Contents

1 Introduction

Given a continuously differentiable function F, which is a mapping from $\mathbb{R}^n$ into itself, the nonlinear complementarity problem (NCP) is to find a vector $x \in \mathbb{R}^n$ satisfying the following system of equations and inequalities [3]:

$$x_i \geq 0,\ F_i(x) \geq 0,\ x_i F_i(x) = 0 \qquad \forall i = 1, ..., n. \tag{1}$$

The concept of complementarity is synonymous with the idea of system equilibrium. Hence, complementarity problems play an important role in the formulation of several optimization and equilibrium problems [8] and are so pervasive in engineering, economics, traffic theory and optimization [3, 6].

To this end, much attention has been paid to approaches which employ an equivalent reformulation of the NCP (1) as a nonlinear and nonsmooth system of equations

$$\Phi(x) = 0, \tag{2}$$

for a suitable equation operator $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ [4, 7-10]. The resulting reformulated system may then be treated as an unconstrained optimization problem, and for its solution, techniques which involve the use of classical methods such as smoothing methods, interior-point methods or regularization methods have been proposed. Works of previous authors have established that under appropriate conditions, a stationary point of the reformulated system is a solution to the original complementarity problem [9].

Based on the aforementioned methods, many algorithms with good convergence properties have been developed. Particularly, for problems exhibiting monotonicity, these algorithms usually achieve global convergence. However, for ill-conditioned problems, there are cases wherein the algorithms tend to become computationally unstable and convergence is not guaranteed.

Generally, any method for optimization may be classified as a trust-region or line-search method. Both of these iterative methods define a new iterate by minimizing a quadratic model of the objective function of the optimization problem [13]. Line-search algorithms obtain a search direction in each iteration and searches along this direction to obtain a better point. The search direction is a descent direction, and hence, unless a stationary point is reached, there always exist better points along the search direction. Trust-region algorithms, on the other hand, start with a current iterate of the solution to the optimization problem. The quadratic model is constructed in a “trust region” near the current point and a solution to the approximate model is taken as the next iterate [13]. As opposed to line-search methods, trust-region algorithms can use non-convex approximate models. This advantage is due to the boundedness of the trust region. Trust-region algorithms are robust and can be applied to ill-conditioned problems. Moreover, they have very strong convergence properties.

A relatively recent concept is the integration of the nonmonotone technique in the development of algorithms. Most trust region algorithms proposed are descent methods, in that they only accept the trial point as the next iterate if its associated objective function value is strictly lower than that of the current iterate. In nonmonotone methods, some growth in the function value is permitted. Nonmonotone schemes can improve the likelihood of finding a global optimum and is helpful in overcoming the case wherein the sequence of iterates is to follow the bottom of curved narrow valleys, which is a common occurrence in difficult nonlinear problems. Many authors have given numerous nonmonotone algorithms based on trust-region or line-search and encouraging numerical results have been reported when nonmonotone schemes were applied especially to ill-conditioned problems.

This research is therefore focused on the development of an algorithm for the solution of the nonlinear complementarity problem which makes use of the trust-region approach. Another aim is to achieve computational stability by adding redundant equations to the nonlinear system of equations obtained by reformulating the nonlinear complementarity problem. In addition, to further improve the performance of the algorithm, the nonmonotone technique is utilized.

The organization of this paper is as follows: In section 2, we describe our approach for solving the nonlinear complementarity problem in more detail and we present our proposed Levenberg-Marquardt-type method. The global convergence of our algorithm is investigated in Section 3 and in Section 4, we finally present our nonmonotone algorithm. Promising numerical results are summarized in Section 5. We conclude the paper in Section 6 with some remarks and recommendations for further improvement of the algorithm.

2 Preliminaries

2.1 Reformulation Approach

A common method for reformulating the NCP (1) makes use of the well-known Fischer-Burmeister (FB) function defined as follows:

$$\phi_{FB}(a,b) := \sqrt{a^2+b^2} - a - b, \qquad a,\, b \in \mathbb{R}.$$

The FB function belongs to the class of NCP functions ϕ having the property

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

Therefore, it is easy to see that the NCP may be transformed into an equivalent system of nonlinear equations $\Phi_{FB}(x) = 0$ with $\Phi_{FB} : \mathbb{R}^n \to \mathbb{R}^n$ defined as

$$\Phi_{FB} := \begin{pmatrix} \phi_{FB}(x_1, F_1(x)) \\ \vdots \\ \phi_{FB}(x_n, F_n(x)) \end{pmatrix}.$$

This reformulation has some stability problems due to the usage of the Fischer-Burmeister function in the transformation process [9]. Note that in the case where $F_i(x) >> x_i > 0$, the value of $\phi_{FB}(F_i(x), x_i)$ is close to zero while the product $x_i F_i(x)$ is not. Hence, although we are close to obtaining a zero of $\phi_{FB}(F_i(x), x_i)$, we are barely successful in reducing the value of the product $x_i F_i(x)$.

To remedy this problem, we introduce the following function:

$$\phi_+(a,b) := a_+ b_+,$$

where $z_+ := \max\{0, z\}$ for all $z \in \mathbb{R}^n$.

As in [9], with the use of the $\phi_+(a,b)$ function, we form the following mapping:

$$\Phi(x) := \begin{pmatrix} \vdots & \\ \lambda \phi_{FB}(x_i, F_i(x)), & i = 1, ..., n \\ \vdots & \\ (1-\lambda)\phi_+(x_i, F_i(x)), & i = 1, ..., n \\ \vdots & \end{pmatrix}, \tag{3}$$

where $\lambda \in (0,1)$ is a fixed but arbitrary parameter. Here, λ acts as a weight between the original set of equations obtained with the FB function and the additional set of redundant equations.

Note that $\Phi(x) = 0$ is an overdetermined system of equations having the property

$$\begin{gathered} x^* \text{ solves } \Phi(x) = 0 \\ \Updownarrow \\ x^* \text{ solves the complementarity problem.} \end{gathered}$$

We have therefore obtained a nonlinear least squares formulation of the nonlinear complementarity problem and for the solution of this problem, we propose a Levenberg-Marquardt-type method.

Our algorithm is based on the reduction of the natural merit function $\Psi : \mathbb{R}^n \to \mathbb{R}$ given by

$$\Psi(x) = \frac{1}{2} \|\Phi(x)\|^2 .$$

From [9], we state the following theorem, which will be helpful in the convergence analysis of our algorithm.

Theorem 2.1 (Kanzow [9]) *The merit function $\Psi(x)$ has the following properties:*

(a) Ψ *is continuously differentiable with $\nabla\Psi(x) = H_k'\Phi(x)$, where $H \in \partial_C\Phi(x)$ can be chosen arbitrarily.*

(b) *If x^* is a stationary point of Ψ and $F'(x^*)$ is a P_0-matrix, then x^* is a solution to the complementarity problem.*

In the theorem, $\partial_C\Phi(x)$ denotes the *C-subdifferential* of Φ at x, which is similarly defined as in [10]. For the definition of a P_0-matrix, the reader is referred to [1].

2.2 The Levenberg-Marquardt Method

The classical Levenberg-Marquardt method (see [11], [12]) for the nonlinear system $\Phi(x) = 0$ obtains the current trial step d^k by solving the following system

$$d = -(H_k'H_k + \mu_k I)^{-1}\nabla\Psi(x^k) \tag{4}$$

where H_k belongs to $\partial_C\Phi(x^k)$, and $\mu_k \geq 0$ is a parameter being updated from iteration to iteration.

The Levenberg-Marquardt step (4) is a modification of the Newton step and the parameter μ_k, which is introduced to overcome the difficulties caused by the singularity or near singularity of H_k, is determined in every iteration by the ratio between the predicted and actual reductions of the penalty function.

The difference between $\|\Phi(x^k)\|^2$ and $\|\Phi(x^k) + H_k d^k\|^2$ can be treated as a predicted reduction of the penalty function $\Psi(x)$, and we denote it by $Pred_k$.

$$Pred_k = \|\Phi(x^k)\|^2 - \|\Phi(x^k) + H_k d^k\|^2. \tag{5}$$

The actual reduction $Ared_k$ of the penalty function is

$$Ared_k = \Psi(x^k) - \Psi(x^k + d^k). \tag{6}$$

Central to the idea of trust-region methods is the ratio between these two reductions

$$r_k = \frac{Ared_k}{Pred_k} = \frac{\Psi(x^k) - \Psi(x^k + d^k)}{\|\Phi(x^k)\|^2 - \|\Phi(x^k) + H_k d^k\|^2}. \tag{7}$$

We choose the Levenberg-Marquardt parameter μ_k in (4) as

$$\mu_k = \alpha_k \|\Phi(x^k)\|, \tag{8}$$

where α_k is updated in every iteration depending on the value of r_k.

Now, we are ready to formally state our Levenberg-Marquardt-type algorithm for finding the stationary points of the merit function $\Psi(x)$. First, we present the monotone method [5], where we base the construction of our nonmonotone algorithm.

Algoritm 2.1 (Monotone Levenberg-Marquardt Method)

Step 0. *Initialization.* Choose starting point $x^0 \in \mathbb{R}^n$, and parameters $\epsilon \geq 0$, $\lambda \in (0,1)$, $0 \leq \rho_0 \leq \rho_1 \leq \rho_2 < 1$, δ_1, δ_2, $m > 0$. Set $k = 0$.

Step 1. *Convergence Test.* If $\|\nabla\Psi(x^k)\| \leq \epsilon$: STOP.

Step 2. Find a solution $d^k \in \mathbb{R}^n$ of

$$d = -(H_k' H_k + \alpha_k \|\Phi(x^k)\| I)^{-1} \nabla\Psi(x^k) \tag{9}$$

where $H_k \in \partial_C\Phi(x^k)$.

Step 3. Compute $r_k = Ared_k/Pred_k$; set

$$x^{k+1} = \begin{cases} x^k + d^k, & \text{if } r_k > \rho_0, \\ x^k, & \text{otherwise,} \end{cases}$$

Step 4. Choose α_{k+1} as

$$\alpha_{k+1} = \begin{cases} \delta_1 \alpha_k, & \text{if } r_k < \rho_1 \\ \alpha_k, & \text{if } r_k \in [\rho_1, \rho_2], \\ \max\{\delta_2 \alpha_k,\, m\}, & \text{if } r_k > \rho_2, \end{cases} \tag{10}$$

Step 5. $k \leftarrow k+1$; go to Step 1.

3 Global Convergence for the Nondifferentiable Case

Algorithm 2.1 is similar to the one by Fan in [5]. The global convergence of the algorithm was proven with the following assumption:

Assumption 3.1 $\Phi(x)$ *is continuously differentiable and both* $\Phi(x)$ *and its Jacobi* $\nabla\Phi(x)$ *are Lipschitz continuous, i.e., there exist positive constants* L_1 *and* L_2 *such that*

$$\|\nabla\Phi(y) - \nabla\Phi(x)\| \le L_1 \|y - x\|, \qquad \forall x, y,$$

and

$$\|\Phi(y) - \Phi(x)\| \le L_2 \|y - x\|, \qquad \forall x, y.$$

This assumption, however, is restrictive especially for our problem, since $\Phi(x)$ in our case is nondifferentiable. For the convergence analysis of Algorithm 2.1, we therefore need the following new assumptions.

Assumption 3.2 *Let* $\{x^k\}$ *be the sequence generated by Algorithm 2.1.* $\{x^k\}$ *is bounded and* F *is continuously differentiable.*

The following lemmas are essential to our convergence analysis.

Lemma 3.1 *Suppose that Assumption 3.2 holds. Then* $\{H_k' H_k\}, \{\Phi(x^k)\}$ *and* $\{H_k' \Phi(x^k)\}$ *are bounded.*

Proof. The statements directly follow from the definitions of H_k and $\Phi(x^k)$ and the continuity of ∇F.

□

Lemma 3.2 $Pred_k = -(H_k'\Phi(x^k))'d^k + \alpha_k\|\Phi(x^k)\|\|d^k\|^2$.

Proof. From the definition of $Pred_k$ (5), we have

$$\begin{aligned} Pred_k &= \|\Phi(x^k)\|^2 - \|\Phi(x^k) + H_k d^k\|^2 \\ &= -2(H_k'\Phi(x^k))'d^k - d^k H_k' H_k d^k. \end{aligned} \tag{11}$$

Multiplying both sides of (9) by d^k and rearranging terms, we get

$$d^k H_k' H_k d^k = -(H_k'\Phi(x^k))'d^k - \alpha_k\|\Phi(x^k)\|\|d^k\|^2. \tag{12}$$

By combining (11) and (12), we get the desired result.

□

Lemma 3.3 *Suppose that Assumption 3.2 holds. Then, if* $\|H_k'\Phi(x^k)\| \geq \tau$ *and* $\|\Phi(x^k)\| \geq \tau'$ *for some* τ, $\tau' > 0$, *then* $\alpha_k \to \infty$.

Proof. Define the set of indices $T = \{k \mid r_k \geq \rho_0\}$. If $|\,T\,|$ is finite, then $\alpha_k \to \infty$ from Step 4 of Algorithm 2.1. Suppose that $|\,T\,|$ is infinite. Then by the definition of r_k and Lemma 3.2,

$$\begin{aligned} \|\Phi(x^0)\| &\geq \sum_{k\in T} \left\|\Phi(x^k)\right\|^2 - \left\|\Phi(x^{k+1})\right\|^2 \\ &\geq \rho_0 \sum_{k\in T} Pred_k \\ &= \rho_0 \sum_{k\in T} \{-(H_k'\Phi(x^k))'d^k + \alpha_k\|\Phi(x^k)\|\|d^k\|^2\}. \end{aligned}$$

Since $-(H_k'\Phi(x^k))'d^k \geq 0$,

$$\sum_{k\in T} \alpha_k\|\Phi(x^k)\|\|d^k\|^2 < \infty.$$

From the assumption that $\|\Phi(x^k)\| > \tau'$ and $\alpha_k > m$ from (10), we have

$$\sum_{k\in T} \|d^k\|^2 < \infty,$$

which implies that

$$\lim_{k\in T} \|d^k\| = 0. \tag{13}$$

Since $(\|H_k' H_k + \alpha_k \|\Phi(x^k)\| I) d^k = \|H_k' \Phi(x^k)\| \geq \tau$, by (13) and Lemma (3.1), we have $\lim_{k\to\infty;\ k\in T} \alpha_k \to \infty$. Since $\alpha_{k+1} \geq \alpha_k$ for $k \notin T$, we have $\alpha_k \to \infty$.

□

Now, we will state our main convergence result. Recall that from Theorem 2.1, $H_k' \Phi(x^k) = 0$ implies that x^k is a solution to the NCP under appropriate conditions.

Theorem 3.1 *Under the conditions of Assumption 3.2, the sequence generated by Algorithm 3.1 satisfies*

$$\liminf_{k\to\infty} \|H_k' \Phi(x^k)\| = 0 \qquad \textit{or} \qquad \liminf_{k\to\infty} \|\Phi(x^k)\| = 0.$$

Proof. If the theorem is not true, then there exists $\tau > 0$ such that

$$\|H_k' \Phi(x^k)\| \geq \tau \quad \text{ and } \quad \|\Phi(x^k)\| \geq \tau \tag{14}$$

for all k. From Lemma 3.3, we have $\alpha_k \to \infty$. Define the set

$$S = \{k \mid r_k < \rho_1\}.$$

Then, $|\,S\,|$ is infinite. For $k \in S$,

$$\rho_1 \geq r_k = \frac{Ared_k}{Pred_k}.$$

From Lemma 3.2 and the definition of $Ared_k$ (6), we have

$$\begin{aligned} &\rho_1(-(H_k'\Phi(x^k))' d^k + \alpha_k \|\Phi(x^k)\| \|d^k\|^2) \\ &\qquad \geq (\|\Phi(x^k)\|^2 - \|\Phi(x^k + d^k)\|^2. \end{aligned} \tag{15}$$

Define

$$s^k = \frac{d^k}{\|d^k\|} \quad \text{ and } \quad t^k = \|d^k\|. \tag{16}$$

Then (15) becomes

$$\begin{aligned} &\rho_1(-t_k (H_k'\Phi(x^k))' s^k + \alpha_k t_k^2 \|\Phi(x^k)\|) \\ &\qquad \geq \|\Phi(x^k)\|^2 - \|\Phi(x^k + t_k s_k)\|^2. \end{aligned}$$

Dividing both sides by t_k, we obtain

$$\begin{aligned}
&\rho_1(-(H_k'\Phi(x^k))'s^k + \alpha_k t_k\|\Phi(x^k)\|) \\
&\qquad \geq \frac{\|\Phi(x^k)\|^2 - \|\Phi(x^k + t_k s_k)\|^2}{t_k} \\
&\qquad \geq \nabla\Phi(x^k)'s_k + o(t_k) \\
&\qquad \geq -2(H_k'\Phi(x^k))'s_k + o(t_k).
\end{aligned}$$

Rearranging the terms, we get

$$\rho_1\alpha_k t_k\|\Phi(x^k)\| \geq (2-\rho_1)(-H_k'\Phi(x^k)'s_k) + o(t_k). \tag{17}$$

Meanwhile, multiplying both sides of (9) by d^k, together with (16) and the fact that $(d^k)'H_k'H_k d^k \geq 0$, we get

$$\begin{aligned}
-(H_k'\Phi(x^k))'d^k &= (d^k)'(H_k'H_k + \alpha_k\|\Phi(x^k)\|I)d^k \\
&\geq \alpha_k\|\Phi(x^k)\|\|d^k\|^2 \\
&\geq \alpha_k t_k^2\|\Phi(x^k)\| \\
-(H_k\Phi(x^k))'s_k &\geq \alpha_k t_k\|\Phi(x^k)\|.
\end{aligned}$$

It follows from (17) that

$$\begin{aligned}
\rho_1(\alpha_k t_k\|\Phi(x^k)\|) &\geq (2-\rho_1)(\alpha_k t_k\|\Phi(x^k)\|) + o(t_k) \\
\rho_1 &\geq 2-\rho_1 + \frac{o(t_k)}{\alpha_k t_k\|\Phi(x^k)\|} \qquad (18)
\end{aligned}$$

From (16),

$$\begin{aligned}
t_k = \|d_k\| &= \left\|(H_k'H_k + \alpha_k\|\Phi(x^k)\|I)^{-1}H_k'\Phi(x^k)\right\| \\
&\leq \left\|(H_k'H_k + \alpha_k\|\Phi(x^k)\|I)^{-1}\right\| \left\|H_k'\Phi(x^k)\right\| \\
&\leq \frac{1}{\alpha_k\|\Phi(x^k)\|}\|H_k'\Phi(x^k)\|.
\end{aligned}$$

It then follows from Lemma 3.1 and $\alpha_k \to \infty$, that $t_k \to 0$, which implies

$$\frac{o(t_k)}{\alpha_k t_k\|\Phi(x^k)\|} \to 0.$$

From this fact and (18), we have $\rho_1 \geq 2 - \rho_1$, which contradicts $\rho_1 < 1$. Therefore, (14) cannot be true and the proof is completed.

□

4 Nonmonotone Levenberg-Marquardt-type Algorithm

In this section, we present our nonmonotone algorithm. Monotone algorithms usually fail on ill-posed problems and many authors have succeeded in improving their algorithms by using the nonmonotone technique.

In the outer (nonmonotone) iteration of the algorithm, the trial step d_k is always accepted. The inner (monotone) step is basically similar to Algorithm 2.1, except that instead of r_k, we compute r_p at the p^{th} monotone iteration by

$$r_p = \begin{cases} -\infty, & \text{if } \|\Phi(y^p)\|^2 - \|\Phi(x^k) + H_k d^k\|^2 \leq 0 \\ \frac{\|\Phi(y^p)\|^2 - \|\Phi(x^k+d^k)\|^2}{\|\Phi(y^p)\|^2 - \|\Phi(x^k)+H_k d^k\|^2}, & \text{otherwise,} \end{cases} \tag{19}$$

where y^p is the iterate updated only during the monotone step.

Algoritm 4.1 (Nonmonotone Levenberg-Marquardt Method)

Step 0. *Initialization.* Choose starting point $x^0 \in \mathbb{R}^n$, and parameters $\epsilon \geq 0$, $\lambda \in (0,1)$, $0 \leq \rho_0 \leq \rho_1 \leq \rho_2 < 1$, δ_1, δ_2, $m > 0$, $N \geq 0$. Set $k = 0$, $\ell = 0$, $y^0 = x^0$.

Step 1. *Convergence Test.* If $\|\nabla\Psi(x^k)\| \leq \epsilon$: STOP.

Step 2. Find a solution $d^k \in \mathbb{R}^n$ of

$$d = -(H_k' H_k + \alpha_k \|\Phi(x^k)\| I)^{-1} \nabla\Psi(x^k)$$

where $H_k \in \partial_C \Phi(x^k)$.

Step 3. If $\ell = N$, perform the monotone step:

Step 3.0 Compute r_p by (19).

Step 3.1 If $r_p > \rho_1$, set $x^{k+1} = x^k + d^k$ and $y^{p+1} = x^{k+1}$; otherwise $x^{k+1} := y^p$.

Step 3.2 Choose α_{k+1} as

$$\alpha_{k+1} = \begin{cases} \delta_1 \alpha_k, & \text{if } r_p < \rho_1 \\ \alpha_k, & \text{if } r_p \in [\rho_1, \rho_2], \\ \max\{\delta_2 \alpha_k, m\}, & \text{if } r_p > \rho_2, \end{cases} \tag{20}$$

Step 3.3 $p \leftarrow p + 1$, set $\ell = 0$; go to Step 3.

Otherwise $x^{k+1} := x^k + d^k$.

Step 4. $k \leftarrow k+1$, $\ell \leftarrow \ell+1$; go to Step 1.

5 Numerical Results

We implemented Algorithm 4.1 on MATLAB 6.5 and here, we present the results of our numerical experiments using all the nonlinear complementarity problems in the MCPLIB test problem collection [2].

In all of the experiments, we set $\epsilon = 10^{-6}$, $\rho_0 = 0.1$, $\rho_1 = 0.1$, $\rho_2 = 0.5$, $\delta_1 = 10$, $\delta_2 = 0.5$, and $m = 10^{-8}$. We terminate the algorithm if either of the following conditions is satisfied:

$$\|\nabla\Psi(x^k)\| \leq 10^{-6} \quad \text{or} \quad k > 300.$$

Results of the preliminary experiments are shown in Tables 1 and 2. For the first experiment (see Table 1), we arbitrarily fix the nonmonotone index $N = 10$, i.e., we let the algorithm perform the monotone step once in every 10 iterations, and test the method for different values of λ. Results show that the algorithm performs best with $\lambda = 0.9$. In particular, the algorithm was able to solve the problem `pgvon106` only with this value of λ.

With $\lambda = 0.9$ from the first experiment, we perform the second preliminary experiment (see Table 2) to determine the best value for N. It seems that our algorithm performs best with $N = 10$, which incidentally coincides with our arbitrary choice in the first experiment.

Note that our results are comparable with the ones in [9]. It is worth pointing out that our algorithm only failed in two problems, namely `duopoly` and `shubik`, as opposed to the algorithm in [9], which failed to solve `duopoly`, `hanskoop`, and `shubik`.

With $\lambda = 0.9$ and $N = 10$, we summarize the final results of our experiments on the nonlinear complementarity problems from MCPLIB in Table 3.

The column headings of Table 3 have the following meanings:

Problem: name of the problem in the MCPLIB
n : dimension of the test problem
k : number of iterations
$\|\nabla\Psi(x^f)\|$: value of $\|\nabla\Psi(x)\|$ at the final iterate, $x = x^f$

Table 1: Numerical results for different values of λ

Problem	$\lambda = 0.1$	0.2	0.4	0.6	0.8	0.9	1.0
bertsekas	42	51	57	49	41	36	43
billups	14	14	13	33	37	48	58
colvdual	–	23	58	17	16	24	–
colvnlp	18	16	14	13	21	17	25
cycle	11	4	5	7	6	7	7
degen	13	11	10	9	9	9	8
duopoly	–	–	–	–	–	–	–
explcp	12	10	9	22	22	22	22
hanskoop	14	12	12	11	11	10	11
jel	32	34	25	23	23	21	22
josephy	11	9	8	7	7	7	6
kojshin	10	9	8	7	7	6	6
mathinum	14	12	10	13	18	21	–
mathisum	16	13	12	12	12	10	9
nash	9	7	6	6	6	6	7
pgvon106	–	–	–	–	–	66	–
powell	13	10	9	9	10	10	10
scarfanum	7	7	6	6	5	5	5
scarfasum	7	8	6	6	5	5	5
scarfbsum	145	60	125	70	88	88	153
shubik	–	–	–	–	–	–	–
simple-red	20	17	15	14	14	13	13
sppe	27	21	13	11	10	10	10
tinloi	13	12	11	8	14	16	12
tobin	19	15	11	9	9	9	9

Table 2: Numerical results for different values of N

Problem Name	$N = 0$	5	10	20
bertsekas	30	56	36	36
billups	–	15	48	88
colvdual	19	24	24	19
colvnlp	17	17	17	17
cycle	7	7	7	7
degen	9	9	9	9
duopoly	–	–	–	–
explcp	22	13	22	22
hanskoop	10	10	10	10
jel	21	21	21	21
josephy	7	7	7	7
kojshin	6	6	6	6
mathinum	27	27	27	27
mathisum	10	10	10	10
nash	6	6	6	6
pgvon106	123	–	66	83
powell	10	10	10	10
scarfanum	5	5	5	5
scarfasum	5	5	5	5
scarfbsum	53	111	88	61
shubik	–	–	–	–
simple-red	13	13	13	13
sppe	10	10	10	10
tinloi	16	17	16	16
tobin	9	9	9	9

Table 3: Numerical results for the nonlinear complementarity problems in MCPLIB

Problem	n	k	$\|\nabla\Psi(x^f)\|$
bertsekas	15	36	3.8718e-10
billups	1	48	4.7804e-07
colvdual	20	24	2.4246e-08
colvnlp	15	17	8.4295e-08
cycle	1	7	1.6684e-08
degen	2	9	1.6080e-08
duopoly	63	-	-
explcp	16	22	3.0956e-07
hanskoop	14	10	7.8498e-08
jel	6	21	2.6687e-08
josephy	4	7	4.4231e-08
kojshin	4	6	8.1764e-07
mathinum	3	27	4.8218e-07
mathisum	4	10	1.7593e-08
nash	10	6	3.1753e-07
pgvon106	106	66	6.5453e-07
powell	16	10	2.2440e-07
scarfanum	13	5	5.4244e-07
scarfasum	14	5	5.7432e-07
scarfbsum	40	88	7.0731e-11
shubik	45	-	-
simple-red	13	13	6.8346e-07
sppe	27	10	2.4583e-07
tinloi	146	16	3.3898e-08
tobin	42	9	5.3686e-08

6 Concluding Remarks

In this paper, we considered solving the NCP, giving particular attention to ill-posed problems. To this end, we proposed three techniques in our approach. First, we used the technique of adding redundant equations to the usual Fischer-Burmeister reformulation of the NCP to stabilize the reformulated system. Next, we developed a new Levenberg-Marquardt-type algorithm for solving the system of equations. We have shown that the proposed algorithm converges globally despite the fact that the mapping (Φ) is nondifferentiable. Furthermore, we applied the nonmonotone technique in the implementation of our new algorithm.

Most of the nonlinear complementarity problems in the MCPLIB test problem collection were successfully solved by our algorithm and numerical results show that the developed algorithm is numerically robust.

For further research, we recommend investigating the local convergence properties of our proposed algorithm. We would also like to point out that our method for the standard complementarity problem can be extended to the more general mixed complementarity problem.

Acknowledgments

I would like to express my sincere gratitude and appreciation to Professor Masao Fukushima and Assistant Professor Nobuo Yamashita. Without their help and advice, I could never have written this paper. Also, I would like to thank Matsushita Electric Industrial Company for providing me with financial support during my entire stay in Japan. Lastly, I thank all my friends and labmates for giving me moral support and helpful advice, especially while I was writing this paper.

References

[1] R. W. Cottle, J. S. Pang and R. E. Stone, *The linear complementarity problem,* Academic Press, San Diego (1992).

[2] S. P. Dirkse and M. C. Ferris, *MCPLIB: A collection of nonlinear mixed complementarity problems,* Optimization Methods and Software, 5 (1995), pp. 319-345.

[3] F. Facchinei and J. S. Pang, *Finite-dimensional variational inequalities and complementarity problems,* Springer, New York (2003).

[4] F. Facchinei and J. Soares, *A new merit function for nonlinear complementarity problems and a related algorithm,* SIAM Journal on Optimization, 7 (1997), pp. 225-247.

[5] J. Y. Fan, *A modified Levenberg-Marquardt algorithm for singular system of nonlinear equations,* Journal of Computational Mathematics, 21 (2003), pp. 625-636.

[6] M. C. Ferris and J. S. Pang, *Engineering and economic applications of complementarity problems,* SIAM Review, 39 (1997), pp. 669-713.

[7] A. Fischer, *Solution of monotone complementarity problems with locally Lipschitzian functions,* Mathematical Programming, 76 (1997), pp. 513-532.

[8] C. Kanzow, *An inexact QP-based method for nonlinear complementarity problems,* Numerische Mathematik, 80 (1998), pp. 557-577.

[9] C. Kanzow and S. Petra, *On a semismooth least squares formulation of complementarity problems with gap reduction,* Optimization Methods and Software, 19 (2004), pp. 507-525.

[10] C. Kanzow and H. Pieper, *Jacobian smoothing methods for nonlinear complementarity problems,* SIAM Journal on Optimization, 9 (1999), pp. 342-373.

[11] K. Levenberg, *A method for the solution of certain nonlinear problems in least squares,* Quarterly of Applied Mathematics, 2 (1944), 164-166.

[12] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear inequalities,* SIAM Journal on Applied Mathematics, 11 (1963), 431-441.

[13] J. Nocedal and Y. Yuan, *Combining trust region and line search techniques,* in: Y. Yuan, ed., Advances in Nonlinear Programming, (1998) pp. 153-175.

[14] Y. Yuan, *A review of trust region algorithms for optimization,* in: J. M. Ball and J. C. R. Hunt, eds., ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, (2000) pp. 271-282.