AN INTERIOR POINT METHOD FOR MATHEMATICAL PROGRAMS WITH COMPLEMENTARITY CONSTRAINTS (MPCCs)

ARVIND U. RAGHUNATHAN[†‡] AND LORENZ T. BIEGLER[†§]

Abstract. Interior point methods for nonlinear programs (NLP) are adapted for solution of mathematical programs with complementarity constraints (MPCCs). The constraints of the MPCC are suitably relaxed so as to guarantee a strictly feasible interior for the inequality constraints. The standard primal-dual algorithm has been adapted with a modified step calculation. The algorithm is shown to be superlinearly convergent in the neighborhood of the solution set under assumptions of MPCC-LICQ, strong stationarity and upper level strict complementarity. The modification can be easily accommodated within most nonlinear programming interior point algorithms with identical local behavior. Numerical experience is also presented and holds promise for the proposed method.

Key words. Barrier method, MPECs, Complementarity, NLP.

AMS subject classifications.

1. Introduction.

The MPCC considered in the paper is,

$$\begin{array}{ll} \min\limits_{x,w,y} & f(x,w,y) \\ \text{s.t.} & h(x,w,y) = 0 \\ & x \geq 0 \\ & 0 \leq w \perp y \geq 0 \end{array} \tag{1.1}$$

where $x \in \mathbb{R}^n$, $w, y \in \mathbb{R}^m$, $f : \mathbb{R}^{n+2m} \to \mathbb{R}$, $h : \mathbb{R}^{n+2m} \to \mathbb{R}^q$. For ease of presentation we consider simple bounds on x as general inequality constraints can be posed in a similar form by the introduction of slacks. The above problem is difficult to solve due to the non-satisfaction of constraint qualification (CQ) such as Mangasarian Fromovitz Constraint Qualification (MFCQ) that is readily satisfied for NLPs. The failure of MPCC (1.1) to satisfy MFCQ renders the set of multipliers, if nonempty, unbounded. This has been put forth as an argument for inapplicability of NLP algorithms in the context of MPCCs. A number of algorithms specific to solving Mathematical Programs with Equilibrium Constraints (MPEC) and MPCCs [14, 16] have been proposed. Global [11, 14, 18] and local convergence [5, 15] of the algorithms have also been studied.

In a parallel development researchers have investigated the applicability of nonlinear programming (NLP) algorithms for solving MPCCs. Anitescu [1] in a recent work has investigated the conditions for existence of multipliers for MPCCs. Fletcher *et al.* [7] have shown local convergence of a SQP algorithm when applied to MPCCs. Anitescu and Fletcher *et al.* pose the MPCC as the following NLP,

$$\begin{array}{ll} \min\limits_{x,w,y} & f(x,w,y) \\ \text{s.t.} & h(x,w,y) = 0 \\ & w^T y \leq 0 \\ & x, w, y \geq 0 \end{array} \tag{1.2}$$

[†]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 51213.
[‡]Email : `aur@andrew.cmu.edu`
[§]Email : `lb01@andrew.cmu.edu`

where all the complementarity constraints are aggregated to a single constraint. Interior point approaches have also been proposed for the solution of the MPCC. Benson *et al.* [2] pose the complementarity constraints as

$$\begin{array}{rcl} w_i y_i & \leq & 0 \\ w_i, y_i & \geq & 0 \end{array} \quad i = 1, \dots, m \tag{1.3}$$

and solve the resulting problem with a NLP interior point algorithm LOQO [20]. In the above, where w_i, y_i denote i-th component of the vectors w, y respectively. Observe that the problem does not possess strictly feasible points and yet, Benson *et al.* have reported encouraging numerical results on the MacMPEC test suite [12]. Liu and Sun [13] relax the complementarity constraint as,

$$\begin{array}{rcl} w_i y_i & \leq & t \\ w_i, y_i & \geq & 0 \end{array} \quad i = 1, \dots, m \tag{1.4}$$

for $t > 0$. The parameter t is driven to zero in the limit. The authors describe an interior point approach that achieves global convergence for a given relaxation parameter t under certain assumptions. De Miguel *et al.* [3] propose to solve the MPCC by relaxing the complementarity constraints and the bounds on the complementary variables as,

$$\begin{array}{rcl} w_i y_i & \leq & t_i \\ w_i, y_i & \geq & -\theta_i \end{array} \quad i = 1, \dots, m$$

for $t_i, \theta_i > 0$. De Miguel *et al.* show that in the limit only one of t_i, θ_i needs to be driven to zero. As a result, a problem with nonempty interior can be guaranteed under standard MPCC regularity assumptions and an NLP interior point algorithm can be readily used. De Miguel *et al.* prove superlinear convergence of their algorithm. In this paper we employ an approach similar to Liu and Sun [13] to relax the complementarity constraints. We present a modified NLP barrier algorithm and show that it achieves superlinear convergence in the neighborhood of a solution.

Mathematical programs with equilibrium constraints (MPECs) allow specifying parametric variational inequalities as constraints and generalize MPCCs. A survey of the applications involving MPECs and algorithms can be found in the monograph of Luo, Pang and Ralph [14]. Bilevel programs with convex inner problem can be converted to an MPEC with the parametric variational inequality representing the first order optimality conditions of the program. The MPEC can be converted to a MPCC using the Karush-Kuhn-Tucker (KKT) conditions of the variational inequality, provided the constraint set is expressed as a finite set of inequalities that satisfy a CQ. The theory that is developed in this work can also be applied to the above class of bilevel problems and MPECs.

The paper is organized as follows. We define the constraint qualification, stationarity and second order conditions associated with MPCCs §2. A characterization of the solution set of the MPCC when posed as an NLP (2.5) and some related results are also presented. A modified interior point algorithm is presented in §3. The local convergence analysis in §4 shows that the generated iterates converge quadratically to a solution of the MPCC. We present numerical experience using our proposed algorithm on a test suite of MPCCs in §5 and conclude in §6.

Notation. We will denote the i-th component of a vector $u \in \mathbb{R}^n$ by u_i and denote by u^k the k-th iterate generated by an algorithm or the k-th element of a sequence which will be clear from the context. Given $u \in \mathbb{R}^n$, the upper case letter U will denote an $n \times n$ matrix with elements of the vector u on the diagonal, *i.e.* $U = diag(u_1, \ldots, u_n)$. Norms $\|\cdot\|$ will denote a fixed vector norm and its compatible matrix norm unless otherwise noted. Given $x, y \in \mathbb{R}^n$, $\min(x, y)$ denotes the componentwise minimum, *i.e.* $(\min(x, y))_i = \min(x_i, y_i)$. We will use the order notation in the following standard way. We will denote by $\mathcal{O}(t^k)$ a sequence $\{u^k\}$ satisfying $\|u^k\| \leq \beta_1 t^k$ for some constant $\beta_1 > 0$ independent of k, by $\Theta(t^k)$ a sequence $\{u^k\}$ satisfying $\beta_2 t^k \leq \|u^k\| \leq \beta_3 t^k$ for some $\beta_2, \beta_3 > 0$ independent of k and by $o(t^k)$ a sequence $\{u^k\}$ satisfying $\lim_{k\to\infty}(u^k/t^k) = 0$.

2. Preliminaries. We will first state some definitions. For any feasible point $\tilde{z} := (\tilde{x}, \tilde{w}, \tilde{y})$ of the MPCC we associate a *relaxed* NLP, in the sense of Luo *et al.* [15] and Scheel and Scholtes [17], as follows

$$\begin{array}{ll} \min\limits_{x,w,y} & f(x,w,y) \\ \text{s.t.} & h(x,w,y) = 0 \\ & x \geq 0 \\ & w_i = 0 \quad i \in I_w(\tilde{z}) \setminus I_y(\tilde{z}) \\ & y_i = 0 \quad i \in I_y(\tilde{z}) \setminus I_w(\tilde{z}) \\ & w_i \geq 0 \quad i \in I_y(\tilde{z}) \\ & y_i \geq 0 \quad i \in I_w(\tilde{z}) \end{array} \tag{2.1}$$

where the index sets are defined as, $I_w(\tilde{z}) := \{i \mid \tilde{w}_i = 0\}$ and $I_y(\tilde{z}) := \{i \mid \tilde{y}_i = 0\}$. Due to the complementarity constraint, $I_w(\tilde{z}) \cup I_y(\tilde{z}) = \{1, \ldots, m\}$. The set $I_w(\tilde{z}) \cap I_y(\tilde{z})$ consists of indices $i : \tilde{w}_i = \tilde{y}_i = 0$. Let, $\mathcal{F}_{\text{MPCC}}$ and $\mathcal{F}_{\text{RNLP}}$ respectively denote the feasible regions of MPCC (1.1) and relaxed NLP (2.1). Note that the relaxed NLP (2.1) does not enforce complementarity for $i \in I_w(\tilde{z}) \cap I_y(\tilde{z})$. Consequently, there exists a neighborhood $\mathcal{U}(\tilde{z})$ such that $\mathcal{F}_{\text{MPCC}} \cap \mathcal{U}(\tilde{z}) \subseteq \mathcal{F}_{\text{RNLP}} \cap \mathcal{U}(\tilde{z})$. This has been exploited by Luo *et al.* [15] and Scheel and Scholtes [17] to obtain stationary conditions of the MPCC in terms of stationary conditions of the relaxed NLP for which constraint qualifications such as linear independence of active constraint gradients (LICQ) readily hold.

Definition 2.1 (MPCC-LICQ). *The MPCC* (1.1) *satisfies an MPCC-LICQ at a feasible point, $\tilde{z}$ if the corresponding relaxed NLP* (2.1) *satisfies LICQ at $\tilde{z}$.*

Definition 2.2 (Strong Stationarity). *A point $\overline{z}$ feasible to the MPCC* (1.1) *is called a strong stationary point of the MPCC if there exists a multiplier $\overline{\lambda} := (\overline{\lambda}_h, \overline{\lambda}_x, \overline{\lambda}_w, \overline{\lambda}_y)$ such that $(\overline{z}, \overline{\lambda})$ satisfy NLP stationarity for the* relaxed *NLP* (2.1), *where $\overline{\lambda}_h \in \mathbb{R}^q, \overline{\lambda}_x \in \mathbb{R}^n$, $\overline{\lambda}_w, \overline{\lambda}_y \in \mathbb{R}^m$.*

Definition 2.2 implies that if $\overline{z}$ is a strongly stationary point of the MPCC then,

$$\begin{array}{rcl} \begin{bmatrix} \nabla_x f(\overline{z}) \\ \nabla_w f(\overline{z}) \\ \nabla_y f(\overline{z}) \end{bmatrix} + \begin{bmatrix} \nabla_x h(\overline{z}) \\ \nabla_w h(\overline{z}) \\ \nabla_y h(\overline{z}) \end{bmatrix} \overline{\lambda}_h - \begin{bmatrix} \overline{\lambda}_x \\ \overline{\lambda}_w \\ \overline{\lambda}_y \end{bmatrix} & = & 0 \\ \overline{\lambda}_{x,i} \geq 0, \quad \overline{x}_i \overline{\lambda}_{x,i} & = & 0 \quad i = 1, \ldots, n \\ \overline{\lambda}_{w,i}, \overline{\lambda}_{y,i} & \geq & 0 \quad i \in I_w(\overline{z}) \cap I_y(\overline{z}) \\ \overline{w}_i \overline{\lambda}_{w,i} & = & 0 \quad i = 1, \ldots, m \\ \overline{y}_i \overline{\lambda}_{y,i} & = & 0 \quad i = 1, \ldots, m \\ \overline{z} & \in & \mathcal{F}_{\text{MPCC}}. \end{array} \tag{2.2}$$

If the MPCC satisfies MPCC-LICQ at $\overline{z}$ then, the multiplier $\overline{\lambda}$ is unique.

We now state a second order sufficient condition for MPCCs. The set of critical directions, $\mathcal{T}$, is defined as

$$\mathcal{T} = \left\{ d = \begin{bmatrix} dx \\ dw \\ dy \end{bmatrix} \neq 0 \;\middle|\; \begin{array}{rcll} \nabla_z h(\overline{z})^T d & = & 0 & \\ dx_i & = & 0 & i : \overline{x}_i = 0, \overline{\lambda}_{x,i} > 0 \\ dx_i & \geq & 0 & i : \overline{x}_i = 0, \overline{\lambda}_{x,i} = 0 \\ dw_i & = & 0 & i : i \in I_w(\overline{z}), \overline{\lambda}_{w,i} \neq 0 \\ dw_i & \geq & 0 & i : i \in I_w(\overline{z}), \overline{\lambda}_{w,i} = 0 \\ dy_i & = & 0 & i : i \in I_y(\overline{z}), \overline{\lambda}_{y,i} \neq 0 \\ dy_i & \geq & 0 & i : i \in I_y(\overline{z}), \overline{\lambda}_{y,i} = 0 \end{array} \right\}. \tag{2.3}$$

DEFINITION 2.3 (MPCC-SOSC). *A point $\overline{z}$ satisfies an MPCC-SOSC [19] if strong stationarity* (2.2) *holds and*

$$d^T \nabla_{zz} \mathcal{L}^{\rm RNLP}(\overline{z}, \overline{\lambda}_h, \overline{\lambda}_x, \overline{\lambda}_w, \overline{\lambda}_y) d > 0, \forall d \in \mathcal{T}$$

where $\nabla_{zz}\mathcal{L}^{\rm RNLP}$ is the Hessian of the Lagrangian of the relaxed NLP, $\mathcal{L}^{\rm RNLP}$,

$$\mathcal{L}^{\rm RNLP}(z, \lambda_h, \lambda_x, \lambda_w, \lambda_y) := f(z) + \lambda_h^T h(z) - \lambda_x^T x - \lambda_w^T w - \lambda_y^T y \tag{2.4}$$

and $\mathcal{T}$ is the set of critical directions defined above.

For the remainder of the paper, we make the following standing assumptions.

Assumptions (A)

(A1) The functions f, h are all twice Lipschitz continuously differentiable.

(A2) $\overline{z} := (\overline{x}, \overline{w}, \overline{y})$ is a strong stationary point of MPCC (1.1) with associated MPCC multiplier, $\overline{\lambda} := (\overline{\lambda}_h, \overline{\lambda}_x, \overline{\lambda}_w, \overline{\lambda}_y)$ and MPCC-LICQ, MPCC-SOSC hold.

(A3) Strict complementarity is required to hold for certain components of the MPCC multiplier, $\overline{\lambda}$ *i.e.* $\overline{x}_i + \overline{\lambda}_{x,i} > 0 \; \forall \; i \in 1, \ldots, n$ and $\overline{\lambda}_{w,i}, \overline{\lambda}_{y,i} > 0 \; \forall i \in I_w(\overline{z}) \cap I_y(\overline{z})$.

The assumption of strict complementarity for $i \in I_w(\overline{z}) \cap I_y(\overline{z})$ is also the upper level strict complementarity assumption employed by Scholtes [19]. Assumption (A3) is weaker than requiring $\overline{w}_i + \overline{\lambda}_{w,i} \neq 0$ for all $i \in I_w(\overline{z})$, $\overline{y}_i + \overline{\lambda}_{y,i} \neq 0$ for all $i \in I_y(\overline{z})$ and $\overline{x}_i + \overline{\lambda}_{x,i} > 0$ for all $i = 1, \ldots, n$.

In this paper, we will pose the MPCC (1.1) equivalently as the following NLP

$$\begin{array}{ll} \min\limits_{x,w,y} & f(x, w, y) \\ \text{s.t.} & h(x, w, y) = 0 \\ & Wy \leq 0 \\ & x, w, y \geq 0 \end{array} \tag{2.5}$$

where $W = diag(w_1, \ldots, w_m)$ is a diagonal matrix with components of w on the diagonal. A point $(\overline{x}, \overline{w}, \overline{y}) \in \mathcal{F}_{\rm MPCC}$ satisfies stationary conditions of the NLP (2.5) if there exists $\lambda := (\lambda_h, \lambda_{cc}, \lambda_x, \lambda_w, \lambda_y) \in \mathbb{R}^q \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ satisfying

$$\begin{bmatrix} \nabla_x f(\overline{z}) \\ \nabla_w f(\overline{z}) \\ \nabla_y f(\overline{z}) \end{bmatrix} + \begin{bmatrix} \nabla_x h(\overline{z}) \\ \nabla_w h(\overline{z}) \\ \nabla_y h(\overline{z}) \end{bmatrix} \lambda_h + \begin{bmatrix} 0 \\ \overline{Y} \\ \overline{W} \end{bmatrix} \lambda_{cc} - \begin{bmatrix} \lambda_x \\ \lambda_w \\ \lambda_y \end{bmatrix} = 0 \tag{2.6a}$$

$$\lambda_{cc,i} \geq 0, \quad \lambda_{cc,i}(\overline{w}_i \overline{y}_i) = 0 \quad i = 1, \ldots, m \tag{2.6b}$$

$$\lambda_{x,i} \geq 0, \quad \lambda_{x,i} \overline{x}_i = 0 \quad i = 1, \ldots, n \tag{2.6c}$$

$$\lambda_{w,i} \geq 0, \quad \lambda_{w,i} \overline{w}_i = 0 \quad i = 1, \ldots, m \tag{2.6d}$$

$$\lambda_{y,i} \geq 0, \quad \lambda_{y,i} \overline{y}_i = 0 \quad i = 1, \ldots, m \tag{2.6e}$$

$$\overline{z} \in \mathcal{F}_{\rm MPCC}. \tag{2.6f}$$

The following proposition relates stationary conditions of MPCC (2.2) and NLP (2.6). This result is closely related to [1, Theorem 2.2].

PROPOSITION 2.4 (NLP-MPCC stationarity). *$\overline{z}$ satisfies the NLP stationary conditions* (2.6) *of NLP* (2.5) *if and only if $\overline{z}$ satisfies the strong stationary conditions* (2.2) *of MPCC* (1.1)*, where $\overline{z} := (\overline{x}, \overline{w}, \overline{y})$.*

Proof. The NLP formulation (2.5) ensures that $\overline{z}$ is feasible to the MPCC (1.1) if and only if it is feasible to the NLP (2.5). It remains to verify the remaining conditions.

Suppose $(\overline{x}, \overline{w}, \overline{y}, \lambda_h, \lambda_{cc}, \lambda_x, \lambda_w, \lambda_y)$ satisfies the NLP stationary conditions (2.6). Then define the MPCC multipliers as follows,

$$\begin{array}{rcl} \overline{\lambda}_h &:=& \lambda_h \\ \overline{\lambda}_x &:=& \lambda_x \\ \overline{\lambda}_w &:=& \lambda_w - \overline{Y}\lambda_{cc} \\ \overline{\lambda}_y &:=& \lambda_y - \overline{W}\lambda_{cc} \end{array} \quad \Rightarrow \quad \left. \begin{array}{rcl} \overline{\lambda}_{w,i} &=& \lambda_{w,i} \overset{(2.6d)}{\geq} 0 \\ \overline{\lambda}_{y,i} &=& \lambda_{y,i} \overset{(2.6e)}{\geq} 0 \end{array} \right\} \quad \text{for all } i \in I_w(\overline{z}) \cap I_y(\overline{z}).$$

Hence, $(\overline{x}, \overline{w}, \overline{y}, \overline{\lambda}_h, \overline{\lambda}_x, \overline{\lambda}_w, \overline{\lambda}_y)$ solves the strong stationarity conditions (2.2).

For the reverse implication, suppose $(\overline{x}, \overline{w}, \overline{y}, \overline{\lambda}_h, \overline{\lambda}_x, \overline{\lambda}_w, \overline{\lambda}_y)$ satisfies the strong stationarity conditions (2.2). In order to satisfy the NLP stationary conditions we need to find $(\lambda_{cc}, \lambda_w, \lambda_y) \geq 0$ such that,

$$\begin{array}{rcl} -\overline{Y}\lambda_{cc} + \lambda_w &=& \overline{\lambda}_w \\ -\overline{W}\lambda_{cc} + \lambda_y &=& \overline{\lambda}_y \end{array}$$

holds. Define the NLP multipliers as,

$$\begin{array}{l} \lambda_h := \overline{\lambda}_h \qquad \lambda_x := \overline{\lambda}_x \\ \lambda_{cc,i} := \begin{cases} \max\{-\frac{\overline{\lambda}_{w,i}}{\overline{y}_i}, 0\} & i \in I_w(\overline{z}) \setminus I_y(\overline{z}) \\ 0 & i \in I_w(\overline{z}) \cap I_y(\overline{z}) \\ \max\{-\frac{\overline{\lambda}_{y,i}}{\overline{w}_i}, 0\} & i \in I_y(\overline{z}) \setminus I_w(\overline{z}) \end{cases} \\ \lambda_{w,i} := \begin{cases} \max\{0, \overline{\lambda}_{w,i}\} & i \in I_w(\overline{z}) \setminus I_y(\overline{z}) \\ \overline{\lambda}_{w,i} & i \in I_y(\overline{z}) \end{cases} \\ \lambda_{y,i} := \begin{cases} \overline{\lambda}_{y,i} & i \in I_w(\overline{z}) \\ \max\{0, \overline{\lambda}_{y,i}\} & i \in I_y(\overline{z}) \setminus I_w(\overline{z}). \end{cases} \end{array} \tag{2.7}$$

Then, $(\overline{x}, \overline{w}, \overline{y}, \lambda_h, \lambda_{cc}, \lambda_x, \lambda_w, \lambda_y)$ solves the NLP stationary conditions (2.6). ◻

The multipliers to the NLP obtained in the proof are analogous to the *basic* multipliers that Fletcher *et al.* [7] describe. We will denote the multipliers as λ^{basic}.

The following are results concerning the solution set of the NLP (2.5), which poses the complementarity constraint as an inequality. Given a strong stationary point $\overline{z}$, $\mathcal{S}_\lambda$ will denote the set of multipliers satisfying NLP stationary conditions (2.6) at the point $\overline{z}$. The set $\mathcal{S}_\lambda$ is defined as,

$$\mathcal{S}_\lambda := \left\{ \lambda \left| \begin{array}{rclrcl} \lambda_h &=& \overline{\lambda}_h, & \overline{Y}\lambda_{cc} - \lambda_w &=& -\overline{\lambda}_w, \\ \lambda_x &=& \overline{\lambda}_x, & \overline{W}\lambda_{cc} - \lambda_y &=& -\overline{\lambda}_y, \\ \lambda_{cc}, \lambda_w, \lambda_y &\geq& 0 & & & \end{array} \right. \right\}. \tag{2.8}$$

Since our interest is in the behavior near a particular strongly stationary point, $\overline{z}$, we will suppress the dependence of $\mathcal{S}_\lambda$ on $\overline{z}$. Further, our assumption of MPCC-LICQ (A2) renders $\overline{\lambda}$ unique and there exists no ambiguity in the definition of $\mathcal{S}_\lambda$. Using the definition of $\mathcal{S}_\lambda$ we can prove the following result on λ^{basic}.

PROPOSITION 2.5 (Characterization of $\lambda^{\rm basic}$). *Let $\overline{z}$ be a strong stationary point of MPCC* (1.1) *satisfying MPCC-LICQ. Then,*

$$\lambda^{\rm basic} = \arg\min_{\lambda \in \mathcal{S}_\lambda} \|\lambda\|_1. \tag{2.9}$$

Proof. From the assumption of MPCC-LICQ the multiplier $\overline{\lambda}$ is unique and as a result, there is ambiguity only in the choice of λ_{cc}, λ_w and λ_y. The form of the objective and constraints in (2.9) imply that we have a separable problem. Hence it suffices to show that for each $i \in \{1,\dots,m\}$, $(\lambda^{\rm basic}_{cc,i},\lambda^{\rm basic}_{w,i},\lambda^{\rm basic}_{y,i})$ solves

$$\begin{array}{lll} \displaystyle\min_{\lambda_{cc,i},\lambda_{w,i},\lambda_{y,i}} & \lambda_{cc,i}+\lambda_{w,i}+\lambda_{y,i} \\ \text{s.t.} & \overline{y}_i\lambda_{cc,i}-\lambda_{w,i} = -\overline{\lambda}_{w,i} \\ & \overline{w}_i\lambda_{cc,i}-\lambda_{y,i} = -\overline{\lambda}_{y,i} \\ & \lambda_{cc,i},\lambda_{w,i},\lambda_{y,i} \ge 0 \end{array} \equiv \begin{array}{ll} \displaystyle\min_{\lambda_{cc,i}} & (1+\overline{w}_i+\overline{y}_i)\lambda_{cc,i} \\ \text{s.t.} & \overline{y}_i\lambda_{cc,i}+\overline{\lambda}_{w,i} \ge 0 \\ & \overline{w}_i\lambda_{cc,i}+\overline{\lambda}_{y,i} \ge 0 \\ & \lambda_{cc,i} \ge 0. \end{array} \tag{2.10}$$

where the reduced problem follows from the elimination of $\lambda_{w,i}$ and $\lambda_{y,i}$. The bounds on $\lambda_{cc,i}$ can be further decomposed based on $I_w(\overline{z})$ and $I_y(\overline{z})$ as,

$$\begin{array}{rcl} \lambda_{cc,i} \ge \max(0,-\frac{\overline{\lambda}_{w,i}}{\overline{y}_i}) & : & \text{if } i \in I_w(\overline{z}) \setminus I_y(\overline{z}) \\ \lambda_{cc,i} \ge 0 & : & \text{if } i \in I_w(\overline{z}) \cap I_y(\overline{z}) \\ \lambda_{cc,i} \ge \max(0,-\frac{\overline{\lambda}_{y,i}}{\overline{w}_i}) & : & \text{if } i \in I_y(\overline{z}) \setminus I_w(\overline{z}). \end{array}$$

From the above it is clear that the solution $(\lambda^*_{cc,i},\, \lambda^*_{w,i},\lambda^*_{y,i})$ to (2.10) satisfies,

$$\lambda^*_{cc,i} = \begin{cases} \max(0,-\frac{\overline{\lambda}_{w,i}}{\overline{y}_i}) & \text{if } \; i \in I_w(\overline{z}) \setminus I_y(\overline{z}) \\ 0 & \text{if } \; i \in I_w(\overline{z}) \cap I_y(\overline{z}) \\ \max(0,-\frac{\overline{\lambda}_{y,i}}{\overline{w}_i}) & \text{if } \; i \in I_y(\overline{z}) \setminus I_w(\overline{z}) \end{cases} \qquad \begin{array}{l} \lambda^*_{w,i} = \overline{y}_i\lambda^*_{cc,i}+\overline{\lambda}_{w,i} \\ \lambda^*_{y,i} = \overline{w}_i\lambda^*_{cc,i}+\overline{\lambda}_{y,i}. \end{array}$$

It is easily seen that $(\lambda^*_{cc,i},\lambda^*_{w,i},\, \lambda^*_{y,i})$ is the same as $(\lambda^{\rm basic}_{cc,i},\lambda^{\rm basic}_{w,i},\, \lambda^{\rm basic}_{y,i})$ (2.7). ☐

The above result continues to hold for any $p-norm$ with $p<\infty$. The existence of $\lambda^{\rm basic}$ given $(\overline{z},\overline{\lambda})$ satisfying the strong stationary conditions of the MPCC yields that $\mathcal{S}_\lambda \neq \emptyset$. Hence, the Euclidean projection of any (z,λ) onto the solution set of NLP, $\{\overline{z}\}\times\mathcal{S}_\lambda$, which we denote as $(\hat{z},\hat{\lambda})$ is well defined. Further, the satisfaction of MPCC-SOSC at $\overline{z}$ yields the following result regarding the satisfaction of a second order condition for the NLP (2.5) on the subspace $\mathcal{T}^{\rm NLP}_\lambda$ for $\lambda \in \mathcal{S}_\lambda$,

$$\mathcal{T}^{\rm NLP}_\lambda = \left\{ d = \begin{bmatrix} dx \\ dw \\ dy \end{bmatrix} \neq 0 \;\middle|\; \begin{array}{rcll} \nabla_z h(\overline{z})^T d & = & 0 & \\ dx_i & = & 0 & i : \overline{x}_i = 0, \lambda_{x,i} > 0 \\ dx_i & \ge & 0 & i : \overline{x}_i = 0, \lambda_{x,i} = 0 \\ dw_i & = & 0 & i : i \in I_w(\overline{z}), \lambda_{w,i} > 0 \\ dw_i & \ge & 0 & i : i \in I_w(\overline{z}), \lambda_{w,i} = 0 \\ dy_i & = & 0 & i : i \in I_y(\overline{z}), \lambda_{y,i} > 0 \\ dy_i & \ge & 0 & i : i \in I_y(\overline{z}), \lambda_{y,i} = 0 \\ \overline{y}_i dw_i + \overline{w}_i dy_i & = & 0 & i : \lambda_{cc,i} > 0 \\ \overline{y}_i dw_i + \overline{w}_i dy_i & \le & 0 & i : \lambda_{cc,i} = 0 \end{array} \right\}. \tag{2.11}$$

LEMMA 2.6 (NLP-SOSC). *Suppose Assumptions* (A) *hold. Then,*

$$d^T \nabla_{zz}\mathcal{L}^{\rm NLP}(\overline{z},\lambda) d > 0 \;\; \forall\, d \in \mathcal{T}^{\rm NLP}_\lambda, \lambda \in \mathcal{S}_\lambda$$

where $\mathcal{L}^{\text{NLP}}$ is the Hessian of the Lagrangian of the NLP (2.5)*, defined as,*

$$\mathcal{L}^{\text{NLP}}(z,\lambda) := f(z) + \lambda_h^T h(z) + \lambda_{cc}^T (Wy) - \lambda_x^T x - \lambda_w^T w - \lambda_y^T y. \tag{2.12}$$

Proof. We first show that $\mathcal{T}_\lambda^{\text{NLP}} \subseteq \mathcal{T}$ for all $\lambda \in \mathcal{S}_\lambda$. Next, the terms in $\mathcal{L}^{\text{NLP}}$ due to $Wy \leq 0$ are shown to not contribute to the curvature of the Hessian of $\mathcal{L}^{\text{NLP}}$ for all $d \in \mathcal{T}_\lambda^{\text{NLP}}$ for all $\lambda \in \mathcal{S}_\lambda$. This allows us to prove the claim by noting that the Lagrangian of the NLP (2.12) differs from the Lagrangian for the relaxed NLP (2.4) only in the terms due to the constraint, $Wy \leq 0$.

From the definition of $\mathcal{T}_\lambda^{\text{NLP}}$ (2.11) we observe that for all $i \in I_w(\overline{z})$ and $\lambda \in \mathcal{S}_\lambda$,

$$dw_i \geq 0,\ \overline{y}_i dw_i \leq 0 \text{ for all } d \in \mathcal{T}_\lambda^{\text{NLP}}$$

with one of the inequalities possibly required to hold as an equality. If $i \in I_w(\overline{z}) \setminus I_y(\overline{z})$ *i.e.* $\overline{y}_i > 0$ then $dw_i = 0$. As for $i \in I_w(\overline{z}) \cap I_y(\overline{z})$, Assumption (A3) yields that $dw_i = 0$ since $\lambda_{w,i} \geq \lambda_{w,i}^{\text{basic}} > 0$ for all $\lambda \in \mathcal{S}_\lambda$ (refer Lemma 2.5). A similar result can be shown to hold for dy_i for $i \in I_y(\overline{z})$. Hence,

$$\left.\begin{array}{ll} dw_i = 0 & \text{for all } i \in I_w(\overline{z}) \\ dy_i = 0 & \text{for all } i \in I_y(\overline{z}) \end{array}\right\} \forall\ d \in \mathcal{T}_\lambda^{\text{NLP}}, \lambda \in \mathcal{S}_\lambda. \tag{2.13}$$

On the other hand, the critical directions in $\mathcal{T}$ satisfy

$$\begin{array}{ll} dw_i = 0 \text{ if } i \in I_w(\overline{z}) : \overline{\lambda}_{w,i} \neq 0 & dy_i = 0 \text{ if } i \in I_y(\overline{z}) : \overline{\lambda}_{y,i} \neq 0 \\ dw_i \geq 0 \text{ if } i \in I_w(\overline{z}) : \overline{\lambda}_{w,i} = 0 & dy_i \geq 0 \text{ if } i \in I_y(\overline{z}) : \overline{\lambda}_{y,i} = 0. \end{array}$$

Further, the definition of $\mathcal{S}_\lambda$ (2.8) shows that $\lambda_x = \overline{\lambda}_x$ for all $\lambda \in \mathcal{S}_\lambda$. As a result, we obtain

$$\mathcal{T}_\lambda^{\text{NLP}} \subseteq \mathcal{T} \text{ for all } \lambda \in \mathcal{S}_\lambda. \tag{2.14}$$

From (2.13) observe that,

$$\left.\begin{array}{l} dw_i = 0 \text{ or } dy_i = 0 \text{ for all } i = 1, \ldots, m \\ \Rightarrow\ d^T(\nabla_{zz}(w_i y_i)(\overline{z}))d = 2(dw_i)(dy_i) = 0 \end{array}\right\} \forall\ d \in \mathcal{T}_\lambda^{\text{NLP}}, \lambda \in \mathcal{S}_\lambda. \tag{2.15}$$

In other words, the additional terms in the Lagrangian of NLP (2.5) do not contribute to the curvature of Hessian of $\mathcal{L}^{\text{NLP}}$ for all $d \in \mathcal{T}_\lambda^{\text{NLP}}$ and $\lambda \in \mathcal{S}_\lambda$.

Hence,

$$\begin{aligned} d^T \nabla_{zz} \mathcal{L}^{\text{NLP}}(\overline{z},\lambda) d &\overset{(2.12)}{=} d^T \Bigg(\nabla_{zz} f(\overline{z}) + \sum_{j=1}^{q} \lambda_{h,j} \nabla_{zz} h_j(\overline{z}) \\ &\qquad + \sum_{j=1}^{m} \lambda_{cc,i} \nabla_{zz}(w_j y_j)(\overline{z}) \Bigg) d \\ &\overset{(2.15)}{=} d^T \left(\nabla_{zz} f(\overline{z}) + \sum_{j=1}^{q} \lambda_{h,j} \nabla_{zz} h_j(\overline{z}) \right) d \\ &\overset{(2.8)}{=} d^T \left(\nabla_{zz} f(\overline{z}) + \sum_{j=1}^{q} \overline{\lambda}_{h,j} \nabla_{zz} h_j(\overline{z}) \right) d \\ &\overset{(2.4)}{=} d^T \nabla_{zz} \mathcal{L}^{\text{RNLP}}(\overline{z},\overline{\lambda}) d \\ &\overset{(2.14)}{>} 0 \text{ for all } d \in \mathcal{T}_\lambda^{\text{NLP}}, \lambda \in \mathcal{S}_\lambda \end{aligned}$$

proving the claim. ∎

To summarize, the assumptions of MPCC-LICQ, strong stationarity, MPCC-SOSC (A2) and strict complementarity condition (A3) yield that the NLP (2.5) satisfies the stationary and second order sufficient conditions. This motivates the idea of approaching the solution to the MPCC (1.1) using nonlinear programming algorithms. The algorithm we employ is a *barrier method*. The method solves a sequence of *barrier problems*,

$$\begin{array}{ll} \min\limits_{x,w,y,s_{cc}} & \varphi_\mu(x,w,y,s_{cc}) \\ \text{s.t.} & h(x,w,y) = 0 \\ & Wy + s_{cc} = te_m \end{array} \tag{2.16}$$

where $s_{cc} \in \mathbb{R}^m$, $e_m \in \mathbb{R}^m$ is a vector of all ones, $\mu > 0$ is the *barrier parameter*, $t > 0$ relaxes the complementarity constraint and φ_μ is the barrier objective defined as,

$$\varphi_\mu(x,w,y,s_{cc}) := f(x,w,y) - \mu\sum_{i=1}^{n}\ln(x_i) - \mu\sum_{i=1}^{m}\ln(w_i) - \mu\sum_{i=1}^{m}\ln(y_i) - \mu\sum_{i=1}^{m}\ln(s_{cc,i}).$$

As is evident we have relaxed the constraint $Wy \le 0$ as $Wy \le te_m$. This gives the barrier problem a strictly feasible region which the MPCC does not possess but is necessary for barrier methods. We solve the barrier problem (2.16) for a decreasing sequence of parameters $(\mu, t) \to 0$.

The first order optimality conditions or Karush-Kuhn-Tucker (KKT) conditions for the barrier problem in the *primal-dual* form are

$$\nabla_x f(z) + \nabla_x h(z)\lambda_h - \lambda_x = 0 \tag{2.17a}$$
$$\nabla_w f(z) + \nabla_w h(z)\lambda_h + Y\lambda_{cc} - \lambda_w = 0 \tag{2.17b}$$
$$\nabla_y f(z) + \nabla_y h(z)\lambda_h + W\lambda_{cc} - \lambda_y = 0 \tag{2.17c}$$
$$h(z) = 0 \tag{2.17d}$$
$$Wy + s_{cc} - te_m = 0 \tag{2.17e}$$
$$S_{cc}\Lambda_{cc}e_m - \mu e_m = 0 \tag{2.17f}$$
$$X\Lambda_x e_n - \mu e_n = 0 \tag{2.17g}$$
$$W\Lambda_w e_m - \mu e_m = 0 \tag{2.17h}$$
$$Y\Lambda_y e_m - \mu e_m = 0. \tag{2.17i}$$

The vectors λ_h and λ_{cc} denote the Lagrange multipliers of the appropriate constraints and $\lambda_x, \lambda_w, \lambda_y$ are multipliers for the bounds on x, w, y respectively. It is clear that if a limit exists as $(\mu, t) \to 0$ then, we obtain the NLP stationary conditions (2.6) for the MPCC. The stationary conditions (2.17) also provide additional information on the limiting multiplier. Under Assumption (A3), the minimal multiplier in $\mathcal{S}_\lambda$, λ^{basic} (2.7) may not satisfy strict complementarity as,

$$\begin{array}{lll} \overline{w}_i = 0 = \lambda^{\text{basic}}_{w,i} & \forall & i \in I_w(\overline{z}) : \overline{\lambda}_{w,i} \le 0 \\ \overline{y}_i = 0 = \lambda^{\text{basic}}_{y,i} & \forall & i \in I_y(\overline{z}) : \overline{\lambda}_{y,i} \le 0 \\ \overline{s}_{cc,i} = 0 = \lambda^{\text{basic}}_{cc,i} & \forall & i \in 1,\ldots,m : \overline{\lambda}_{w,i} \ge 0 \text{ or } \overline{\lambda}_{y,i} \ge 0. \end{array} \tag{2.18}$$

From the proof of Proposition 2.5, for any $\lambda \in \mathcal{S}_\lambda$ and all $i \in \{1,\ldots,m\}$ we have that,

$$\lambda_{w,i} \ge \lambda^{\text{basic}}_{w,i},\ \lambda_{y,i} \ge \lambda^{\text{basic}}_{y,i} \text{ and } \lambda_{cc,i} \ge \lambda^{\text{basic}}_{cc,i}. \tag{2.19}$$

The equations (2.18) and (2.19) indicate that $\mathcal{S}_\lambda$ includes multipliers that do not satisfy strict complementarity. More significantly, (2.19) shows that there exist multipliers satisfying strict complementarity for the indices in (2.18). In the following we show that convergence to multipliers in $\mathcal{S}_\lambda$ violating strict complementarity can be avoided by bounding the multipliers with indices in (2.18) away from zero.

LEMMA 2.7 (Lower bounds for $\lambda^j_{cc}, \lambda^j_w, \lambda^j_y$). *Suppose Assumptions (A) hold. Let $\{(z^j, \lambda^j)\}$ solve the barrier problems (2.17) for some $(t^j, \mu^j) > 0$ with $(t^j, \mu^j) \to 0$ and $z^j \to \overline{z}$. Then, for all j sufficiently large*

$$\begin{array}{rcll} \lambda^j_{cc,i} & > & \frac{\mu^j}{t^j} & \forall\ i \in 1, \ldots, m \\ \lambda^j_{w,i} & \geq & \frac{\mu^j}{2t^j}\overline{y}_i & \forall\ i \in I_w(\overline{z}) \setminus I_y(\overline{z}) \\ \lambda^j_{y,i} & \geq & \frac{\mu^j}{2t^j}\overline{w}_i & \forall\ i \in I_y(\overline{z}) \setminus I_w(\overline{z}). \end{array} \tag{2.20}$$

Proof. For $\mu^j > 0$ the solution to the barrier problem satisfies the inequality constraints strictly. Consider $\lambda^j_{cc,i}$,

$$\lambda^j_{cc,i} = \frac{\mu^j}{s^j_{cc,i}} = \frac{\mu^j}{t^j - w^j_i y^j_i} > \frac{\mu^j}{t^j}.$$

As for $\lambda^j_{w,i}$ with $i \in I_w(\overline{z}) \setminus I_y(\overline{z})$,

$$w^j_i = \frac{t^j - s^j_{cc,i}}{y^j_i}.$$

The above is well defined for all j sufficiently large since $\overline{y}_i > 0$ for all $i \in I_w(\overline{z}) \setminus I_y(\overline{z})$ by definition of the index partitioning. This implies that,

$$\lambda^j_{w,i} = \frac{\mu^j}{w^j_i} = \frac{\mu^j}{t^j - s^j_{cc,i}} y^j_i > \frac{\mu^j}{t^j} y^j_i \geq \frac{\mu^j}{2t^j}\overline{y}_i$$

and the claim follows. The remaining claim in (2.20) can be proved similarly. ◻

From Proposition 2.4 we have that any solution to NLP stationary conditions satisfies the MPCC stationary conditions (2.2). When Assumption (A3) holds, Lemma 2.6 ensures that the limit satisfies a second order sufficient condition and Lemma 2.7 ensures that the multiplier limit satisfies strict complementarity by appropriate choice of μ and t. This motivates the use of a barrier approach for solving MPCCs.

We present an interior point algorithm for the solution of MPCCs in the next section, which shows quadratic convergence in the neighborhood of a solution.

3. Algorithm. To simplify the notation we will assume without loss of generality that the NLP (2.5) with relaxed complementarity constraints can be posed as follows

$$\begin{array}{ll} \min\limits_{z} & f(z) \\ \text{s.t.} & c(z;t) = 0 \\ & z \geq 0 \end{array} \tag{3.1}$$

where $z \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ and $c(\cdot;t) : \mathbb{R}^n \to \mathbb{R}^q$. We may identify the variables z and constraints c respectively with the variables (x, w, y, s_{cc}) and constraints $(h(x,w,y), Wy + s_{cc} - te_m)$ of the NLP (2.16). We denote the Lagrangian of the above NLP (3.1) as

$$\mathcal{L}(z, \lambda) := f(z) + \lambda_c^T c(z;t) - \lambda_z^T z \tag{3.2}$$

where $\lambda := (\lambda_c, \lambda_z)$. The algorithm attempts to solve the barrier problem (2.16) approximately and then, decrease t and μ. In the following sections we will describe an algorithm, without any enhancement to promote global convergence, which achieves quadratic convergence when started sufficiently close to a solution of the MPCC (1.1). The stationary conditions of barrier problem associated with NLP in (3.1) are,

$$\begin{array}{rcl} \nabla_z \mathcal{L}(z,\lambda) + \nabla_z c(z;t)\lambda_c - \lambda_z & = & 0 \\ c(z;t) & = & 0 \\ Z\Lambda_z e_n & = & \mu e_n. \end{array} \tag{3.3}$$

We will assume that the MPCC has a solution $(\overline{z}, \overline{\lambda})$ satisfying Assumptions (A). Let (z^*, λ^*) denote any solution of the NLP where $z^* := (\overline{z}, s^*_{cc}), \lambda^* \in \mathcal{S}_\lambda$ with $s^*_{cc} = 0$. The indices of z are partitioned based on the ones active at the solution z^* as,

$$B := \{i | z^*_i > 0\} \text{ and } N := \{i | z^*_i = 0\}. \tag{3.4}$$

The above partitioning is required only for the analysis and not the algorithm.

Given an initial estimate (z^0, λ^0) with $(z^0, \lambda^0_z) > 0$, the algorithm generates a sequence of improved estimates (z^k, λ^k) of the solution to the MPCC. For this purpose in each iteration k, the standard primal-dual interior point method solves the linearization of the KKT conditions (3.3),

$$\begin{bmatrix} \nabla_{zz}\mathcal{L}(z^k,\lambda^k) & \nabla_z c(z^k;t) & -I \\ \nabla_z c(z^k;t)^T & 0 & 0 \\ \Lambda^k_z & 0 & Z^k \end{bmatrix} \begin{pmatrix} dz^k \\ d\lambda^k_c \\ d\lambda^k_z \end{pmatrix} = - \begin{pmatrix} \nabla_z \mathcal{L}(z^k,\lambda^k) \\ c(z^k;t) \\ Z^k\Lambda^k_z e_n - \mu e_n \end{pmatrix} \tag{3.5}$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix. The Jacobian of the constraints $c(\overline{z}; 0)$ is,

$$\nabla_z c(\overline{z};0)^T = \begin{bmatrix} \nabla_x h(\overline{z})^T & \nabla_w h(\overline{z})^T & \nabla_y h(\overline{z})^T & 0 \\ 0 & \overline{Y} & \overline{W} & I_m \end{bmatrix} \tag{3.6}$$

where $I_m \in \mathbb{R}^{m\times m}$ is the identity matrix. Under the assumed MPCC-LICQ the full rank of $\nabla_z h(\overline{z})^T$ is guaranteed and the full rank of $c(\overline{z}; 0)$ is immediate. As has been noted earlier the MPCC (1.1) does not satisfy even the MFCQ at any feasible point. Hence, columns of $\nabla_z c(\overline{z}; 0)$ and gradients of the active bounds at $\overline{z}$ are linearly dependent. Further, strict complementarity can be violated for some $i \in N$; $z^*_i = \lambda^*_{z,i} = 0$. The loss of strict complementarity can result in poor local convergence behavior of the algorithm [24]. Lemma 2.7 shows that convergence to nonstrict complementarity multipliers can be avoided by careful choice of μ and t provided Assumptions (A3) hold. But singularity of the system (3.5) due to non-satisfaction of MFCQ in the limit requires modification of the step calculation as described below. This modification can be shown to allow for rapid local convergence of the algorithm.

The modification we propose is motivated by, but differs from, recent work on modifications of SQP methods in order to guarantee fast local convergence on degenerate problems [9, 25]. The modification ensures that the KKT matrix remains non-singular. The step is obtained by solving the linear system,

$$\begin{bmatrix} \nabla_{zz}\mathcal{L}(z^k,\lambda^k) & \nabla_z c(z^k;t) & -I \\ \nabla_z c(z^k;t)^T & 0 & 0 \\ \tilde{\Lambda}^k_z & 0 & \tilde{Z}^k \end{bmatrix} \begin{pmatrix} dz^k \\ d\lambda^k_c \\ d\lambda^k_z \end{pmatrix} = - \begin{pmatrix} \nabla_z \mathcal{L}(z^k,\lambda^k) \\ c(z^k;t) \\ Z^k\Lambda^k_z e_n - \mu e_n \end{pmatrix} \tag{3.7}$$

where $\tilde{\Lambda}_z^k$ and $\tilde{Z}^k$ are modifications of the matrices Λ_z^k and Z^k satisfying the following for all (z, λ) sufficiently close to a solution,

(M1) $\tilde{z}_i, \tilde{\lambda}_{z,i} \geq 0$.
(M2) $\frac{\tilde{\lambda}_{z,i}}{\tilde{z}_i} = \mathcal{O}(\rho(z, \lambda))$ for all $i \in B$ and $\rho(z, \lambda) = \mathcal{O}(\frac{\tilde{z}_i}{\tilde{\lambda}_{z,i}})$ for all $i \in N$.
(M3) $|\tilde{z}_i - z_i| = \mathcal{O}(\rho(z, \lambda))$ for all $i \in B$ and $|\tilde{\lambda}_{z,i} - \lambda_{z,i}| = \mathcal{O}(\rho(z, \lambda))$ for all $i \in N$.

In the above, $\rho(z, \lambda)$ denotes an estimate of distance of the point $(z, \lambda) > 0$ from the solution set, $\{z^*\} \times \mathcal{S}_\lambda$. A discussion on a suitable estimate follows but we will first focus on the modification.

The following are choices for $\tilde{\Lambda}_z$ and $\tilde{Z}$ that satisfy the conditions (M),

$$\begin{array}{ll} \text{(i)} & \tilde{z}_i = \max(z_i, \eta), \tilde{\lambda}_{z,i} = \max(\lambda_{z,i}, \eta) \quad \forall \quad i \in \{1, \dots, m\} \\ \text{(ii)} & \tilde{z}_i = z_i + \eta\lambda_{z,i}, \tilde{\lambda}_{z,i} = \lambda_{z,i} \quad \forall \quad i \in \{1, \dots, m\} \end{array} \tag{3.8}$$

where η is chosen proportional to the error at the current iterate,

$$\eta = \Theta(\rho(z, \lambda)). \tag{3.9}$$

The first choice, proposed by Vicente and Wright [21], modifies the primal and dual variables while the second modification perturbs only the primal variables. In either case we can show that the local analysis continues to hold true.

We denote the iterate obtained by taking the full step in (3.7) as $(z^{k,\text{full}}, \lambda^{k,\text{full}})$,

$$(z^{k,\text{full}}, \lambda^{k,\text{full}}) := (z^k, \lambda^k) + (dz^k, d\lambda^k). \tag{3.10}$$

The algorithm ensures strict satisfaction of bounds using the projection of the above iterate into the bounds as follows,

$$(z^{k+1}, \lambda^{k+1}) := (z^k, \lambda^k) + \tau^k(P(z^{k,\text{full}}, \lambda^{k,\text{full}}) - (z^k, \lambda^k)) \tag{3.11}$$

where $\tau^k \in (0, 1)$ is usually chosen close to 1 and the projection $P(z, \lambda)$ is given by,

$$P(z, \lambda) := (\max(z, 0), \lambda_c, \max(\lambda_z, 0)). \tag{3.12}$$

It is easily shown that

$$z^{k+1} \geq (1 - \tau^k)z^k \quad \text{and} \quad \lambda_z^{k+1} \geq (1 - \tau^k)\lambda_z^k.$$

The parameter τ^k should converge to 1 in the limit to allow for superlinear convergence. We now define an estimate for the distance of (z, λ) from the solution set,

$$\rho(z, \lambda) := \|(\nabla_z \mathcal{L}(z, \lambda^{\text{HG}}), \tilde{c}(z), \min(z, \lambda))\| \tag{3.13}$$

where

$$\lambda_c^{\text{HG}} = \lambda_c, \quad \lambda_{z,i}^{\text{HG}} = \begin{cases} \lambda_{z,i} & \text{if } \lambda_{z,i} \geq z_i \text{ and } \lambda_{z,i} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \dots, n \tag{3.14}$$

$$\tilde{c}(z) = \begin{bmatrix} h(x, w, y) \\ \min(w, y) \end{bmatrix}. \tag{3.15}$$

We have redefined the constraint residual calculation for NLP (2.5) so that $\rho(z,\lambda)$ serves as an estimate of the distance to the solution set even when $w_i^* = 0 = y_i^*$. From the above definition of the multiplier, λ^{HG} the following result holds,

$$\nabla_z \mathcal{L}(z,\lambda) = \nabla_z \mathcal{L}(z,\lambda^{\mathrm{HG}}) - (\lambda_z - \lambda_z^{\mathrm{HG}}). \tag{3.16}$$

Hager and Gowda [10] show that $\rho(z,\lambda)$ serves as a lower and an upper bound for the distance from the solution set even when constraint qualifications are not satisfied, but a sufficient second order condition holds and (z,λ) is sufficiently close to the solution set, *i.e.*

$$\|z - z^*\| + \|\lambda - \hat{\lambda}\| = \Theta(\rho(z,\lambda)) \tag{3.17}$$

where $\hat{\lambda}$ is the element closest to λ in $\mathcal{S}_\lambda$. Assumptions (A) holding at (z^*,λ^*) ensure that the requirements are met for MPCCs. A summary of the algorithm follows.

Algorithm 1
Choose (z^0,λ^0) with $(z^0,\lambda_z^0) > 0$, tolerance tol > 0 and $\chi_\mu, \chi_t > 0$.
Calculate $\rho(z^0,\lambda^0)$. Set $k \leftarrow 0$.
while $\rho(z^k,\lambda^k) >$ tol **do**
  Choose $\eta^k = \Theta(\rho^k)$, $\mu^k = \chi_\mu(\rho^k)^2$ and $t^k = \chi_t \mu^k$.
  Compute $(dz^k, d\lambda^k)$ by solving (3.7)
  Choose τ^k such that $(1-\tau^k) = \mathcal{O}(\rho(z^k,\lambda^k))$
  Calculate (z^{k+1},λ^{k+1}) using (3.11).
  Set $k \leftarrow k+1$.
end while

4. Local convergence. We analyze the local convergence behavior of Algorithm 1 in this section. The limit point of the sequence of iterates generated by the algorithm will depend greatly on how t and μ are changed. Much of the analysis is along the lines of Vicente and Wright [21]. We establish that the length of the step obtained from the solution of the modified linear system (3.7) is $\mathcal{O}(\rho(z,\lambda))$ in the neighborhood of the solution to the MPCC. Subsequently, we characterize the decrease in the distance to the solution set by taking the step defined in (3.11) and conclude with the convergence result.

Since our interest is in the local behavior we will confine our analysis to a neighborhood around a solution point $(z^*,\lambda^*) \in \{\overline{z}\} \times \mathcal{S}_\lambda$,

$$\mathcal{N}(\epsilon) = \{(z,\lambda) \mid \|(z,\lambda) - (z^*,\lambda^*)\| \le \epsilon\}. \tag{4.1}$$

With a choice of $t = \chi_t \mu$ for some $\chi_t > 0$, we can provide a positive lower bound on the NLP multipliers that could potentially violate strict complementarity ($z_i^* = \lambda_{z,i}^* = 0$) based on Lemma 2.7. Hence, we can make a stronger assumption that we enter a neighborhood, $\mathcal{N}_\gamma(\epsilon)$ of a solution (z^*,λ^*) satisfying strict complementarity *i.e.*,

$$\mathcal{N}_\gamma(\epsilon) := \{(z,\lambda) \mid \|(z,\lambda) - (z^*,\lambda^*)\| \le \epsilon, \lambda_{z,N} \ge \gamma e_N\} \tag{4.2}$$

where $\gamma > 0$ is a constant and e_N is a vector of ones of dimension $|N|$. N is the index set partition defined in (3.4). Lemma 2.7 provides an obvious choice of γ,

$$\gamma \overset{(2.20)}{=} \min \quad \Big(\tfrac{1}{2\chi_t}\min(1, \overline{w}_{i\in I_y(\overline{z})\setminus I_w(\overline{z})}, \overline{y}_{i\in I_w(\overline{z})\setminus I_y(\overline{z})}), \overline{\lambda}_{w,i\in I_w(\overline{z})\cap I_y(\overline{z})}, \overline{\lambda}_{y,i\in I_w(\overline{z})\cap I_y(\overline{z})}, \overline{\lambda}_{x,i:\overline{x}_i=0}\Big).$$

In the following we will drop the superscripts on the variables and assume that they represent the current iterates, k. Additionally, given a vector $v \in \mathbb{R}^n$, v_N will represent a vector in $\mathbb{R}^{|N|}$ formed from components of v corresponding to components with indices in set $N \subseteq \{1,\dots,n\}$, $v_{i:i\in N}$. In a similar manner given the Hessian of the Lagrangian $\nabla_{zz}\mathcal{L} \in \mathbb{R}^{n\times n}$ and $\nabla_{zz}\mathcal{L}_{BN}$ will represent the submatrix of $\nabla_{zz}\mathcal{L}$ formed from rows and columns of $\nabla_{zz}\mathcal{L}$ with indices respectively in the sets B and N (3.4).

We begin by showing that conditions (M) hold for the modifications of Λ_z and Z in (3.8).

Lemma 4.1. *Suppose Assumptions (A) hold, then $\tilde{z}$ and $\tilde{\lambda}_z$ defined in* (3.8) *satisfy conditions (M) for all $(z,\lambda) \in \mathcal{N}_\gamma(\epsilon)$ such that $(z,\lambda_z) \geq 0$.*

Proof. Consider the proposed modification (ii) in (3.8). The definition of $\tilde{z}$ and $\tilde{\lambda}_z$ in (ii) of (3.8) implies the satisfaction of (M1) provided $(z,\lambda_z) \geq 0$. To show that (M2) holds, consider the set B. For all $i \in B$ and $\epsilon > 0$ sufficiently small

$$\begin{aligned}&\tilde{\lambda}_{z,i} \overset{(3.8)}{=} \lambda_{z,i} \overset{(3.17)}{=} \mathcal{O}(\rho(z,\lambda)) \quad \text{and} \quad \tilde{z}_i \overset{(3.8)}{=} z_i + \eta\lambda_{z,i} \geq (z_i^*/2)\\ &\Rightarrow \quad (\tilde{\lambda}_{z,i}/\tilde{z}_i) = \mathcal{O}(\rho(z,\lambda)).\end{aligned}$$

Similarly using $\eta = \Theta(\rho(z,\lambda))$, we have for all $i \in N$ and $\epsilon > 0$ sufficiently small and a constant $\beta > 0$,

$$\begin{aligned}&(\lambda_{z,i}^*/2) \leq \lambda_{z,i} \leq 2\lambda_{z,i}^* \quad \text{and} \quad z_i + \eta\lambda_{z,i} \geq \eta\lambda_{z,i} \geq \beta(\lambda_{z,i}^*/2)\rho\\ &\Rightarrow \rho = \mathcal{O}(\tilde{z}_i/\tilde{\lambda}_{z,i})\end{aligned}$$

which shows that (M2) holds. Satisfaction of (M3) follows from (3.8) and (3.9). The claim for modification (i) can be shown in a similar manner. ∎

The following lemma bounds the step $(dz, d\lambda)$ obtained by solving the linear system in (3.7).

Lemma 4.2 (Bound on $(dz, d\lambda)$). *Suppose Assumptions (A), conditions (M) on the modifications hold, and that for some constant $\chi_t > 0$ we have*

$$\mu = \mathcal{O}(\rho(z,\lambda)^2) \quad \textit{and} \quad t = \chi_t\mu. \tag{4.3}$$

Then for any $\gamma > 0$ there exist positive constants ϵ and $\kappa_1(\epsilon,\gamma)$ such that

$$\|(dz, d\lambda)\| \leq \kappa_1(\epsilon,\gamma)\rho(z,\lambda), \tag{4.4}$$

for all points $(z,\lambda) \in \mathcal{N}_\gamma(\epsilon)$ with $(z,\lambda_z) > 0$.

Proof. We divide the proof into five sections labeled (A)-(E) and simplify notation by dropping the argument (z,λ) along the lines of Vicente and Wright [21].

(A) Transforming (3.7) *via a singular value decomposition (SVD) of the active constraint Jacobian.* Utilizing the partition of $\{1,\dots,n\} = B \cup N$ we can eliminate $d\lambda_{z,B}$ from (3.7) to obtain,

$$d\lambda_{z,B} = -\tilde{Z}_B^{-1}\tilde{\Lambda}_{z,B}dz_B - \tilde{Z}_B^{-1}Z_B\Lambda_{z,B}e_B + \mu\tilde{Z}_B^{-1}e_B. \tag{4.5}$$

Substituting this in (3.7) yields the following reduced system,

$$\begin{aligned}&\begin{bmatrix}\nabla_{zz}\mathcal{L}_{BB} + \tilde{Z}_B^{-1}\tilde{\Lambda}_{z,B} & \nabla_{zz}\mathcal{L}_{BN} & \nabla_z c_B & 0\\ \nabla_{zz}\mathcal{L}_{NB} & \nabla_{zz}\mathcal{L}_{NN} & \nabla_z c_N & -I\\ \nabla_z c_B^T & \nabla_z c_N^T & 0 & 0\\ 0 & -I & 0 & -\tilde{Z}_N\tilde{\Lambda}_{z,N}^{-1}\end{bmatrix}\begin{pmatrix}dz_B\\ dz_N\\ d\lambda_c\\ d\lambda_{z,N}\end{pmatrix} =\\ &\qquad -\begin{pmatrix}\nabla_z\mathcal{L}_B + \tilde{Z}_B^{-1}Z_B\Lambda_{z,B}e_B - \mu\tilde{Z}_B^{-1}e_B\\ \nabla_z\mathcal{L}_N\\ c(z;t)\\ -Z_N\tilde{\Lambda}_{z,N}^{-1}\Lambda_{z,N}e_N + \mu\tilde{\Lambda}_{z,N}^{-1}e_N\end{pmatrix}.\end{aligned} \tag{4.6}$$

where e_B, e_N denote vectors of all ones with $|B|$ and $|N|$ elements, respectively and the last block row has been scaled by $-\tilde{\Lambda}_{z,N}^{-1}$.

We now require the SVD of the Jacobian matrix of the active constraints at z^*,

$$\begin{bmatrix} \nabla_z c_B^{*T} & \nabla_z c_N^{*T} \\ 0 & -I \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{U}^T \\ \hat{V}^T \end{bmatrix}, \tag{4.7}$$

where $U \in \mathbb{R}^{(q+|N|)\times r}, V \in \mathbb{R}^{(q+|N|)\times(q+|N|-r)}, S \in \mathbb{R}^{r\times r}, \hat{U}^T \in \mathbb{R}^{r\times n}, \hat{V}^T \in \mathbb{R}^{(n-r)\times n}$ and r is the rank of the Jacobian matrix. Partitioning rows of V in an obvious way,

$$0 = \begin{bmatrix} V_1^T & V_2^T \end{bmatrix} \begin{bmatrix} \nabla_z c_B^{*T} & \nabla_z c_N^{*T} \\ 0 & -I \end{bmatrix} = \begin{bmatrix} V_1^T \nabla_z c_B^{*T} & (V_1^T \nabla_z c_N^{*T} - V_2^T) \end{bmatrix}. \tag{4.8}$$

We will apply a change of variables using the orthogonal bases $\hat{U}$, $\hat{V}$ and U, V

$$\begin{bmatrix} dz_B \\ dz_N \end{bmatrix} = \hat{U} c_{\hat{U}} + \hat{V} c_{\hat{V}}, \quad \begin{bmatrix} d\lambda_c \\ d\lambda_{z,N} \end{bmatrix} = U c_U + V c_V.$$

Substituting the change of variables in the linear system (4.6) results in,

$$\begin{bmatrix} \hat{U}^T L \hat{U} & \hat{U}^T L \hat{V} & \hat{U}^T J^T U & \hat{U}^T J^T V \\ \hat{V}^T L \hat{U} & \hat{V}^T L \hat{V} & \hat{V}^T J^T U & \hat{V}^T J^T V \\ U^T J \hat{U} & U^T J \hat{V} & U^T M U & U^T M V \\ V^T J \hat{U} & V^T J \hat{V} & V^T M U & V^T M V \end{bmatrix} \begin{pmatrix} c_{\hat{U}} \\ c_{\hat{V}} \\ c_U \\ c_V \end{pmatrix} = \begin{pmatrix} l_{\hat{U}} \\ l_{\hat{V}} \\ l_U \\ l_V \end{pmatrix}, \tag{4.9}$$

where the matrices L, J and M are given by

$$\begin{aligned} L &= \begin{bmatrix} \nabla_{zz}\mathcal{L}_{BB} + \tilde{Z}_B^{-1}\tilde{\Lambda}_{z,B} & \nabla_{zz}\mathcal{L}_{BN} \\ \nabla_{zz}\mathcal{L}_{NB} & \nabla_{zz}\mathcal{L}_{NN} \end{bmatrix} \quad J = \begin{bmatrix} \nabla_z c_B^T & \nabla_z c_N^T \\ 0 & -I \end{bmatrix} \\ M &= \begin{bmatrix} 0 & 0 \\ 0 & -\tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1} \end{bmatrix} \end{aligned} \tag{4.10}$$

and the residuals $l_{\hat{U}}, l_{\hat{V}}, l_U$ and l_V are given by

$$\begin{aligned} l_{\hat{U}} &= -\hat{U}^T \begin{bmatrix} \nabla_z \mathcal{L}_B + \tilde{Z}_B^{-1} Z_B \Lambda_{z,B} e_B - \mu \tilde{Z}_B^{-1} e_B \\ \nabla_z \mathcal{L}_N \end{bmatrix} \\ l_{\hat{V}} &= -\hat{V}^T \begin{bmatrix} \nabla_z \mathcal{L}_B + \tilde{Z}_B^{-1} Z_B \Lambda_{z,B} e_B - \mu \tilde{Z}_B^{-1} e_B \\ \nabla_z \mathcal{L}_N \end{bmatrix} \\ l_U &= -U^T \begin{bmatrix} c(z;t) \\ -Z_N \tilde{\Lambda}_{z,N}^{-1} \Lambda_{z,N} e_N + \mu \tilde{\Lambda}_{z,N}^{-1} e_N \end{bmatrix} \\ l_V &= -V^T \begin{bmatrix} c(z;t) \\ -Z_N \tilde{\Lambda}_{z,N}^{-1} \Lambda_{z,N} e_n + \mu \tilde{\Lambda}_{z,N}^{-1} e_N \end{bmatrix}. \end{aligned} \tag{4.11}$$

(B) Examining $V^T M V$ and its inverse. We can write the blocks of (4.9) involving M using (4.10) as,

$$\begin{aligned} \begin{bmatrix} U^T \\ V^T \end{bmatrix} M \begin{bmatrix} U & V \end{bmatrix} &= \begin{bmatrix} U_1^T & U_2^T \\ V_1^T & V_2^T \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -\tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1} \end{bmatrix} \begin{bmatrix} U_1 & V_1 \\ U_2 & V_2 \end{bmatrix} \\ &= -\begin{bmatrix} U_2^T \tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1} U_2 & U_2^T \tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1} V_2 \\ V_2^T \tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1} U_2 & V_2^T \tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1} V_2 \end{bmatrix} =: \begin{bmatrix} \overline{M}_{11} & \overline{M}_{12} \\ \overline{M}_{21} & \overline{M}_{22} \end{bmatrix}. \end{aligned} \tag{4.12}$$

From the definition of $\overline{M}_{22}$ block and the property of $\tilde{\lambda}_z$ and $\tilde{z}$ in (M2), $\rho = \mathcal{O}(\tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1})$. This implies that the matrix $\tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1}$ is positive diagonal. It is easily shown from full rank of $\nabla_z c^*$ (3.6) and partitioning of V (4.8) that V_2 has full column rank [21]. Hence, the eigenvalues of $\overline{M}_{22}$ behave the same way as $\tilde{Z}_N \tilde{\Lambda}_{z,N}^{-1}$ and this yields that,

$$\overline{M}_{22}^{-1} = \mathcal{O}(\rho^{-1}). \tag{4.13}$$

(C) Estimating other blocks in the linear system (4.9) The contribution of the Jacobian terms may be estimated using the twice Lipschitz continuously differentiable assumption on constraints (A1) as follows,

$$\begin{bmatrix} U \\ V \end{bmatrix}^T J \begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix}^T (J - J^*) \begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix} + \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \overset{\text{(A1)},(3.13)}{=} \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \mathcal{O}(\rho). \tag{4.14}$$

The Hessian of the Lagrangian can be similarly bounded,

$$\nabla_{zz}\mathcal{L}(z,\lambda) \overset{\text{(A1)},(3.17)}{=} \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda}) + \mathcal{O}(\rho),$$

where $\hat{\lambda}$ is the element in $\mathcal{S}_\lambda$ closest to the current iterate λ. Using this,

$$\begin{bmatrix} \hat{U}^T L \hat{U} & \hat{U}^T L \hat{V} \\ \hat{V}^T L \hat{U} & \hat{V}^T L \hat{V} \end{bmatrix} \overset{\text{(M2)},(4.10)}{=} \begin{bmatrix} \hat{U}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{U} & \hat{U}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V} \\ \hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{U} & \hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V} \end{bmatrix} + \mathcal{O}(\rho). \tag{4.15}$$

Substituting estimates (4.14), (4.15) into the matrix in (4.9) we obtain

$$\begin{bmatrix} \hat{U}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{U} & \hat{U}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V} & S & 0 \\ \hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{U} & \hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V} & 0 & 0 \\ S & 0 & 0 & 0 \\ 0 & 0 & 0 & \overline{M}_{22} \end{bmatrix} + \begin{bmatrix} \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) \\ \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) \\ \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) \\ \mathcal{O}(\rho) & \mathcal{O}(\rho) & \mathcal{O}(\rho) & 0 \end{bmatrix}. \tag{4.16}$$

Eliminating c_V from the linear system (4.9) and using the estimate of $\overline{M}_{22}$ from (4.13),

$$c_V = \mathcal{O}(\rho^{-1})(l_V + \mathcal{O}(\rho)c_{\hat{U}} + \mathcal{O}(\rho)c_{\hat{V}} + \mathcal{O}(\rho)c_U). \tag{4.17}$$

Using c_V from (4.17), the linear system (4.9) after reordering reduces to,

$$\left(\begin{bmatrix} S & \hat{U}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V} & \hat{U}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{U} \\ 0 & \hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V} & \hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{U} \\ 0 & 0 & S \end{bmatrix} + \mathcal{O}(\rho) \right) \begin{bmatrix} c_U \\ c_{\hat{V}} \\ c_{\hat{U}} \end{bmatrix} = \begin{bmatrix} l_U \\ l_{\hat{V}} \\ l_{\hat{U}} \end{bmatrix} + \mathcal{O}(\|l_V\|). \tag{4.18}$$

Lemma 2.6 applies to yield that the matrix $\hat{V}^T \nabla_{zz}\mathcal{L}(z^*,\hat{\lambda})\hat{V}$ is uniformly nonsingular for all $\hat{\lambda} \in \mathcal{S}_\lambda$ and the matrix in (4.18) is an $\mathcal{O}(\rho)$ perturbation of a nonsingular matrix.

(D) Estimating the right hand side components in the linear system (4.9). From boundedness of $\mathcal{N}_\gamma(\epsilon)$,

$$\begin{aligned} z_i \lambda_{z,i} &= \min(z_i, \lambda_{z,i}) \max(z_i, \lambda_{z,i}) = \mathcal{O}(\min(z_i, \lambda_{z,i})) \\ &\overset{(3.13)}{=} \mathcal{O}(\rho) \text{ for all } i = 1, \dots, n. \end{aligned} \tag{4.19}$$

We have that $\nabla_z \mathcal{L}(z,\lambda) = \mathcal{O}(\rho)$ by,

$$\nabla_z \mathcal{L}(z,\lambda) \overset{(3.16)}{=} \nabla_z \mathcal{L}(z,\lambda^{\text{HG}}) - (\lambda_z - \lambda_z^{\text{HG}}) \overset{(3.13),(3.14)}{=} \mathcal{O}(\rho). \tag{4.20}$$

Using these estimates, the bound on $\tilde{z}_B$ from (M3) and $\mu = \mathcal{O}(\rho^2)$ in (4.11),

$$l_{\hat{U}} = \mathcal{O}(\rho), \quad l_{\hat{V}} = \mathcal{O}(\rho). \tag{4.21}$$

For all $i \in N$, $z_N = z_N - z_N^* = \mathcal{O}(\rho)$. We have $c(z;0) = \mathcal{O}(\rho)$ from (3.13) and $\mu, t = \mathcal{O}(\rho^2)$ from (4.3) yielding

$$l_U = \mathcal{O}(\rho). \tag{4.22}$$

We bound l_V using (3.17), (4.3), (4.8), (A1), (M3) in (4.11),

$$\begin{aligned}
l_V &\overset{(4.11)}{=} -V^T \begin{bmatrix} c(z;t) \\ -Z_N \tilde{\Lambda}_{z,N}^{-1} \Lambda_{z,N} e_N + \mu \tilde{\Lambda}_{z,N}^{-1} e_N \end{bmatrix} \\
&\overset{(A1)}{=} -V^T \begin{bmatrix} c(z^*;0) + \nabla_z c(z^*)^T (z - z^*) + \mathcal{O}(\|z - z^*\|^2) + \Theta(t) \\ -(Ze - Z^* e)_N + \mu \tilde{\Lambda}_{z,N}^{-1} e_N + Z_N \tilde{\Lambda}_{z,N}^{-1} (\tilde{\Lambda}_{z,N} - \Lambda_{z,N}) e_N \end{bmatrix} \\
&\overset{(M3)}{=} -V^T \begin{bmatrix} \nabla_z c_B^{*T} & \nabla_z c_N^{*T} \\ 0 & -I \end{bmatrix} \begin{bmatrix} z_B - z_B^* \\ z_N - z_N^* \end{bmatrix} - V^T \begin{bmatrix} \mathcal{O}(\|z - z^*\|^2) + \Theta(t) \\ \mu \tilde{\Lambda}_{z,N}^{-1} e_N + \mathcal{O}(\rho^2) \end{bmatrix} \\
&\overset{(4.8)}{=} -V^T \begin{bmatrix} \mathcal{O}(\|z - z^*\|^2) + \Theta(t) \\ \mu \tilde{\Lambda}_{z,N}^{-1} e_N + \mathcal{O}(\rho^2) \end{bmatrix} \overset{(3.17),(4.3)}{=} \mathcal{O}(\rho^2).
\end{aligned} \tag{4.23}$$

(E) Estimating the size of the components $c_U, c_V, c_{\hat{U}}$ and $c_{\hat{V}}$ From uniform nonsingularity of the matrix in (4.18) for ρ sufficiently small,

$$c_U, c_{\hat{U}}, c_{\hat{V}} \overset{(4.21),(4.22)}{=} \mathcal{O}(\rho).$$

The bound on c_V is obtained by using (4.23) and the above result in (4.17),

$$c_V = \mathcal{O}(\rho).$$

Since the step $(dz, d\lambda_c, d\lambda_{z,N})$ is an orthogonal transformation of $(c_{\hat{U}}, c_{\hat{V}}, c_U, c_V)$,

$$\|(dz, d\lambda_c, d\lambda_{z,N})\| \overset{(3.9)}{=} \mathcal{O}(\rho).$$

To complete we provide the estimate on $d\lambda_{z,B}$ using (3.9), (4.3) and (M3),

$$\begin{aligned}
& d\lambda_{z,B} = -\tilde{Z}_B^{-1} \tilde{\Lambda}_{z,B} dz_B - \tilde{Z}_B^{-1} Z_B \Lambda_{z,B} e_B + \mu \tilde{Z}_B^{-1} e_B = \mathcal{O}(\rho) \\
\Rightarrow \quad & \|(dz, d\lambda)\| = \mathcal{O}(\rho).
\end{aligned}$$

□

The following result states the progress resulting from a full step.

LEMMA 4.3 (Bound on $\rho(z^{\text{full}}, \lambda^{\text{full}})$). *Suppose Assumptions (A), conditions (M) on the modifications hold and μ, t are given by (4.3). Then given any $\gamma > 0$, there exist positive constants ϵ and $\kappa_2(\epsilon, \gamma)$ such that*

$$\rho(z^{\text{full}}, \lambda^{\text{full}}) \leq \kappa_2(\epsilon, \gamma) \rho(z, \lambda)^2 \tag{4.24}$$

for all $(z, \lambda) \in \mathcal{N}_\gamma(\epsilon)$ with $(z, \lambda_z) \geq 0$.

Proof. To establish the bound on $\rho(z^{\text{full}}, \lambda^{\text{full}})$, we will bound each of the components in the error estimate (3.13). We will begin by establishing bounds on z^{full} and λ_z^{full}. Consider dz_i for $i \in N$,

$$dz_N \overset{(3.7)}{=} -Z_N \tilde{\Lambda}_{z,N}^{-1} \Lambda_{z,N} e_N - Z_N \tilde{\Lambda}_{z,N}^{-1} d\lambda_{z,N} + \mu \tilde{\Lambda}_{z,N}^{-1} e_N$$

and hence from (M3),(4.3), (4.4) and $z_N = \mathcal{O}(\rho)$,

$$\begin{array}{rcl} \|z_N + dz_N\| & \leq & \|Z_N \tilde{\Lambda}_{z,N}^{-1}(\tilde{\Lambda}_{z,N} - \Lambda_{z,N})e_N\| + \|\tilde{\Lambda}_{z,N}^{-1} Z_N d\lambda_{z,N}\| + \mu\|\tilde{\Lambda}_{z,N}^{-1} e_N\| \\ & = & \mathcal{O}(\rho^2). \end{array} \tag{4.25}$$

Consider $d\lambda_{z,B}$,

$$d\lambda_{z,B} = -\tilde{Z}_B^{-1} Z_B \Lambda_{z,B} e_B - \tilde{Z}_B^{-1}\tilde{\Lambda}_{z,B} dz_B + \mu \tilde{Z}_B^{-1} e_B$$

and hence from (M3),(4.3), (4.4) and $\lambda_{z,B} = \mathcal{O}(\rho)$,

$$\begin{array}{rcl} \|\lambda_{z,B} + d\lambda_{z,B}\| & \leq & \|\tilde{Z}_B^{-1}(\tilde{Z}_B - Z_B)\Lambda_{z,B} e_B\| + \|\tilde{Z}_B^{-1}\tilde{\Lambda}_{z,B} dz_B\| + \mu\|\tilde{Z}_B^{-1} e_B\| \\ & = & \mathcal{O}(\rho^2). \end{array} \tag{4.26}$$

The bounds on z_B^{full} and $\lambda_{z,N}^{\text{full}}$ may be derived for all $\epsilon > 0$ sufficiently small as,

$$\begin{array}{rcl} z_B^{\text{full}} & = & z_B + dz_B \overset{(4.4)}{=} z_B + \mathcal{O}(\rho) \overset{(3.4)}{\geq} (1/2) z_B^* > 0 \\ \lambda_{z,N}^{\text{full}} & = & \lambda_{z,N} + d\lambda_{z,N} \overset{(4.4)}{=} \lambda_{z,N} + \mathcal{O}(\rho) \overset{(3.4)}{\geq} (\gamma/2) e_N > 0. \end{array} \tag{4.27}$$

Consider each of the components of ρ starting with the last component in (3.13). The bound is established below,

$$|\min(z_i^{\text{full}}, \lambda_{z,i}^{\text{full}})| = \begin{cases} |\lambda_{z,i}^{\text{full}}| \overset{(4.26),(4.27)}{=} \mathcal{O}(\rho^2), & \text{for all} \quad i \in B \\ |z_i^{\text{full}}| \overset{(4.25),(4.27)}{=} \mathcal{O}(\rho^2), & \text{for all} \quad i \in N. \end{cases} \tag{4.28}$$

We now derive a bound on $\nabla_z \mathcal{L}(z^{\text{full}}, \lambda^{\text{full,HG}})$. Expanding $\nabla_z \mathcal{L}(z^{\text{full}}, \lambda^{\text{full}})$ using Taylor's series and substituting in the first row of (3.7) we obtain,

$$\begin{array}{rcl} \nabla_z \mathcal{L}(z^{\text{full}}, \lambda^{\text{full}}) & = & \nabla_z \mathcal{L}(z + dz, \lambda + d\lambda) \\ & \overset{(\text{A1})}{=} & \nabla_z \mathcal{L}(z, \lambda + d\lambda) + \nabla_{zz} f(z) dz + \mathcal{O}(\|dz\|^2) \\ & & + \sum_{i=1}^{q} (\lambda_{c,i} + d\lambda_{c,i}) \nabla_{zz} c_i(z) dz + \mathcal{O}(\|dz\|^2 \|\lambda + d\lambda\|) \\ & = & (\nabla_z \mathcal{L}(z, \lambda) + (\nabla_{zz} f(z) + \sum_{i=1}^{q} \lambda_{c,i} \nabla_{zz} c_i(z)) dz + \\ & & \nabla_z c(z) d\lambda_c - d\lambda_z) + \sum_{i=1}^{q} d\lambda_{c,i} \nabla_{zz} c_i(z) dz \\ & & + \mathcal{O}(\|dz\|^2) + \mathcal{O}(\|dz\|^2 \|\lambda + d\lambda\|) \\ & \overset{(3.7)}{=} & \sum_{i=1}^{q} d\lambda_{c,i} \nabla_{zz} c_i(z) dz + \mathcal{O}(\|dz\|^2) + \mathcal{O}(\|dz\|^2 \|\lambda + d\lambda\|) \\ & \overset{(\text{A1})}{=} & \mathcal{O}(\|d\lambda\| \|dz\|) + \mathcal{O}(\|dz\|^2) + \mathcal{O}(\|dz\|^2 \|\lambda + d\lambda\|) \\ & \overset{(4.4)}{=} & \mathcal{O}(\rho^2). \end{array} \tag{4.29}$$

From the definition of λ^{HG} (3.14) and $(z^{\text{full}}, \lambda_z^{\text{full}}) > 0$ we have that,

$$\begin{array}{rcl} \nabla_z \mathcal{L}(z^{\text{full}}, \lambda^{\text{full,HG}}) & \overset{(3.16)}{=} & \nabla_z \mathcal{L}(z^{\text{full}}, \lambda^{\text{full}}) + (\lambda_z^{\text{full}} - \lambda_z^{\text{full,HG}}) \\ & \overset{(4.29)}{=} & \mathcal{O}(\rho^2) + \mathcal{O}(\min(z^{\text{full}}, \lambda_z^{\text{full}})) \overset{(4.28)}{=} \mathcal{O}(\rho^2). \end{array}$$

Similarly, for $c(z+dz;0)$ we have that,

$$\begin{array}{rcl} c(z+dz;0) & = & c(z;0) + \nabla_z c(z)^T dz + \mathcal{O}(\|dz\|^2) \\ & \overset{(3.7)}{=} & c(z;0) - c(z;t) + \mathcal{O}(\|dz\|^2) \\ & = & \mathcal{O}(t) + \mathcal{O}(\|dz\|^2) \\ & \overset{(4.3),(4.4)}{=} & \mathcal{O}(\rho^2). \end{array} \tag{4.30}$$

The bounds on $\min(w^{\text{full}}, y^{\text{full}})$ may be deduced as,

$$\|\min(w^{\text{full}}, y^{\text{full}})\| \le \|z_N^{\text{full}}\| \overset{(4.25)}{=} \mathcal{O}(\rho^2). \tag{4.31}$$

Collecting results in (4.30) and (4.31), $\tilde{c}(z^{\text{full}}) = \mathcal{O}(\rho^2)$. From the above results we have that, $\rho(z^{\text{full}}, \lambda^{\text{full}}) = \mathcal{O}(\rho^2)$. ∎

We now show a similar result to hold for the new iterate (z^+, λ^+).

LEMMA 4.4 (Bound on $\rho(z^+, \lambda^+)$). *Suppose Assumptions* (A), *conditions* (M) *on the modifications hold with* μ, t *are given by* (4.3) *and* τ *satisfying,*

$$1 - \tau \le \zeta\rho. \tag{4.32}$$

for some $\zeta > 0$. *Then given* γ, $\zeta > 0$, *there exist positive constants* ϵ *and* $\kappa_3(\epsilon, \gamma, \zeta)$ *such that*

$$\rho(z^+, \lambda^+) \le \kappa_3(\epsilon, \gamma, \zeta)\rho(z, \lambda)^2 \tag{4.33}$$

for all $(z, \lambda) \in \mathcal{N}_\gamma(\epsilon)$ *with* $(z, \lambda_z) \ge 0$.

Proof. We begin by providing a bound on difference between $(z^{\text{full}}, \lambda^{\text{full}})$ and its projection onto the bounds, $P(z^{\text{full}}, \lambda^{\text{full}})$. The projection onto the bounds alters only those negative components of $(z^{\text{full}}, \lambda^{\text{full}})$ that have bounds. Hence,

$$\begin{array}{rcl} \|P(z^{\text{full}}, \lambda^{\text{full}}) - (z^{\text{full}}, \lambda^{\text{full}})\| & = & \|(\max(z^{\text{full}}, 0), \max(\lambda_z^{\text{full}}, 0)) - (z^{\text{full}}, \lambda_z^{\text{full}})\| \\ & \overset{(4.27)}{=} & \|(\max(z_N^{\text{full}}, 0) - z_N^{\text{full}}, \max(\lambda_{z,B}^{\text{full}}, 0) - \lambda_{z,B}^{\text{full}})\| \\ & \le & \|z_N^{\text{full}}\| + \|\lambda_{z,B}^{\text{full}}\| \overset{(4.25),(4.26)}{=} \mathcal{O}(\rho^2). \end{array}$$

The distance between (z^+, λ^+) and $(z^{\text{full}}, \lambda^{\text{full}})$ can be bounded using the above as,

$$\begin{array}{rcl} \|(z^+, \lambda^+) - (z^{\text{full}}, \lambda^{\text{full}})\| & \overset{(3.12)}{=} & \|(1-\tau)((z, \lambda) - P(z^{\text{full}}, \lambda^{\text{full}})) \\ & & +(P(z^{\text{full}}, \lambda^{\text{full}}) - (z^{\text{full}}, \lambda^{\text{full}}))\| \\ & \le & |1-\tau|\|(dz, d\lambda)\| + \mathcal{O}(\rho^2) \overset{(4.4),(4.32)}{=} \mathcal{O}(\rho^2). \end{array} \tag{4.34}$$

This sets the stage for bounding the distance of (z^+, λ^+) from the solution set,

$$\begin{array}{rcl} \|(z^+, \lambda^+) - (z^*, \widehat{\lambda^+})\| & \le & \|(z^+, \lambda^+) - (z^{\text{full}}, \lambda^{\text{full}})\| + \|(z^{\text{full}}, \lambda^{\text{full}}) - (z^*, \widehat{\lambda^{\text{full}}})\| \\ & & +\|(z^*, \widehat{\lambda^{\text{full}}}) - (z^*, \widehat{\lambda^+})\| \\ & \le & \|(z^+, \lambda^+) - (z^{\text{full}}, \lambda^{\text{full}})\| + \|(z^{\text{full}}, \lambda^{\text{full}}) - (z^*, \widehat{\lambda^{\text{full}}})\| \\ & & +\|\widehat{\lambda^{\text{full}}} - \widehat{\lambda^+}\| \\ & \overset{(4.34)}{\le} & \mathcal{O}(\rho^2) + \mathcal{O}(\rho(z^{\text{full}}, \lambda^{\text{full}})) + \|\lambda^{\text{full}} - \lambda^+\| \\ & \overset{(4.24)}{\le} & \mathcal{O}(\rho^2) + \|\lambda^{\text{full}} - \lambda^+\| \overset{(4.34)}{=} \mathcal{O}(\rho^2) \end{array}$$

where $\widehat{\lambda^+}$, $\widehat{\lambda^{\text{full}}}$ respectively refer to the projection of λ^+, λ^{full} onto $\mathcal{S}_\lambda$. The penultimate inequality follows from the non-expansive property of the projection onto the set of NLP multipliers, $\mathcal{S}_\lambda$. The above result with (3.17) shows the claim. ☐

From Lemmas 4.2 and 4.4 we have respectively, that the steps from Algorithm 1 at each iteration is bounded by ρ and further, make excellent progress towards the solution. Using these we are in a position to state the convergence result. We omit the proof of the following result for brevity as it follows from Theorem 4.3 of Vicente and Wright [21] and the compactness of $\mathcal{N}_\gamma(\epsilon)$.

THEOREM 4.5 (Quadratic convergence). *Suppose Assumptions* (A)*, conditions* (M) *on the modifications hold and* μ, t *are given by* (4.3)*. Let* $\gamma, \zeta > 0$ *be given, and consider a value of* ϵ *such that Lemmas 4.2 and 4.4 are applicable. If the initial point* $(z^0, \lambda^0) \in \mathcal{N}_{2\gamma}(\epsilon/2)$ *and*

$$\rho(z^0, \lambda^0) \leq \min\left\{\frac{1}{2\kappa_3(\epsilon,\gamma,\zeta)}, \frac{\epsilon}{4\kappa_1(\epsilon,\gamma)}, \frac{\gamma}{2\kappa_1(\epsilon,\gamma)}\right\},$$

then the iterates (z, λ) *generated by Algorithm* 1 *remain inside the neighborhood* $\mathcal{N}_\gamma(\epsilon)$ *and converge q-quadratically to a point* (z^*, λ^{**}) *where* $\lambda^{**} \in \mathcal{S}_\lambda$*.*

5. Numerical Experience. Our motivation for the numerical study stems from a desire to understand the practical performance of the proposed interior point algorithm, Algorithm I. In order to assume that the iterates eventually enter the neighborhood $\mathcal{N}_\gamma$ we need to bound μ/t away from zero but we cannot guarantee bounded multipliers. It will be interesting to see if we converge to bounded multipliers in this case. We provide a brief description of the implementation of the algorithm and also provide a comparison with two other algorithms, LOQO [2] and filtermpec [6].

5.1. Implementation with IPOPT. The algorithm proposed in §3 can be easily incorporated within most existing interior point NLP algorithms. Here we present preliminary numerical experience with incorporating the above modification within such a NLP algorithm, IPOPT [22, 23]. The basic steps of the algorithm with the proposed step modification given a NLP, such as in (3.1), are provided in Algorithm - IPOPT-C.

IPOPT [22, 23] is an interior point NLP algorithm employing a filter to promote global convergence from initial iterates (z^0, λ^0) that are far removed from a solution to the problem. A search direction is obtained by solving a linear system, such as (3.7). The step size, $\alpha^{\max}$ is calculated so that the resulting iterate is within the bounds, *i.e.*

$$z + \alpha^{\max} dz > 0 \text{ and } \lambda_z + \alpha^{\max} d\lambda_z > 0.$$

We have employed this instead of the projection step (3.11) as global convergence cannot be guaranteed in the latter case for the current implementation in IPOPT. Our experience has been that in this case full steps are taken in the limit, *i.e.*

$$(1 - \alpha^{\max}) = \mathcal{O}(\rho).$$

Using the above, quadratic convergence of the implementation within IPOPT can be shown using the analysis in §4. A backtracking line search is performed to obtain a step size, α^k, and an iterate that achieves sufficient decrease either in the objective function or the constraint violation. Further information on the filter can be obtained from Wächter [22] and Wächter and Biegler [23]. In certain iterations, even sufficiently

Algorithm - IPOPT-C

Choose (z^0, λ^0) with $(z^0, \lambda_z^0) > 0$, tolerance tol > 0. Set $l \leftarrow 0$. Choose $\tau := 0.99$, $\mu^0, \mu_{\text{thresh}}, \delta_\mu, \chi_t, \chi_\mu > 0$, $\nu_\mu \geq 1$ and $\nu_\epsilon \in (0,1]$ such that $\nu_\epsilon \geq (1/\nu_\mu)$. Set $t^l \leftarrow \chi_t \mu^l$.

while $\left\| \begin{bmatrix} \nabla_z \mathcal{L}(z^l, \lambda^l) \\ c(z^l; 0) \\ Z^l \lambda_z^l \end{bmatrix} \right\|_\infty > \text{tol}$ **do**

  Set $k \leftarrow 0$, $(z^{l,k}, \lambda^{l,k}) \leftarrow (z^l, \lambda^l)$.

  repeat

    if $\mu^l \leq \mu_{\text{thresh}}$ **then**

      Set $\eta^k \leftarrow 0.1 \frac{\mu^{l-1}}{1+\|\lambda^{l,k}\|_\infty}$.

    else

      Set $\eta^k \leftarrow 0$.

    end if

    Compute $(dz^k, d\lambda^k)$ by solving (3.7) with $(z^{l,k}, \lambda^{l,k}, t^l, \mu^l)$ replacing (z^k, λ^k, t, μ) and $\tilde{z}_i^{l,k}, \tilde{\lambda}_{z,i}^{l,k}$ defined by (ii) in (3.8) in (3.7).

    Calculate $\alpha^{\max} := \max\{\alpha \in (0,1] | \; \alpha dz^k \geq -\tau z^{l,k},\ \alpha d\lambda_z^{l,k} \geq -\tau \lambda_z^{l,k}\}$

    Perform backtracking line search to obtain $\alpha^k \in (0, \alpha^{\max}]$ such that, $(z^{l,k+1}, \lambda^{l,k+1}) \leftarrow (z^{l,k}, \lambda^{l,k}) + \alpha^k (dz^k, d\lambda^k)$, achieves progress in solving barrier problem of (3.1) for (t^l, μ^l) as gauged by a filter [22, 23].

    if line search fails **then**

      Invoke a restoration phase to find $(z^{l,k+1}, \lambda^{l,k+1})$ that decreases infeasibility [22, 23].

    end if

    Set $k \leftarrow k+1$.

  until $\left\| \begin{bmatrix} \nabla_z \mathcal{L}(z^{l,k}, \lambda^{l,k}) \\ c(z^{l,k}; t^l) \\ Z^{l,k} \lambda_z^{l,k} - \mu^l e_n \end{bmatrix} \right\|_\infty \leq \delta_\mu (\mu^l)^{\nu_\epsilon}$

  Set $(z^{l+1}, \lambda^{l+1}) \leftarrow (z^{l,k}, \lambda^{l,k})$, $\mu^{l+1} \leftarrow \chi_\mu (\mu^l)^{\nu_\mu}$, $t^{l+1} \leftarrow \chi_t \mu^{l+1}$, $\tau = \max(\tau, 1 - \mu^{l+1})$ and $l \leftarrow l+1$.

end while

close to a solution, we can fail to find a step size, $\alpha^k > \alpha^{\min}$ that accomplishes this. IPOPT then invokes a restoration phase to locate an iterate that has lower constraint infeasibility. We will not elaborate further on the particular restoration phase algorithm but emphasize that successful termination of the restoration phase algorithm is crucial to the convergence of the algorithm [22, 23]. In our experience, as also noted below, there have been instances in which convergence has been impeded due to failure of the restoration phase.

In the algorithm IPOPT-C , a choice of $\nu_\mu > 1$ may be used to reduce the barrier parameter at a superlinear rate as required by our analysis in the previous section. The analysis in the previous section applies only in the neighborhood of the solution and we use μ_{thresh} as a threshold for μ below which the step is modified. We obtain the original algorithm IPOPT, with no modification of the step, if we set $\eta^k \leftarrow 0$ for all k. For the modification, we have employed (ii) in (3.8). Our experience with other choices for the modification have been similar. In checking for the error in satisfying the NLP stationary conditions for the MPCC we have used $c(\cdot; 0)$ instead of $\tilde{c}(\cdot)$ as these are equivalent when the strict complementarity condition holds for the MPCC

(A3). The modification, η^k has been made proportional to the barrier parameter for the previous subproblem instead of the error ρ and this has worked well in practice. Also, note that the algorithm solves each barrier problem to a tolerance of $\delta_\mu(\mu^l)^{\nu_\epsilon}$ for $\nu_\epsilon < 1$, whenever $\nu_\mu > 1$, instead of $\nu_\epsilon = 1$ as is generally used in NLP interior point methods. This is due to the fact that the algorithm yields an iterate with an error on the order of μ^l and the choice of $\nu_\epsilon < 1$ serves as a safeguard.

5.2. AMPL interface. The algorithm IPOPT has been interfaced to AMPL [8], a modeling language. AMPL provides for easy specification of the complementarity constraint using the *complements* operator [8]. The canonical MPCC that AMPL conveys to the solver is the following form,

$$\begin{array}{ll} \min\limits_{x,y} & f(x,y) \\ \text{s.t.} & g(x,y) \leq 0 \\ & h(x,y) = 0 \\ & l_{\tilde{g},i} \leq \tilde{g}_i(x,y) \leq u_{\tilde{g},i} \perp l_{y,i} \leq y_i \leq u_{y,i} \quad i = 1,\dots,m. \end{array}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m, g : \mathbb{R}^{n+m} \to \mathbb{R}^p$, $\tilde{g} : \mathbb{R}^{n+m} \to \mathbb{R}^m$, $h : \mathbb{R}^{n+m} \to \mathbb{R}^q$. In the above program, for each i exactly one of $\{l_{\tilde{g},i}, u_{\tilde{g},i}\}$ and one of $\{l_{y,i}, u_{y,i}\}$ is finite. The interface will reformulate by addition of slack variables, $s_g \in \mathbb{R}^p, w \in \mathbb{R}^m$ for $g, \tilde{g}$ respectively and $s_{cc} \in \mathbb{R}^m$ for the inequality relaxation of the complementarity constraints to yield the NLP,

$$\begin{array}{ll} \min\limits_{x,w,y,s_g,s_{cc}} & f(x,y) \\ \text{s.t.} & g(x,y) + s_g = 0 \\ & h(x,y) = 0 \\ & \tilde{g}(x,y) - w = 0 \\ & (w_i - l_{\tilde{g},i})(y_i - l_{y,i}) + s_{cc,i} = t \quad i = 1,\dots,m \\ & w_i \geq l_{\tilde{g},i}, y_i \geq l_{y,i} \quad i = 1,\dots,m \\ & s_g, s_{cc} \geq 0. \end{array}$$

for some $t > 0$ approaching zero. A similar reformulation is easily done when we have a few or all of upper bounds $u_{\tilde{g},i}, u_{y,i}$ being finite instead of the lower bounds. Due to the reformulation, IPOPT-C sees a problem with $(n + 3m + p)$ variables, $(p + q + 2m)$ constraints including m complementarity constraints plus $(p+3m)$ bounds. The AMPL interface of IPOPT-C conveys the appropriate function and derivative information to the solver.

5.3. IPOPT-C - Results & Discussion. We present results from solving 136 problems of MacMPEC test suite [12] for different μ/t ratios. The suite contains 140 problems of which we have excluded `ex9.2.2`, `qpec2`, `ralph1` and `scholtes4` as these problems do not possess a strongly stationary point. In the full set, there are a number of problems that do not satisfy strict complementarity or the MPCC-SOSC at the solution and so we opt for a conservative choice of the constants in the algorithm IPOPT-C. In particular we choose the constants as $\mu^0 = 0.1, \chi_\mu = 1, \nu_\mu = 1.3, \delta_\mu = 5$, $\mu_{\text{thresh}} = 5 \cdot 10^{-6}$, $\mathsf{tol} = 5 \cdot 10^{-7}$. The analysis of Wright [26] on effect of machine precision for interior point methods can be readily shown to hold for MPCCs as well. Wright [26] suggests terminating the interior point algorithm at $\mu > \sqrt{\mathbf{u}} = 10^{-8}$, where $\mathbf{u}$ is the floating point machine precision. Due to this and non-satisfaction of Assumptions (A) on some of the MacMPEC problems, we have chosen a higher value ($\mathsf{tol} = 5 \cdot 10^{-7} = 50\mathbf{u}$) for our termination criterion. The initial

TABLE 5.1
Number of MacMPEC problems solved by different μ/t ratios.

$\frac{\mu}{t}$	# solved
10	129
1	130
0.1	128
$\mu^{0.1}$	126

points for all the problems are the ones provided in the AMPL model. The slacks for the complementarity constraints are set to 0.1 while those for the inequalities are set to the evaluation of the inequality constraints. The initial point may be perturbed by IPOPT-C to ensure strict satisfaction of the bounds [22].

We compare the performance on MacMPEC for different μ/t ratios. The number of problems solved by each of the four ratios that we compare are provided in Table 5.1. We do not report the times for each problem but just mention that all the options completed the entire set of 136 problems within 45 CPU minutes on a 2.4 GHz Intel Xeon processor with LINUX as the operating system. The appendix provides complete information on problem size and computational requirements for solving the MacMPEC problems for the choice of $\chi_t = 1$.

We provide a comparison of the different options using the Dolan and Moré [4] performance profiles. Let $t_{pr,s}$ denote a performance characteristic, such as function evaluations required, of an option s in solving a problem pr. Define the ratios

$$r_{pr,s} = \frac{t_{pr,s}}{\min\limits_{s'} t_{pr,s'}}.$$

Then the performance profile for each option s is defined as,

$$\pi_s(\tau) = \frac{\text{no. of problems with } r_{pr,s} \leq \tau}{\text{total no. of problems}}, \quad \tau \geq 0.$$

This performance profile will be used to compare the magnitudes of the multipliers at the solution, function evaluation and iteration count for different options.

Figures 5.1, 5.2 and 5.3 respectively plot the performance profiles in terms of magnitude of multipliers, iterations and function evaluations for different ratios of μ/t. We have considered only 111 problems on which all options converged to a solution with same objective value. The legend in the plots indicate the ratio of μ/t.

Decreasing μ/t may result in convergence to smaller multipliers as indicated by Lemma 2.7. Figure 5.1 confirms this observation. We obtain the smallest multipliers when $\mu/t = 0.1$. On the other hand, smaller multipliers imply that the limiting multipliers are close to ones in $\mathcal{S}_\lambda$ not satisfying strict complementarity. This leads to increased function evaluations and iterations, refer Figures 5.3 and Figure 5.2. It is interesting that even in the case of $t = \mu^{0.9}$, for which the assumption of iterates entering the neighborhood $\mathcal{N}_\gamma$ may fail to hold, we can solve most of the problems. The results indicate that the choice of $t = \chi_t \mu$ with $\chi_t \geq 1$ should be fairly robust.

5.4. Discussion on IPOPT-C failures. The algorithm can fail to converge to a solution due to one or more of the following reasons,

1. *Failure of Assumption* (A2) or (A3) - This can occur if the limit point does not satisfy strong stationarity, MPCC-LICQ, MPCC-SOSC or strict complementarity. This often results in unbounded multipliers and breakdown of the algorithm.

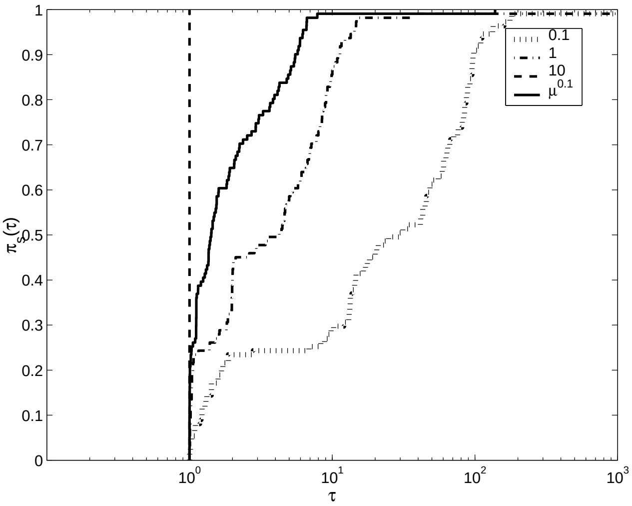

FIG. 5.1. *Performance of IPOPT-C in terms of* $\|(\lambda_w, \lambda_y, \lambda_{cc})\|_1$ *for different* μ/t *on the MacMPEC suite.*

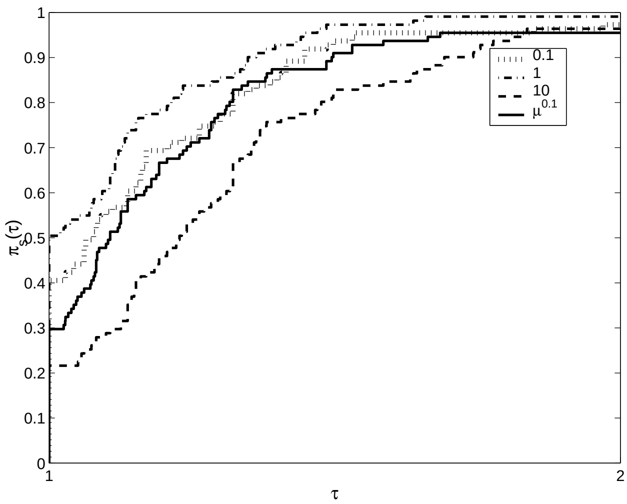

FIG. 5.2. *Performance of IPOPT-C in terms of iterations for different* μ/t *on the MacMPEC suite.*

2. *Failure of restoration phase* - As mentioned previously, IPOPT employs a filter with a backtracking line search procedure to aid convergence from poor initial points. If the algorithm does not make sufficient progress then, it invokes a restoration phase algorithm which attempts to locate a point with lower infeasibility. Restoration can be invoked arbitrarily close to the solution and if unsuccessful can prevent convergence of the algorithm.

The options presented in Table 5.1 all fail on `siouxfls1`, `water-FL` and `water-net`. On these problems, the algorithm terminates without solving even the initial few barrier problems, attributable to failure mode (2).

The failure of Assumption (A2) does not automatically imply failure of the algorithm. The non-satisfaction of MPCC-LICQ or strict complementarity is not as

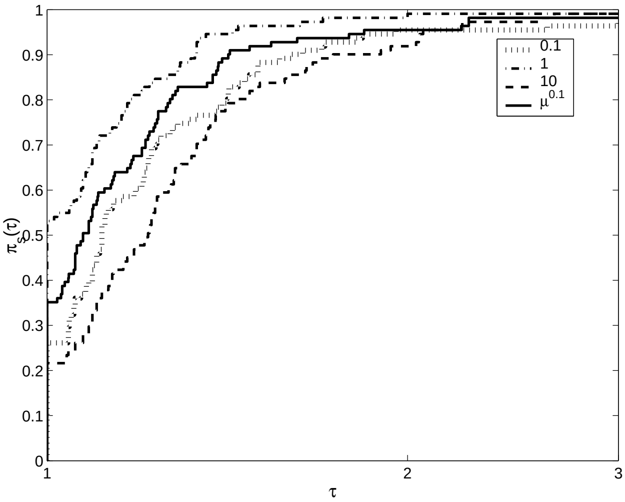

FIG. 5.3. *Performance of IPOPT-C in terms of function evaluations for different μ/t on the MacMPEC suite.*

severe as failure of strong stationarity. We have observed convergence, albeit at a linear rate, even when strict complementarity does not hold, for example `bard1`, `bard3` etc. In some cases there is a combined effect of failure of strict complementarity and an unsuccessful restoration phase that have caused failure. The analysis of the algorithm in §4 can be readily applied for the case where MPCC-MFCQ holds in place of MPCC-LICQ provided that we generate iterates that enter $\mathcal{N}_\gamma$. We have observed this on `scholtes5` for which only the MPCC-MFCQ holds.

Convergence of algorithms depend critically on both effective modeling as well as algorithmic robustness. The preceding paragraph earmarked scope for improvement of the algorithm. However, there are certain problem formulations in the MacMPEC test suite for which interior point methods do not seem a natural choice. In certain cases, there exists a reliable work around that can redress it. The problem `qpec2` is a fit case. The presence of constraints such as,

$$0 \leq y \perp y \geq 0$$

renders a MPCC whose limit does not satisfy MPCC-LICQ although MPCC-MFCQ may hold. Replacing this constraint with $y = 0$ results in the limit satisfying Assumptions (A) and the algorithm converges. An effectual redress does not exist in the optimum packaging problems, `pack-comp1-*`, `pack-comp2-*`, `packcomp1c-*`, `pack-comp2c-*`, `pack-rig1-*`, `pack-rig2-*`, `pack-rig1c-*` and `pack-rig2c-*`. The problems possess constraints other than complementarity constraints that do not have a strict interior rendering the interior point algorithm an ineffectual choice. A reformulation has been proposed in the form of penalization of some of the inequalities in the objective so as to present a problem with a strict interior. The reformulated problems are named, `pack-comp1p-*`, `pack-comp2p-*`, `pack-comp1cp-*`, `pack-comp2cp-*`, `pack-rig1p-*`, `pack-rig2p-*`, `pack-rig1cp-*` and `pack-rig2cp-*`.

5.5. Comparison with LOQO and filtermpec. Finally, we compare our algorithm to the approaches of Benson *et al.* [2] and Fletcher *et al.* [6]. Benson *et al.* [2] employ an interior point algorithm, LOQO [20], with no reformulation of the

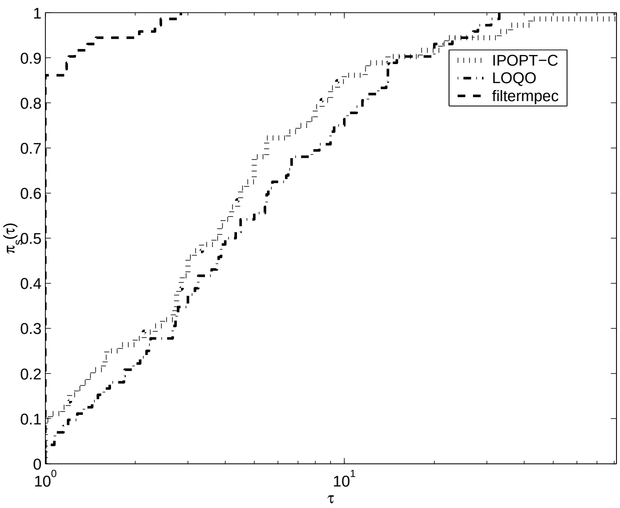

FIG. 5.4. *Performance of IPOPT-C, LOQO and filtermpec on MacMPEC in terms of number of iterations taken.*

complementarity constraints to solve MPECs. They also suggest heuristics to handle degeneracy in the complementarity constraints and provide results for their approach. Fletcher *et al.* [6] present results from using an SQP algorithm, filtermpec to solve MPECs. Figure 5.4 compares the iteration count of filtermpec [6], LOQO [2] and IPOPT-C. The comparison considers only 77 of the MacMPEC problems on which all the algorithms converged to the same objective value. The results for LOQO and filtermpec are obtained from the respective papers. We also acknowledge the fact that results for the algorithms vary in the tolerances but we can draw some general conclusions from the results.

The plot in Figure 5.4 indicates that filtermpec outperforms the interior point algorithms when comparing iterations. SQP algorithms generally require fewer iterations than interior point algorithms especially when NLPs have few inequalities. Further, SQP algorithms are more robust to failure of strict complementarity assumption (A3) than interior point algorithms. We present iteration counts of the algorithms on a few representative problems in Table 5.2. Problems `bard1`, `bard2`, `jr1`, `jr2` and `nash1` have linear equality, inequality constraints and quadratic objective. Table 5.2 indicates that filtermpec performs much better than the interior point algorithms on the above problems. The remaining problems possess larger number of complementarity constraints and as a result more bounds. On these problems, filtermpec is sometimes better, sometimes worse than IPOPT-C and LOQO. A more interesting comparison would be the solution times of the algorithms on the larger problems. Our observation has been that interior point algorithms are faster than SQP algorithms when the number of complementarity constraints increases. The solution time for SQP algorithms can be larger due to the combinatorially intensive step of identifying the active-set. The results in Fletcher *et al.* [6] also reveal that filtermpec requires more CPU time than LOQO. We avoid comparing solution times as the papers [2, 6] do not provide the specifications on the processor speed and operating system on which results have been obtained.

Table 5.2
Iteration summary of IPOPT-C, LOQO and filtermpec on representative MacMPEC problems.

	n	$p+q$	m	IPOPT-C	LOQO	filtermpec
bard1	5	1	3	11	14	1
bard2	12	5	4	23	34	2
jr1	2	1	1	8	9	1
jr2	2	1	1	8	11	6
nash1	6	2	2	10	14	1
incid-set1c-16	485	491	225	49	75	108
incid-set1c-32	1989	2003	961	60	164	41
incid-set2-16	485	491	225	54	59	69
incid-set2-32	1989	2003	961	93	151	39
incid-set2c-16	485	506	225	34	59	66
incid-set2c-32	1989	2034	961	57	456	33

6. Conclusions. For the convergence of the algorithm, the assumption that the iterates eventually enter the neighborhood of a solution satisfying strict complementarity, $\mathcal{N}_\gamma(\epsilon)$ is critical. This assumption is also made by Vicente and Wright [21] for interior point methods and Wright [25] for SQP methods for the solution of degenerate nonlinear programs satisfying MFCQ. Such an assumption, as also noted by these authors, cannot be guaranteed by their algorithms. Wright [27] has outlined an algorithm for identifying the strongly active indices and ensuring that the iterates enter a neighborhood, $\mathcal{N}_\gamma(\epsilon)$ if it exists. This requires the solution of linear programs. In contrast, the assumption that our iterates enter the neighborhood, $\mathcal{N}_\gamma(\epsilon)$ can be easily satisfied by appropriate choice of the complementarity relaxation parameter, t and the barrier parameter, μ under the Assumptions (A2)-(A3). Numerical experience also confirms the above observation. The convergence analysis of our algorithm will continue to hold if the MPCC-LICQ is relaxed to MPCC-MFCQ but will need an algorithm such as that of Wright [27] to guarantee that iterates enter $\mathcal{N}_\gamma(\epsilon)$.

Global convergence and improving robustness of the restoration phase algorithm will be investigated in the future. It remains to be seen if a mechanism for ensuring bounded multipliers can be developed. Additionally, there are a number of MacMPEC problems on which our algorithm converges to solutions not satisfying some of Assumptions (A). The numerical behavior on these problems merits special attention.

Appendix. Summary of Numerical results.

This section lists the problem size and performance of algorithm, IPOPT-C on all MacMPEC problems. The results provided correspond to choosing $\chi_t = \mu/t = 1$ and all other constants as described in §5.3. The columns of the table correspond to name of the problem (Name), number of variables (# var), number of constraints including complementarity constraints (# con), number of complementarity constraints (# ccon), number of iterations (# it), number of function evaluations (# fe), CPU seconds required for completion (CPU sec.) and objective value at convergence (Objective). The number of variables and constraints are the ones obtained after introduction of slacks by the AMPL interface as described in §5.2. For a few problems the table lists IL in the objective column indicating that the problem was not solved within the iteration limit of 1000 iterations. The problem tap-09 has RF in the objective column to denote that the restoration phase failure caused termination of the algorithm. The results have been obtained on a 2.4 GHz Intel Xeon processor running LINUX as the operating system.

Name	# var	# con	# ccon	# it	# fe	CPU sec.	Objective
bar-truss-3	47	40	6	17	18	0.01	1.01666e+04
bard1	11	7	3	11	12	0.01	1.70000e+01
bard2	19	12	3	23	24	0.01	6.59800e+03
bard3	9	6	1	20	21	0.01	-1.26787e+01
bard1m	12	7	3	21	26	0.01	1.70000e+01
bard2m	19	12	3	23	24	0.01	-6.59800e+03
bard3m	13	8	3	20	21	0.01	-1.26787e+01
bem-milanc30-s	6365	4897	1464	1000	11836	243.68	IL
bilevel1	23	15	6	14	15	0.01	5.00000e+00
bilevel2	33	21	8	33	34	0.02	-6.60000e+03
bilevel3	18	13	3	22	23	0.01	-1.26787e+01
bilin	21	13	6	10	11	0	1.30000e+01
dempe	5	3	1	51	156	0.01	2.82503e+01
design-cent-1	20	14	3	8	9	0	1.86065e+00
design-cent-2	25	18	3	9	10	0	3.48382e+00
design-cent-3	23	14	3	17	19	0.02	3.72337e+00
design-cent-4	40	28	8	13	14	0.01	3.07920e+00
desilva	10	6	2	17	18	0	-9.99999e-01
df1	6	4	1	14	15	0.01	6.78632e-08
ex9.1.1	23	17	5	15	16	0	-1.30000e+01
ex9.1.2	12	9	2	9	10	0	-6.25000e+00
ex9.1.3	35	27	6	22	24	0.01	-2.92000e+01
ex9.1.4	12	9	2	20	26	0	-3.70000e+01
ex9.1.5	23	17	5	13	14	0.01	-1.00000e-00
ex9.1.6	26	19	6	18	19	0.01	-2.10000e+01
ex9.1.7	29	21	6	36	49	0.01	-2.60000e+01
ex9.1.8	14	10	2	11	12	0.01	-3.25000e+00
ex9.1.9	22	16	5	19	26	0.01	3.11111e+00
ex9.1.10	14	10	2	11	12	0	-3.25000e+00
ex9.2.1	18	13	4	18	19	0.01	1.70000e+01
ex9.2.3	23	17	4	14	16	0	5.00000e+00
ex9.2.4	12	9	2	14	15	0.01	5.00000e-01
ex9.2.5	14	10	3	17	18	0.01	5.00000e+00
ex9.2.6	28	18	6	14	15	0.01	-9.99999e-01
ex9.2.7	18	13	4	18	19	0	1.70000e+01
ex9.2.8	10	7	2	10	12	0	1.50000e+00
ex9.2.9	15	11	3	14	15	0.01	2.00000e+00
flp2	8	4	2	16	17	0	7.82036e-10
flp4-1	170	90	30	27	28	0.16	1.55380e-06
flp4-2	280	170	60	28	29	0.39	2.15161e-08
flp4-3	380	240	70	27	28	1.76	2.67991e-06
flp4-4	550	350	100	29	30	2.62	4.56477e-06
gauvin	7	4	2	12	15	0.01	2.00000e+01
gnash10	29	20	8	17	18	0.01	-2.30823e+02
gnash11	29	20	8	19	20	0.01	-1.29912e+02
gnash12	29	20	8	18	19	0.01	-3.69331e+01
gnash13	29	20	8	17	18	0	-7.06178e+00
gnash14	29	20	8	17	18	0	-1.79046e-01
gnash15	29	20	8	23	24	0.01	-3.54699e+02
gnash16	29	20	8	21	24	0	-2.41442e+02
gnash17	29	20	8	27	28	0.01	-9.07491e+01
gnash18	29	20	8	26	30	0.01	-2.56982e+01
gnash19	29	20	8	16	17	0	-6.11671e+00
hakonsen	18	12	4	17	19	0.01	2.43668e+01
hs044-i	40	24	10	20	22	0.01	1.56178e+01
incid-set1-08	236	168	49	24	28	0.23	6.84555e-06
incid-set1-16	976	716	225	43	47	2.16	3.13668e-05
incid-set1-32	3992	2964	961	145	286	54.83	1.35960e-04
incid-set1c-08	243	175	49	25	28	0.21	6.85070e-06
incid-set1c-16	991	731	225	49	61	4.91	3.70241e-05
incid-set1c-32	4023	2995	961	60	92	19.2	1.51486e-04

Name	# var	# con	# ccon	# it	# fe	CPU sec.	Objective
incid-set2-08	236	168	49	32	41	0.34	4.52651e-03
incid-set2-16	976	716	225	54	67	2.86	3.03272e-03
incid-set2-32	3992	2964	961	93	120	30.45	1.86717e-03
incid-set2c-08	243	175	49	25	27	0.23	5.48088e-03
incid-set2c-16	991	731	225	34	36	1.52	3.63571e-03
incid-set2c-32	4023	2995	961	57	65	14.79	2.57590e-03
jr1	4	2	1	8	9	0.01	5.00000e-01
jr2	4	2	1	8	9	0	5.00000e-01
kth1	4	2	1	10	11	0	2.83376e-07
kth2	4	2	1	9	10	0	5.38925e-08
kth3	4	2	1	8	9	0	5.00000e-01
liswet1-050	253	153	50	38	39	0.06	1.40025e-02
liswet1-100	503	303	100	51	52	0.15	1.37495e-02
liswet1-200	1003	603	200	74	75	0.44	1.70386e-02
nash11	10	6	2	10	11	0.01	7.43400e-15
nash12	10	6	2	10	11	0.01	8.01999e-15
nash13	10	6	2	10	11	0.01	8.20657e-15
nash14	10	6	2	18	19	0.01	2.75803e-09
nash15	10	6	2	10	11	0	8.12723e-15
outrata31	13	8	4	20	24	0	3.20770e+00
outrata32	13	8	4	18	19	0	3.44940e+00
outrata33	13	8	4	16	17	0	4.60425e+00
outrata34	13	8	4	16	17	0.01	6.59268e+00
pack-comp1cp-08	220	162	49	59	63	0.31	6.00002e-01
pack-comp1cp-16	948	706	225	44	46	1.31	6.23065e-01
pack-comp1cp-32	3940	2946	961	50	51	8.28	6.61555e-01
pack-comp1p-08	213	155	49	57	72	0.3	6.00005e-01
pack-comp1p-16	933	691	225	39	47	1.33	6.16980e-01
pack-comp1p-32	3909	2915	961	57	60	11.54	6.53089e-01
pack-comp2cp-08	220	162	49	80	97	0.45	6.73074e-01
pack-comp2cp-16	948	706	225	32	34	122.85	7.27545e-01
pack-comp2cp-32	3940	2946	961	51	55	9.52	7.83058e-01
pack-comp2p-08	213	155	49	49	52	0.28	6.72532e-01
pack-comp2p-16	933	691	225	39	41	1.81	7.27276e-01
pack-comp2p-32	3909	2915	961	42	48	10.22	7.82717e-01
pack-rig1cp-08	214	158	47	29	31	0.13	7.88300e-01
pack-rig1cp-16	882	662	203	35	36	0.96	8.26530e-01
pack-rig1cp-32	3640	2746	861	48	49	6.47	8.51760e-01
pack-rig1p-08	207	151	47	27	29	0.12	7.87932e-01
pack-rig1p-16	867	647	203	33	34	1.1	8.26044e-01
pack-rig1p-32	3609	2715	861	53	60	502.22	8.51010e-01
pack-rig2cp-08	208	154	45	27	29	0.11	7.99314e-01
pack-rig2cp-16	855	644	194	40	43	1.14	9.49266e-01
pack-rig2cp-32	3499	2652	814	62	64	9.34	1.30423e+00
pack-rig2p-08	201	147	45	28	29	0.14	7.80413e-01
pack-rig2p-16	840	629	194	49	50	1.33	1.08517e+00
pack-rig2p-32	3468	2621	814	57	58	8.23	1.13600e+00
portfl-i-1	111	37	12	24	25	0.05	1.90844e-05
portfl-i-2	111	37	12	22	23	0.04	1.89189e-05
portfl-i-3	111	37	12	21	22	0.04	1.14944e-05
portfl-i-4	111	37	12	23	24	0.06	6.75604e-06
portfl-i-6	111	37	12	22	23	0.03	2.68413e-06
qpec-100-1	307	202	100	24	25	1.1	1.09237e-01
qpec-100-2	312	202	100	45	50	2.01	-6.44521e+00
qpec-100-3	314	204	100	28	29	1	-5.48288e+00
qpec-100-4	324	204	100	25	27	0.99	-3.98212e+00
qpec1	70	40	20	10	11	0.01	8.00000e+01
qpec-200-1	614	404	200	38	44	14.68	-1.93490e+00
qpec-200-2	624	404	200	44	46	16.33	-2.40774e+01
qpec-200-3	628	408	200	1000	11295	292.23	IL
qpec-200-4	648	408	200	27	34	12.15	-6.23052e+00

Name	# var	# con	# ccon	# it	# fe	CPU sec.	Objective
ralph2	4	2	1	14	15	0	-6.07653e-07
ralphmod	304	200	100	85	91	3.23	-6.83033e+02
scholtes1	5	2	1	14	17	0	2.00000e+00
scholtes2	5	2	1	15	18	0.01	1.50000e+01
scholtes3	4	2	1	12	14	0	5.00000e-01
scholtes5	7	4	2	8	9	0	1.00000e+00
siouxfls	5899	4124	1748	655	702	397.04	2.08258e+02
siouxfls1	5899	4124	1748	1000	7200	342.74	IL
sl1	14	8	3	26	27	0.01	1.00296e-04
stackelberg1	5	3	1	11	12	0.01	-3.26667e+03
tap-09	154	100	32	148	207	0.75	RF
tap-15	376	250	83	631	684	11.08	2.53404e+02
water-FL	301	204	44	1000	7797	5.44	IL
water-net	94	64	14	1000	9509	1.89	IL

Acknowledgment. The authors are grateful to Andreas Wächter for help in implementing the step modification within the algorithm, IPOPT and for many interesting discussions which helped improve the performance of the proposed algorithm, IPOPT-C . We are also thankful to Mihai Anitescu and Steve Wright for careful reading of an earlier version of the manuscript. Finally, we gratefully acknowledge two anonymous referees whose suggestions helped improve the presentation and also strengthen the paper.

REFERENCES

[1] M. Anitescu, *On solving mathematical programs with complementarity constraints as nonlinear programs*, Tech. Rep. ANL/MCS-P864-1200, Argonne National Laboratories, Argonne, Illinois, 60439., 2000.

[2] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming : complementarity constraints*, Tech. Rep. Research Report ORFE-02-02, Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2002.

[3] A. V. De Miguel, M. P. Freidlander, F. J. Nogales, and S. Scholtes, *An interior-point method for mathematical programs with equilibrium constraints.* LBS Working Paper, 2003.

[4] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with COPS*, Mathematical Programming, 91 (2002), pp. 201–213.

[5] F. Facchinei, H. Jiang, and L. Qi, *A smoothing method for mathematical programs with equilibrium constraints*, Mathematical Programming, 85 (1999), pp. 107–134.

[6] R. Fletcher and S. Leyffer, *Numerical experience with solving MPECs as NLPs*, Tech. Rep. Numerical Analysis Report NA/210, Department of Mathematics, University of Dundee, Dundee, UK, 2002.

[7] R. Fletcher, S. Leyffer, S. Scholtes, and D. Ralph, *Local convergence of SQP methods for mathematical programs with equilibrium constraints*, Tech. Rep. Numerical Analysis Report NA/183, Department of Mathematics, University of Dundee, Dundee, UK, 2002.

[8] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL A Modeling Language for Mathematical Programming*, Thomson Brooks/Cole, second ed., 2003.

[9] W. W. Hager, *Stabilized sequential quadratic programming*, Computational Optimization and Applications, 85 (1999), pp. 181–192.

[10] W. W. Hager and M. S. Gowda, *Stability in the presence of degeneracy and error estimation*, Mathematical Programming, 85 (1999), pp. 181–192.

[11] H. Jiang and D. Ralph, *Smooth SQP methods for mathematical programs with nonlinear complementarity problems*, SIAM Journal on Optimization, 10 (2000), pp. 779–808.

[12] S. Leyffer, *MacMPEC - www.unix.mcs.anl.gov/leyffer/macmpec/*, 2002.

[13] X. Liu and J. Sun, *Generalized stationary points and an interior point method for mathematical programs with equilibrium constraints*, tech. rep., Department of Decision Sciences and Singapore-MIT Alliance, Natinal University of Singapore, Republic of Singapore, 2002.

[14] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints (MPECs)*, Cambridge, UK : Cambridge University Press, 1996.

[15] ———, *Piecewise sequential quadratic programming for mathematical programs with nonlinear complementarity constraints*, in Multilevel Optimization : Algorithms and Applica-

tions, A. Migdalas, P. Pardalos, and P. Värbrand, eds., Kluwer Academic Publishers, 1998, pp. 209–229.

[16] J. Outrata, M. Kočvara, and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Cconstraints : Theory, Applications and Numerical results*, Kluwer Academic Publishers, Dordrecht-Boston-London, 1998.

[17] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints : Stationarity, optimality and sensitivity*, Mathematics of Operations Research, 23 (2000), pp. 1–22.

[18] H. Scheel and S. Stöhr, *Exact penalization of mathematical programs with equilibrium constraints*, SIAM Journal on Control and Optimization, 37 (1999), pp. 617–652.

[19] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM Journal on Optimization, 11 (2001), pp. 918–936.

[20] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications, 13 (1999), pp. 231–252.

[21] L. N. Vicente and S. Wright, *Local convergence of a primal-dual method for degenerate nonlinear programming*, Computational Optimization and Applications, 22 (2002), pp. 311–328.

[22] A. Wächter, *Interior Point Methods for Large-Scale Nonlinear Programming with Applications*, PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 2002.

[23] A. Wächter and L. T. Biegler, *Global and local convergence of line search filter methods for nonlinear programming*, tech. rep., Dept. of Chemical Engineering, Canegie Mellon University, Pittsburgh, Pennsylvania. USA, 2002.

[24] M. H. Wright, *Interior point methods for constrained optimization*, Acta Numerica, 1 (1992), pp. 341–407.

[25] S. J. Wright, *Superlinear convergence of a stabilized SQP method to a degenerate solution*, Computational Optimization and Applications, 11 (1998), pp. 253–275.

[26] ———, *Effect of finite precision arithmetic on interior-point methods for nonlinear programming*, Tech. Rep. Preprint ANL/MCS-P705-0198, Argonne National Laboratory, January 1998. Revised May 2000 2000. To appear in SIAM J. Optimization.

[27] ———, *Constraint identification and algorithm stabilization for degenerate nonlinear programming*, Tech. Rep. ANL-MCS-P865-1200, Argonne National Laboratory, Revised December 2001. To appear in Mathematical Programming, Series B.