)–(6.3.9) are satisfied for sufficiently large k. Then, from Theorem 6.2.2, $\{\mathrm{dist}(z^{k,l}, x^*_G)\}_{l=0,1,2,\ldots}$ converges to 0 quadratically, i.e., case (b) holds. This completes the proof. □

Note that the set X^*_G does not necessarily coincide with X^*. So (b) does not imply that the sequence generated by Algorithm HYBRID converges to the solution set X^* quadratically. However, it is easy to see that, for sufficiently large k, $x^*_G \cap B(x^*, c_3\hat{r}^k)$ becomes a subset of $x^* \cap B(x^*, c_3\hat{r}^k)$, where $\hat{r}^k := \|y^{k,0} - x^*\|$. Therefore (b) implies that the sequence $\{y^{k,l}\}_{l=0,1,2,\ldots}$ converges to a subset of the solution set X^* quadratically.

Some remarks about Algorithm HYBRID are in order.

- The SNE (6.3.4) has only $|P^k|$ variables actually, since $x_{N^k} = 0$ and $x_{C^k} = 0$. Hence, (6.3.4) becomes smaller than the original problem.

- The condition $\|H_F(x^k)\| \leq M_1$ in Step 2 is not needed to establish Theorem 6.3.1. However, using this condition, we can skip useless calculation which may occur when

$$P(x^*) \neq P^k = P^{k-1},\ N(x^*) \neq N^k = N^{k-1},\ C(x^*) \neq C^k = C^{k-1},$$
$$P^k \cup N^k \cup C^k = \{1, \ldots, n\}.$$

- The calculation in Step 3 may become in vain when the index sets are not identified correctly or an iteration point is far from the solution set. To avoid vain calculations, we may replace the condition in Step 2 by

$$P^k = \cdots = P^{k-j},\ N^k = \cdots = N^{k-j},\ C^k = \cdots = C^{k-j},$$
$$P^k \cup N^k \cup C^k = \{1, \ldots, n\},$$

  where j is any positive integer.

- By Theorem 2.4.2, Algorithm PPA has a genuine superlinear rate of convergence if x^* is nondegenerate. Consequently, when we judge x^* to be nondegenerate in Step 2, we may immediately return to Step 1 instead of proceeding to Step 3.

- The algorithm identifies the index sets $P(x^*), N(x^*)$, and $C(x^*)$ by using the technique proposed in Chapter 5. We may adopt another identification technique such as the one proposed in [19]. The identification technique of [19] is designed to solve LCP rather than NCP. However, it can deal with the nonmonotone LCP, while the technique proposed in Chapter 5 cannot be applied for the nonmonotone LCP. Since Step 3 of Algorithm HYBRID is independent of an identification technique, we can construct another hybrid algorithm by combining the technique of [19] with the ILMM, and show that it has similar convergence properties when applied to LCP.

- For solving constrained nonlinear programs, approaches similar to Algorithm HYBRID have been proposed in [27, 53]. The methods of [27, 53] have superlinear rate of convergence without the strict complementarity condition nor the uniqueness of the Lagrange multiplier. Note that the results require the local uniqueness of a solution of the problem.

6.4 Concluding remarks

In this chapter, we have constructed an SNE which is useful in dealing with degenerate NCPs, and proposed an algorithm based on it. We have shown that the proposed algorithm has a quadratic or superlinear rate of convergence even if the NCP has a degenerate solution.

Finally, we mention some future research topics.

(i) In order to guarantee that Algorithm HYBRID has a quadratic or superlinear rate of convergence, we need Assumption 6.3.2. In Lemma 6.3.1, we have given a sufficient condition for Assumption 6.3.2 to hold, but this condition is imposed on the solution set itself. It is an interesting and important subject to find a milder and/or simpler sufficient condition for Assumption 6.3.2.

(ii) We have used the technique proposed in Chapter 5 to identify P^*, N^* and C^*. However, it is difficult in practice to confirm that $P^k = P^*, N^k = N^*$ and $C^k = C^*$ are attained. If we have a criterion to make sure that the correct identification of the index sets has been accomplished, we may design an algorithm that avoids useless calculations in Step 3 of Algorithm HYBRID.

Chapter 7

Conclusion

In this thesis, we have proposed algorithms for the optimization problem, the system of equations and the complementarity problem. These algorithms can be applied to nonlinear problems, and they are particularly suitable for large-scale problems. The features of the proposed algorithms are summarized as follows:

- In Chapter 3, we have proposed an SQP-based algorithm for NLP. In this algorithm, we solve two types of subproblems. One subproblem is a convex QP with equality and inequality constraints. This subproblem is easy to handle, because it can be solved by many traditional methods. The other is a QP with equality constraints, which is not necessarily convex. The KKT condition for this subproblem is the system of linear equations, so we can solve it easily. We can expect that these subproblems have the sparse structure, so this algorithm is suitable for large-scale NLPs.

- In Chapter 4, we have proposed the ILMM for SNE. This method has a superlinear rate of convergence under a local error bound condition. This condition is milder than conditions which are assumed in the previous algorithms to establish a superlinear or quadratic rate of convergence. Moreover, this method allows inexact solutions for subproblems. This method is suitable for large-scale problems, since it is difficult to find an exact solution of subproblems.

- In Chapters 5 and 6, we have proposed two algorithms for the monotone NCP.

  The algorithm proposed in Chapter 5 aims at identifying degenerate indices when a convergent sequence is generated by Algorithm PPA. The local uniqueness of a solution is assumed in previous identification methods, but in our algorithm, we can identify degenerate indices without such assumptions.

  In Chapter 6, we have proposed a hybrid algorithm for the monotone NCP, which combines the identification method in Chapter 5 and the ILMM in Chapter 4. This method converges superlinearly under milder conditions than those for any existing method. Moreover, this method can efficiently deal with degenerate indices, which may easily occur in large-scale NCPs.

Next we explain some future works concerning problems which are dealt with in this thesis.

- When we solve nonlinear optimization and related problems, we often use Newton's or Newton's-like method. However, we may fail to achieve fast convergence without the nonsingularity of the Jacobian at a solution. In Chapters 4, 5 and 6 of this thesis, we have analyzed convergence properties under a local error bound condition which is milder than the nonsingularity of the Jacobian, and we have shown that our algorithms can achieve fast convergence. It is a challenging subject to construct algorithms which achieve fast convergence under even weaker conditions than a local error bound condition.

- For large-scale problems, we have proposed some algorithms in Chapters 3, 4 and 6, which exploit sparse structure and allow inexact solutions of subproblems. We may utilize advances of hardware more efficiently to deal with huge problems. One of hopeful approaches for this goal is to perform parallel computation in solving subproblems. It is certainly worthwhile to carry out intensive research on this approach.

Appendix A

Complete Proof of Lemma 4.2.3

First we consider the case $k = 0$. Since $e < \bar{b} \leq \frac{b_1}{2}$, we have $x^0 \in B(x^*, \frac{b_1}{2})$. It then follows from Lemma 4.2.1 that

$$\begin{aligned} \|x^1 - x^*\| &= \|x^0 + d^0 - x^*\| \leq \|x^* - x^0\| + \|d^0\| \\ &\leq \|x^* - x^0\| + c_3 \operatorname{dist}(x^0, X^*) + \frac{\|r^0\|}{\mu_0}. \end{aligned}$$

Since

$$\operatorname{dist}(x^0, X^*) \leq \|x^* - x^0\| \leq e \leq b_2,$$

we have from Lemma 4.2.15

$$\begin{aligned} \|x^1 - x^*\| &\leq \|x^* - x^0\| + c_3 \operatorname{dist}(x^0, X^*) + \operatorname{dist}(x^0, X^*) \\ &\leq \|x^* - x^0\| + c_3\|x^* - x^0\| + \|x^* - x^0\| \\ &\leq (2 + c_3)e \leq \frac{2 + c_3}{3 + 2c_3}\, \bar{b} \;\leq\; \bar{b}. \end{aligned}$$

Next we consider the case $k \geq 1$. Suppose that $x^l \in B(x^*, \bar{b}),\ l = 1, \ldots, k$. Since $e < \bar{b}$, we have $x^0 \in B(x^*, \bar{b})$. It follows from $\operatorname{dist}(x^l, X^*) \leq \|x^* - x^l\| \leq \bar{b} \leq b_3$ and Lemma 4.2.2 that

$$\begin{aligned} \operatorname{dist}(x^l, X^*) &\leq \frac{1}{2} \operatorname{dist}(x^{l-1}, X^*) \leq \cdots \leq \left(\frac{1}{2}\right)^l \operatorname{dist}(x^0, X^*) \leq \left(\frac{1}{2}\right)^l \|x^* - x^0\| \\ &\leq \left(\frac{1}{2}\right)^l e \qquad\qquad 0 \leq \forall l \leq k. \end{aligned}$$

Since $x^l \in B\left(x^*, \frac{b_1}{2}\right) \cap B(x^*, b_2),\ l = 0, \ldots, k$, Lemmas 4.2.1 and 4.2.15 yield

$$\|d^l\| \leq (c_3 + 1)\operatorname{dist}(x^l, X^*) \leq (c_3 + 1) \left(\frac{1}{2}\right)^l e. \tag{A.1}$$

Consequently, we obtain

$$\begin{aligned} \|x^{k+1} - x^*\| &\leq \|x^0 - x^*\| + \sum_{l=0}^{k} \|d^l\| \\ &\leq \{1 + 2(c_3 + 1)\}e = \bar{b}. \end{aligned}$$

This completes the proof. □

Appendix B

Complete Proof of Theorem 4.2.1

We obtain the first half of Theorem 4.2.1 from Lemmas 4.2.2 and 4.2.3 directly. So, we only show that $\{x^k\}$ converges to $\hat{x} \in B(x^*, \bar{b})$. For this purpose, we only have to show that $\{x^k\}$ is a convergent sequence because $\{x^k\} \subset B(x^*, \bar{b})$ and $\{\mathrm{dist}(x^k, X^*)\}$ converges to 0.

Note that we have

$$\|d^l\| \leq (c_3 + 1) \left(\frac{1}{2}\right)^l e \quad \forall l \geq 0$$

from (A.1) in the proof of Lemma 4.2.3. Therefore, for all integers $p > q \geq 0$, we obtain

$$\begin{aligned} \|x^p - x^q\| &\leq \sum_{l=q}^{p-1} \|d^l\| \leq (c_3 + 1)e \sum_{l=q}^{p-1} \left(\frac{1}{2}\right)^l \leq (c_3 + 1)e \sum_{l=q}^{\infty} \left(\frac{1}{2}\right)^l \\ &\leq (c_3 + 1)e \left(\frac{1}{2}\right)^{q-1}. \end{aligned}$$

This means that $\{x^k\}$ is a Cauchy sequence, and hence it converges. □

Appendix C

Proof of Sufficiency of Conditions (c) and (d) for Assumption 5.2.1 (ii)

First we show the sufficiency of condition (c). Let $x^* = (y^*, z^*)$ be a solution of NCP(F). Since G is strongly monotone, NCP(G) has the unique solution $\bar{y}$. Then y^* must coincide with $\bar{y}$. Hence z^* is a solution of the linear complementarity problem

$$M_{\bar{y}}(z) \geq 0, \quad z \geq 0, \quad z^T M_{\bar{y}}(z) = 0, \tag{C.1}$$

where $M_y(z) := Uy + Vz$. In view of the fact that conditions (a) and (b) both ensure the error bound condition (2.3.1), there exist positive constants b_1, τ_1 and τ_2 such that for all $(y, z) \in B(x^*, b_1)$

$$\|y - \bar{y}\| \leq \tau_1 \|H_G(y)\| \tag{C.2}$$

and

$$\text{dist}\{z, Z^*\} \leq \tau_2 \|H_{M_{\bar{y}}}(z)\|,$$

where Z^* is the solution set of LCP (C.1). Therefore, we have for any $x = (y, z) \in B(x^*, b_1)$

$$\text{dist}\{x, X\} \;\leq\; \|y - \bar{y}\| + \text{dist}\{z, Z^*\} \;\leq\; \tau_1 \|H_G(y)\| + \tau_2 \|H_{M_{\bar{y}}}(z)\|. \tag{C.3}$$

Since ϕ satisfying (2.2.1) is Lipschitz continuous, there exist a constant L such that for all $(y, z) \in B(x^*, b_1)$

$$\|H_{M_{\bar{y}}}(z) - H_{M_y}(z)\| \leq L \|y - \bar{y}\|$$

It then follows from (C.3) that for any $(y, z) \in B(x^*, b_1)$

$$\begin{aligned} \text{dist}\{x, X\} &\leq \tau_1 \|H_G(y)\| + \tau_2 \|H_{M_y}(z)\| + \tau_2 L \|y - \bar{y}\| \\ &\leq \tau_1 (1 + \tau_2 L) \|H_G(y)\| + \tau_2 \|H_{M_y}(z)\| \\ &\leq \max\{\tau_1 (1 + \tau_2 L), \tau_2\} \|H_F(x)\|, \end{aligned}$$

where the second inequality follows from (C.2).

Next we show the sufficiency of (d). Let y^* be a solution of (5.2.4) where the second order sufficient conditions and the Mangasarian-Fromovitz constraint qualification holds. Note that y^* is a locally unique solution. Let $\Lambda(y^*)$ be the set of Lagrange multipliers associated with y^*, i.e.,

$$\Lambda(y^*) := \{(\lambda, \mu) \mid \nabla f(y^*) + \nabla g(y^*)\lambda - \mu = 0,\ \lambda \geq 0,\ \mu \geq 0,\ \lambda^T g(y^*) = 0,\ \mu^T y^* = 0\}$$

and let $(\lambda^*, \mu^*) \in \Lambda(y^*)$. Then, from Theorem 3.6 in [18], there exist b_1 and $\tau > 0$ such that, for all $(y, \lambda, \mu) \in B((y^*, \lambda^*, \mu^*), b_1)$

$$\mathrm{dist}((y, \lambda, \mu), \{y^*\} \times \Lambda(y^*)) \;\leq\; \tau \left\| \begin{array}{c} \nabla f(y) + \nabla g(y)\lambda - \mu \\ \min\{\mu, y\} \\ \min\{-g(y), \lambda\} \end{array} \right\|. \tag{C.4}$$

Since ∇f and ∇g are continuous, there exist a positive constant $\hat{b}_1 < b_1$ such that $(y, \lambda, \nabla f(y) + \nabla g(y)\lambda) \in B((y^*, \lambda^*, \mu^*), b_1)$ for all $(y, \lambda) \in B((y^*, \lambda^*), \hat{b}_1)$. Substituting $\mu = \nabla f(y) + \nabla g(y)\lambda$ into (C.4), we have for all $x = (y, \lambda) \in B((y^*, \lambda^*), \hat{b}_1)$

$$\begin{array}{rcl} \mathrm{dist}\{x, X\} & = & \mathrm{dist}\{(y, \lambda), \{y^*\} \times \{\lambda \mid (\lambda, \mu) \in \Lambda(y^*)\}\} \\ & \leq & \mathrm{dist}((y, \lambda, \nabla f(y) + \nabla g(y)\lambda), \{y^*\} \times \Lambda(y^*)) \\ & \leq & \tau \left\| \begin{array}{c} \min\{\nabla f(y) + \nabla g(y)\lambda, y\} \\ \min\{-g(y), \lambda\} \end{array} \right\| \\ & = & \tau \| \min\{x, F(x)\} \|. \end{array}$$

It then follows from the definition of H_F that $\|H_F(x)\|$ provides a local error bound for NCP(F).

□

Bibliography

[1] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific (1995).

[2] S. C. Billups, Improving the Robustness of Descent-Based Methods for Semismooth Equations Using Proximal Perturbations, *Mathematical Programming* 87 (2000), 269–284.

[3] P. T. Boggs and J. W. Tolle, Sequential Quadratic Programming. In *Acta Numerica*, Cambridge University Press (1995), 1-51.

[4] I. Bongartz, A. R. Conn, N. I. M. Gould and Ph. L. Toint, CUTE: Constrained and Unconstrained Testing Environment, *ACM Transactions on Mathematical Software* 21 (1995), 123-160.

[5] R. H. Byrd, R. B. Schnabel and G. A. Schultz, A Trust Region Algorithm for Nonlinearly Constrained Optimization, *SIAM Journal on Numerical Analysis* 24 (1987), 1152-1170.

[6] M. R. Celis, J. E. Dennis, Jr. and R. A. Tapia, A Trust Region Strategy for Nonlinear Equality Constrained Optimization. In P. T. Boggs, R. H. Byrd and R. B. Schnabel (eds.), *Numerical Optimization 1984*, SIAM (1985), 71-82.

[7] C. Chen and O. L. Mangasarian, Smoothing Methods for Convex Inequalities and Linear Complementarity Problems, *Mathematical Programming* 71 (1995), 51-69.

[8] C. Chen and O. L. Mangasarian, A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problems, *Computational Optimization and Applications* 5 (1996), 97-138.

[9] X. Chen, Smoothing Methods for Complementarity Problems and Their Applications: A Survey, *Journal of the Operations Research Society of Japan* 43 (2000), 32-47.

[10] A. R. Conn, N. I. M. Gould and P. L. Toint, *Trust-Region Methods*, SIAM (2000).

[11] R. W. Cottle, J.-S. Pang and R. E. Stone, *The Linear Complementarity Problem*, Academic Press (1992).

[12] T. De Luca, F. Facchinei and C. Kanzow, A Semismooth Equation Approach to the Solution of Nonlinear Complementarity Problems, *Mathematical Programming* 75 (1996), 407-439.

[13] J. E. Dennis, Jr. and J. J. Moré, A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods, *Mathematics of Computation* 28 (1974), 549-560.

[14] H. Dan, N. Yamashita and M. Fukushima, Convergence Properties of the Inexact Levenberg-Marquardt Method under Local Error Bound Conditions, *Optimization Methods and Software* 17 (2002), 605-626.

[15] H. Dan, N. Yamashita and M. Fukushima, A Superlinearly Convergent Algorithm for the Monotone Nonlinear Complementarity Problem without Uniqueness and Nondegeneracy Conditions, *Mathematics of Operations Research* 27 (2002), 743-753.

[16] A. S. El-Bakry, R. A. Tapia and Y. Zhang, A Study of Indicators for Identifying Zero Variables in Interior-Point Methods, *SIAM Review* 36 (1994), 45-72.

[17] A. S. El-Bakry, R. A. Tapia and Y. Zhang, On the Convergence Rate of Newton Interior-Point Methods in the Absence of Strict Complementarity, *Computational Optimization and Applications* 6 (1996), 157-167.

[18] F. Facchinei, A. Fischer and C. Kanzow, On the Accurate Identification of Active Constraints, *SIAM Journal on Optimization* 9 (1998), 14-32.

[19] F. Facchinei, A. Fischer and C. Kanzow, On the Identification of Zero Variables in an Interior-point Framework, *SIAM Journal on Optimization* 10 (2000), 1058-1078.

[20] F. Facchinei and C. Kanzow, A Nonsmooth Inexact Newton Method for the Solution of Large-Scale Nonlinear Complementarity Problems, *Mathematical Programming* 76 (1997), 493-512.

[21] F. Facchinei and C. Kanzow, Beyond Monotonicity in Regularization Methods for Complementarity Problems, *SIAM Journal on Control and Optimization* 37 (1999), 1150-1161.

[22] F. Facchinei and J.-S. Pang, *Finete-Dimensional Variational Inequalities and Complementarity Problems*, Springer-Verlag (2003).

[23] F. Facchinei and J. Soares, A New Merit Function for Nonlinear Complementarity Problems and a Related Algorithm, *SIAM Journal on Optimization* 7 (1997), 225-247.

[24] M.C. Ferris and J.-S. Pang, Engineering and Economic Applications of Complementarity Problem, *SIAM Review* 39 (1997), 669-713.

[25] A. Fischer, A Special Newton-Type Optimization Method, *Optimization* 24 (1992), 269–284.

[26] A. Fischer, An NCP-Function and Its Use for the Solution of Complementarity Problems. In D. Z. Du, L. Qi, R. S. Womersley (eds.), *Recent Advances in Nonsmooth Optimization*, World Scientific (1995), 88-105.

[27] A. Fischer, Modified Wilson method for Nonlinear Programs with Nonunique Multipliers, *Mathematics of Operations Research* 24 (1999), 699-727.

[28] R. Fletcher, *Practical Methods of Optimization*, second ed., John Wiley & Sons (1987).

[29] M. Fukushima and L. Qi (eds.), *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, Kluwer Academic Publishers (1999).

[30] P. E. Gill and W. Murray, Numerically Stable Methods for Quadratic Programming, *Mathematical Programming* 14 (1978), 349-372.

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

[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*, second ed., The Johns Hopkins University Press (1989).

[33] N. I. M. Gould, D. Orban and Ph. L. Toint, CUTEr and SifDec: A Constrained and Unconstrained Testing Environment, revisited, *ACM Transactions on Mathematical Software* 29 (2003), 373-394.

[34] M. R. Hestenes, *Conjugate Direction Methods in Optimization*, Springer-Verlag (1980).

[35] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems 187, Springer-Verlag (1981).

[36] H. Jiang, Global Convergence Analysis of the Generalized Newton and Gauss-Newton Methods of the Fischer-Burmeister Equation for the Complementarity Problem, *Mathematics of Operations Research* 24 (1999), 529-543.

[37] M. Kojima and S. Shindo, Extensions of Newton and Quasi-Newton Methods to Systems of PC^1 Equations, *Journal of the Operations Research Society of Japan* 29 (1986), 352-374.

[38] M. Lalee, J. Nocedal and T. Plantenga, On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization, *SIAM Journal on Optimization* 8 (1998), 682-706.

[39] Z. -Q. Luo, O. L. Mangasarian, J. Ren, M. V. Solodov, New error Bounds for the Linear Complementarity Problem, *Mathematics of Operations Research* 19 (1994), 880-892.

[40] E. O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph. D. Thesis, University of Colorado (1989).

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

[42] J. -S. Pang, A Posteriori Error Bounds for the Linearly-Constrained Variational Inequality Problem, *Mathematics of Operations Research* 12 (1987), 474-484.

[43] J. -S. Pang, Error Bounds in Mathematical Programming, *Mathematical Programming* 79 (1997), 299-332.

[44] M. J. D. Powell and Y. Yuan, A Trust Region Algorithm for Equality Constrained Optimization, *Mathematical Programming* 49 (1991), 189-211.

[45] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Mathematics of Operations Research* 18 (1993), 227-244.

[46] S.M. Robinson, Some Continuity Properties of Polyhedral Multifunctions, *Mathematical Programming Study* 14 (1981), 206-214.

[47] R. T. Rockafellar, *Convex Analysis*, Princeton University Press (1970).

[48] R. T. Rockafellar, Monotone Operators and the Proximal Point Algorithm, *SIAM Journal on Control Optimization* 14 (1976), 877-898.

[49] E. Rothberg and A. Gupta, Efficient Sparse Matrix Factorization on High Performance Workstations — Exploiting the Memory Hierarchy, *ACM Transactions on Mathematical Software* 17 (1991), 313-334.

[50] J. Stoer, M. Wechs and S. Mizuno, High Order Infeasible-Interior-Point Methods for Sufficient Linear Complementarity Problems, *Mathematics of Operations Research* 23 (1998), 832-862.

[51] P. Tseng, Growth Behavior of a Class of Merit Functions for the Nonlinear Complementarity Problem, *Journal of Optimization Theory and Applications* 89 (1996), 17-37.

[52] A. Vardi, A Trust Region Algorithm for Equality Constrained Minimization: Convergence Properties and Implementation, *SIAM Journal on Numerical Analysis* 22 (1985), 575-591.

[53] S.J. Wright, Constraint Identification and Algorithm Stabilization for Degenerate Nonlinear Programs, Preprint ANL/MCS-P865-1200, Mathematics and Computer Science Division, Argonne National Laboratory (2000).

[54] H. Yamashita and H. Dan, Global Convergence of a Trust Region Sequential Quadratic Programming Method, to appear in *Journal of the Operations Research Society of Japan.*

[55] H. Yamashita and H. Yabe, A Nonmonotone SQP Method with Global and Superlinear Convergence Properties, Technical report, Mathematical Systems, Inc. (1996, Revised 1998).

[56] H. Yamashita, H. Yabe and T. Tanabe, A Globally and Superlinearly Convergent Primal-Dual Interior Point Trust Region Method for Large Scale Constrained Optimization, to appear in *Mathematical Programming.*

[57] N. Yamashita, H. Dan and M. Fukushima, On the Identification of Degenerate Indices in the Nonlinear Complementarity Problem with the Proximal Point Algorithm, *Mathematical Programming* 99 (2004), 377-397.

[58] N. Yamashita and M. Fukushima, Modified Newton Methods for Solving a Semismooth Reformulation of Monotone Complementarity Problems, *Mathematical Programming* 76 (1997), 469-491.

[59] N. Yamashita and M. Fukushima, The Proximal Point Algorithm with Genuine Superlinear Convergence for the Monotone Complementarity Problem, *SIAM Journal on Optimization* 11 (2001), 364-379.

[60] N. Yamashita and M. Fukushima, On the Rate of Convergence of the Levenberg-Marquardt Method, *Computing [Suppl]* 15 (2001), 227-238.

[61] N. Yamashita, J. Imai and M. Fukushima, The Proximal Point Algorithm for the P_0 Complementarity Problem. In M. C. Ferris, O. L. Mangasarian and J.-S. Pang (eds.), *Complementarity: Applications, Algorithms and Extensions*, Kluwer Academic Publishers (2001), 361-379.

List of Author's works

1. H. Dan, N. Yamashita and M. Fukushima, Convergence Properties of the Inexact Levenberg-Marquardt Method under Local Error Bound Conditions, *Optimization Methods and Software* 17 (2002), 605-626.

2. H. Dan, N. Yamashita and M. Fukushima, A Superlinearly Convergent Algorithm for the Monotone Nonlinear Complementarity Problem without Uniqueness and Nondegeneracy Conditions, *Mathematics of Operations Research* 27 (2002), 743-753.

3. H. Yamashita and H. Dan, Global Convergence of a Trust Region Sequential Quadratic Programming Method, to appear in *Journal of the Operations Research Society of Japan.*

4. N. Yamashita, H. Dan and M. Fukushima, On the Identification of Degenerate Indices in the Nonlinear Complementarity Problem with the Proximal Point Algorithm, *Mathematical Programming* 99 (2004), 377-397.