Optimization Methods and Software

ISSN: 1055-6788 (Print) 1029-4937 (Online) Journal homepage: http://www.tandfonline.com/loi/goms20

Convex and concave relaxations of implicit functions

Matthew D. Stuber, Joseph K. Scott & Paul I. Barton

To cite this article: Matthew D. Stuber, Joseph K. Scott & Paul I. Barton (2015) Convex and concave relaxations of implicit functions, Optimization Methods and Software, 30:3, 424-460, DOI: 10.1080/10556788.2014.924514

To link to this article: http://dx.doi.org/10.1080/10556788.2014.924514

Accepted author version posted online: 20 May 2014.
Published online: 17 Jun 2014.

Optimization Methods & Software, 2015
Vol. 30, No. 3, 424–460, http://dx.doi.org/10.1080/10556788.2014.924514

Taylor & Francis
Taylor & Francis Group

Convex and concave relaxations of implicit functions

Matthew D. Stuber, Joseph K. Scott and Paul I. Barton*

Process Systems Engineering Laboratory, Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Ave., Bldg. 66-464, Cambridge 02139, MA, USA

(*Received 28 June 2013; accepted 3 May 2014*)

A deterministic algorithm for solving nonconvex NLPs globally using a reduced-space approach is presented. These problems are encountered when real-world models are involved as nonlinear equality constraints and the decision variables include the state variables of the system. By solving the model equations for the dependent (state) variables as implicit functions of the independent (decision) variables, a significant reduction in dimensionality can be obtained. As a result, the inequality constraints and objective function are implicit functions of the independent variables, which can be estimated via a fixed-point iteration. Relying on the recently developed ideas of generalized McCormick relaxations and McCormick-based relaxations of algorithms and subgradient propagation, the development of McCormick relaxations of implicit functions is presented. Using these ideas, the reduced space, implicit optimization formulation can be relaxed. When applied within a branch-and-bound framework, finite convergence to ϵ-optimal global solutions is guaranteed.

Keywords: global optimization; McCormick relaxations; nonconvex programming

AMS Subject Classifications: 65K05; 65H10; 90C26

1. Introduction

Nonconvex nonlinear programs (NLPs) of the form:

$$\begin{aligned} \min_{\mathbf{y}\in Y\subset\mathbb{R}^{n_y}} \quad & f(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{y}) = \mathbf{0} \end{aligned} \tag{1}$$

can be formulated to solve a wide variety of problems in diverse disciplines ranging from operations research to engineering design. Local algorithms, such as sequential quadratic programming (SQP) [7], are not guaranteed to find the desired global optima. Thus deterministic global optimization algorithms such as branch-and-bound (B&B) [6] and branch-and-reduce [30] have been developed. However, all currently known deterministic global optimization algorithms suffer from worst-case exponential run time. Therefore, if the original program (1) can be reformulated as an equivalent program with reduced dimensionality, there is potential for a significant reduction in computational cost. The primary framework of the algorithm presented in this paper is based on the B&B algorithm.

*Corresponding author. Email: pib@mit.edu

'Selective branching' strategies (i.e. where only a subset of the variables are branched on) have been developed to reduce the run time. The works [13,14,23,26] all require very special problem structures that can be exploited to reduce the number of variables that are branched on. A more general reduced-space B&B approach was first introduced in [4], which builds on the previous selective branching ideas. In the work of Epperly and Pistikopoulos [4], the variables $\mathbf{y}$ are partitioned into two sets of variables and the nonconvex functions are factored according to which set of variables the factors are dependent upon. It is required that all functions of the first set of variables are convex and all functions of the second set of variables are continuous. Under some other assumptions, convergence of the B&B algorithm is guaranteed while only branching on the second set of variables. The authors of Epperly and Pistikopoulos [4] state that this type of factorization and partitioning is applicable to most practical problems. However, the method was developed largely with inequality constraints in mind. The authors of Epperly and Pistikopoulos [4] state that equality constraints can be handled using a pair of opposing inequality constraints. However, given the requirements of their algorithm for selective branching, it can be shown that this restricts the equality constraints that can be handled to parametric linear systems (Section 3.3). Therefore, general nonlinear systems of equations cannot be addressed. Furthermore, in [19], the authors compared their method of relaxing implicit functions with selective branching and experienced a significant performance benefit from relaxing implicit functions.

Consider the equality constraints of (1) as the system of equations:

$$\mathbf{h}(\mathbf{y}) = \mathbf{0}, \tag{2}$$

where $\mathbf{h} : D_y \to \mathbb{R}^{n_x}$ is continuously differentiable, with $D_y \subset \mathbb{R}^{n_y}$ open. Here, it is assumed that the vector $\mathbf{y} \in D_y$ can be separated into *dependent* and *independent* variables $\mathbf{z} \in \mathbb{R}^{n_x}$ and $\mathbf{p} \in \mathbb{R}^{n_p}$, respectively, with $\mathbf{y} = (\mathbf{z}, \mathbf{p})$ such that $\mathbf{h}$ can be solved for $\mathbf{z}$ in terms of $\mathbf{p}$, with $(\mathbf{z},\mathbf{p}) \in D_y$. (2) can then be written as

$$\mathbf{h}(\mathbf{z}, \mathbf{p}) = \mathbf{0}. \tag{3}$$

If, for some n_p-dimensional interval $P \subset \mathbb{R}^{n_p}$, such $\mathbf{z}$ exist that satisfy (3) at each $\mathbf{p} \in P$, then they define an implicit function of $\mathbf{p}$, that will be expressed as $\mathbf{x} : P \to \mathbb{R}^{n_x}$. Such a partition of the vector $\mathbf{y}$ is valid, and even natural, for many practical 'real-world' problems. For instance, consider $\mathbf{h}$ as a steady-state model of a chemical process. The variables $\mathbf{z}$ would correspond to the process state variables and $\mathbf{p}$ would correspond to the model parameters. Unless otherwise stated, it will be assumed that for some $X \subset \mathbb{R}^{n_x}$, there exists at least one continuously differentiable implicit function $\mathbf{x} : P \to X$ such that $\mathbf{h}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$ holds for every $\mathbf{p} \in P$. Conditions under which $\mathbf{x}$ is unique in X are given by the so-called *semilocal implicit function theorem* [24]. Continuous differentiability follows from the same result.

Just as $\mathbf{y}$ was partitioned into $(\mathbf{z}, \mathbf{p})$, the search space of the optimization problem (1) is partitioned as $Y = X \times P$. The program (1) may then be reformulated as the following program:

$$\begin{aligned} \min_{\mathbf{p}\in P} \quad & f(\mathbf{x}(\mathbf{p}), \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}(\mathbf{p}), \mathbf{p}) \leq \mathbf{0}. \end{aligned} \tag{4}$$

It can readily be deduced that if $n_y - n_x$ is small ($n_x >> n_p$), the formulation (4) offers a significant reduction in dimensionality. Contrasting the selective branching works, no structural assumptions have been made nor are required beyond the existence of an implicit function.

In order to solve (4) to global optimality with B&B, a method for calculating convex relaxations of $f(\mathbf{x}(\cdot), \cdot)$ and $\mathbf{g}(\mathbf{x}(\cdot), \cdot)$ on P is required. The major complication is that $\mathbf{x}$ is not known explicitly and may not even have a closed algebraic form, but can only be approximated using a fixed-point algorithm, for instance. Thus, the objective function, $f(\mathbf{x}(\cdot), \cdot)$, and the inequality constraint(s),

$\mathbf{g}(\mathbf{x}(\cdot),\cdot)$, are implicitly defined and must be evaluated with embedded fixed-point iterations. Because of this, the involved functions no longer have a *factorable representation*. Therefore relaxation techniques that rely on explicit algebraic and/or factorable functions, such as standard McCormick [18] relaxations or αBB [1], are no longer applicable. However, if relaxations of the implicit function $\mathbf{x}$ were made available by some method, the functions f and $\mathbf{g}$ could be composed with them, using a generalization of the ideas of McCormick [31], and relaxations of f and $\mathbf{g}$ could be calculated.

In [19], Mitsos and coworkers laid the foundations for calculating relaxations of implicit functions $\mathbf{x}$ evaluated by an algorithm with a fixed number of iterations known a priori. They outline the automatic construction of McCormick convex/concave relaxations of factorable functions and automatic subgradient calculation. The automatic construction of McCormick relaxations and subgradient calculation was done using `libMC`, a predecessor of the currently available `C++` library `MC++` [3]. The types of algorithms considered in their work, however, only included algorithms in which the number of iterations is known a priori, such as Gauss elimination, thus their methods are not applicable to problems in which $\mathbf{x}$ is evaluated by more general fixed-point algorithms, such as Newton's method.

In [31], the authors present the *generalized* McCormick relaxations. This generalization of McCormick relaxations allows for the application of McCormick relaxations to a much broader class of functions, such as those defined by iterative algorithms [31]. Beyond the important theoretical results, the generalized formulation has the important property that they may take previously known or calculated relaxations as arguments, say for further refinement. One focus in [31], that is of interest here, was on the relaxation of the successive-substitution fixed-point iteration. In relaxing the fixed-point iteration, the authors show that relaxations of the sequence of approximations of $\mathbf{x}$ could be calculated [31]. However, in order to relax f and $\mathbf{g}$ rigorously, valid relaxations of $\mathbf{x}$ are required, not of approximations of $\mathbf{x}$. This will be the primary focus of the new theoretical developments contained in this paper.

In the next section, the necessary background information, including interval analysis, fixed-point iterations, and McCormick's relaxations and subgradients, will be discussed. In Section 3, new ideas and results involved in relaxing implicit functions are presented, followed by the global optimization algorithm in Section 4. It should be noted that the theoretical developments in these sections assume that for some interval X, the implicit function $\mathbf{x} : P \to X$ is unique. In general, there are multiple implicit functions that are solutions of (3). Details on the uniqueness assumption and how multiple solution branches can be handled are discussed in Section 4.

2. Background

This section contains the definitions and previously developed material from the literature required for the development of global optimization of implicit functions.

2.1 *Fixed-point iterations*

The term *fixed-point iteration* applies to a general class of iterative methods, for which the iteration count required to satisfy a given convergence tolerance is not typically known a priori. They are commonly employed to solve systems of equations such as (2). The general ideas are introduced here. For the focus of this paper, fixed-point iterations will be used to evaluate the embedded implicit functions in (4). For a more in-depth look at these iterative methods, the reader is directed to [25].

DEFINITION 2.1 (Fixed-Point) *Let* $\mathbf{f} : Z \subset \mathbb{R}^m \to \mathbb{R}^m$. *A point* $\mathbf{z} \in Z$ *is a* fixed point *of* $\mathbf{f}$ *if* $\mathbf{z} = \mathbf{f}(\mathbf{z})$.

An iteration will be referred to as a *fixed-point iteration* if it takes the form

$$\mathbf{z}^{k+1} := \boldsymbol{\phi}(\mathbf{z}^k), \quad k \in \mathbb{N},$$

with $\boldsymbol{\phi} : A \subset Z \to \mathbb{R}^m$. The name suggests that the iteration will be used to find a fixed-point of $\boldsymbol{\phi}$. However, this is ambitious in the sense that these iterations are not guaranteed to do so except under certain conditions. One such condition is if $\boldsymbol{\phi}$ is a *contraction mapping*.

DEFINITION 2.2 (Contraction mapping [29]) *Let* Z *be a metric space with metric* d. *A function* $\boldsymbol{\phi} : A \subset Z \to Z$ *is said to be a* contraction mapping *or contractive on a set* $B \subset A$ *if* $\boldsymbol{\phi}(B) \subset B$ *and there exists an* $\alpha \in (0, 1)$ *such that*

$$d(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{y})) \leq \alpha d(\mathbf{x}, \mathbf{y}), \quad \forall\, \mathbf{x}, \mathbf{y} \in B.$$

DEFINITION 2.3 ($\mathbf{J_x}, \boldsymbol{\nabla}_\mathbf{x}$) *Let* $A \subset \mathbb{R}^m$ *and* $B \subset \mathbb{R}^n$ *be open. Suppose* $\mathbf{h} : A \times B \to \mathbb{R}^m$ *is differentiable on* $A \times B$. *Then for each* $\mathbf{b} \in B$, *let* $\mathbf{J}_\mathbf{x}\mathbf{h}(\mathbf{z}, \mathbf{b})$ *denote the* $m \times m$ *Jacobian matrix of* $\mathbf{h}(\cdot, \mathbf{b})$ *evaluated at* $\mathbf{z} \in A$. *Similarly,* $\boldsymbol{\nabla}_\mathbf{x} h_i(\mathbf{z}, \mathbf{b})$ *denotes the* $m \times 1$ *gradient vector of* $h_i(\cdot, \mathbf{b})$ *evaluated at* $\mathbf{z} \in A$.

Newton-type methods for (2) are based on the form $\mathbf{z} := \boldsymbol{\phi}(\mathbf{z}) = \mathbf{z} - \mathbf{Y}(\mathbf{z})\mathbf{h}(\mathbf{z})$, where it is not guaranteed that $\boldsymbol{\phi}$ is contractive on *any* set. Taking $\mathbf{Y}(\mathbf{z})$ to be the inverse of the (nonsingular) Jacobian matrix $\mathbf{J}_\mathbf{x}\mathbf{h}$ evaluated at the current iterate $\mathbf{z}^k$ gives the standard Newton's method, which under mild assumptions is guaranteed to be contractive in a neighbourhood of an isolated solution. Likewise, taking $\mathbf{Y}(\mathbf{z})$ to be a (nonsingular) constant matrix results in the parallel-chord method [25]. In [25], the authors present an in-depth analysis of the theoretical results on fixed-point iterations including conditions for guaranteed convergence, etc. The key result on which Newton-type methods rely is the mean-value theorem. A slightly modified form of that stated in [21] is presented here.

THEOREM 2.4 (Mean-value theorem) *Let* $A \in \mathbb{R}^m$ *be open and connected and let* $f : A \to \mathbb{R}$ *be differentiable on* A. *If* A *contains the line segment with endpoints* $\mathbf{a}$ *and* $\mathbf{b}$, *then there exists a point* $\mathbf{c} = \lambda\mathbf{a} + (1 - \lambda)\mathbf{b}$ *with* $\lambda \in (0, 1)$ *such that*

$$f(\mathbf{b}) - f(\mathbf{a}) = \nabla f(\mathbf{c})^{\mathrm{T}}(\mathbf{b} - \mathbf{a}). \tag{5}$$

The result that we rely upon is the parametric extension of the mean-value theorem.

COROLLARY 2.5 (Parametric mean-value theorem) *Let* $A \in \mathbb{R}^m$ *be open and connected and let* $P \subset \mathbb{R}^{n_p}$, *and let* $f : A \times P \to \mathbb{R}$ *be differentiable on* A *for every* $\mathbf{p} \in P$. *Let* $\mathbf{v}, \mathbf{w} : P \to A$. *Suppose that, for every* $\mathbf{p} \in P$, *the set* A *contains the line segment with endpoints* $\mathbf{v}(\mathbf{p})$ *and* $\mathbf{w}(\mathbf{p})$. *Then there exists* $\mathbf{y} : P \to A$ *such that, for each* $\mathbf{p} \in P$, $\mathbf{y}(\mathbf{p}) = \lambda(\mathbf{p})\mathbf{v}(\mathbf{p}) + (1 - \lambda(\mathbf{p}))\mathbf{w}(\mathbf{p})$ *for some* $\lambda : P \to (0, 1)$, *and*

$$f(\mathbf{w}(\mathbf{p}), \mathbf{p}) - f(\mathbf{v}(\mathbf{p}), \mathbf{p}) = \nabla_\mathbf{x} f(\mathbf{y}(\mathbf{p}), \mathbf{p})^{\mathrm{T}}(\mathbf{w}(\mathbf{p}) - \mathbf{v}(\mathbf{p})). \tag{6}$$

Proof Proof can be found in [34]. ■

2.2 *Interval analysis*

This section contains a very brief overview of interval analysis, specifically for the application of calculating convex and concave relaxations of functions below. For more information on interval arithmetic, the reader is directed to [24].

DEFINITION 2.6 (Intervals) *An interval will be defined as the connected compact set*

$$Z = [\mathbf{z}^L, \mathbf{z}^U] = \{\mathbf{z} \in \mathbb{R}^m : \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U\},$$

with $\mathbf{z}^L, \mathbf{z}^U \in \mathbb{R}^m$ ($\mathbf{z}^L \leq \mathbf{z}^U$) *as the lower and upper bounds, respectively. The set of all nonempty interval subsets of* $\mathbb{R}^m$ *is denoted* $\mathbb{IR}^m$. *The set of all nonempty interval subsets of any set* $A \subset \mathbb{R}^m$ *is denoted* $\mathbb{I}A$.

DEFINITION 2.7 (Interval vector) *An* interval vector $Z \in \mathbb{IR}^m$ *is an m-dimensional vector whose components are intervals denoted by a subscript* Z_i *for* $i = 1, 2, \ldots, m$.

Interval vectors will be referred to as intervals where the context is clear. An interval-valued function $F : \mathbb{I}A \to \mathbb{IR}^n$, evaluated at any $Z \in \mathbb{I}A$, is denoted in capitals as $F(Z)$.

DEFINITION 2.8 (Width) *The* width *of an interval* $Z \in \mathbb{IR}$ *is defined as* $w(Z) = z^U - z^L$.

DEFINITION 2.9 (Interval extension) *Let* $Z \subset \mathbb{R}^m$. *An interval-valued function* $F : \mathbb{I}Z \to \mathbb{IR}^n$ *is called an* interval extension *of the real-valued function* $\mathbf{f} : Z \to \mathbb{R}^n$, *if*

$$[\mathbf{f}(\mathbf{z}), \mathbf{f}(\mathbf{z})] = F([\mathbf{z}, \mathbf{z}]), \quad \forall\, \mathbf{z} \in Z.$$

DEFINITION 2.10 (Inclusion monotonicity [20,24]) *Let* $Z \subset \mathbb{R}^m$. *An interval-valued function* $F : \mathbb{I}Z \to \mathbb{IR}^n$ *is called* inclusion monotonic *if for every* $A, B \in \mathbb{I}Z$,

$$B \subset A \Rightarrow F(B) \subset F(A). \tag{7}$$

DEFINITION 2.11 (Inclusion function) *Let* $Z \subset \mathbb{R}^m$. *An interval-valued function* $F : \mathbb{I}Z \to \mathbb{IR}^n$ *is called an* inclusion function *of* $\mathbf{f}$ *on* Z *if*

$$\mathbf{f}(A) \subset F(A), \quad \forall\, A \in \mathbb{I}Z,$$

where $\mathbf{f}(A)$ *is the image of* A *under* $\mathbf{f}$.

THEOREM 2.12 (Fundamental theorem of interval analysis) *Let* $Z \in \mathbb{IR}^m$ *and let* $F : \mathbb{I}Z \to \mathbb{IR}^n$ *be an inclusion monotonic interval extension of* $\mathbf{f} : Z \to \mathbb{R}^n$. *Then* F *is an inclusion function of* $\mathbf{f}$ *on* Z.

2.3 *McCormick relaxations*

McCormick [18] developed a technique for generating convex and concave relaxations of a given function, defined as follows.

DEFINITION 2.13 (Relaxations of functions [19]) *Given a convex set* $Z \subset \mathbb{R}^n$ *and a function* $f : Z \to \mathbb{R}$, *a convex function* $f^c : Z \to \mathbb{R}$ *is a* convex relaxation (*or convex underestimator*) *of* f *on* Z *if* $f^c(\mathbf{z}) \leq f(\mathbf{z})$ *for every* $\mathbf{z} \in Z$. *A concave function* $f^C : Z \to \mathbb{R}$ *is a* concave relaxation (*or concave overestimator*) *of* f *on* Z *if* $f^C(\mathbf{z}) \geq f(\mathbf{z})$ *for every* $\mathbf{z} \in Z$.

The relaxations of vector-valued or matrix-valued functions are defined by applying the above inequalities componentwise.

DEFINITION 2.14 (Univariate intrinsic function [31]) *The function $u : B \subset \mathbb{R} \to \mathbb{R}$ is a* univariate intrinsic function *if, for any $A \in \mathbb{I}B$, the following are known and can be evaluated computationally*:

(1) *an interval extension of u on A that is an inclusion function of u on A,*
(2) *a concave relaxation of u on A,*
(3) *a convex relaxation of u on A.*

In order to construct relaxations of a function using the rules outlined by McCormick [18], the function must be *factorable*, defined as follows.

DEFINITION 2.15 (Factorable function [31]) *A function $f : Z \subset \mathbb{R}^{n_z} \to \mathbb{R}$ is* factorable *if it can be expressed in terms of a finite number of factors $v_1, \ldots, v_m$ such that, given $\mathbf{z} \in Z$, $v_i(\mathbf{z}) = z_i$ for $i = 1, \ldots, n_z$, and for each $n_z < k \leq m$, v_k is defined as either*

(a) $v_k = v_i + v_j,\ i, j < k$, *or*
(b) $v_k = v_i v_j,\ i, j < k$, *or*
(c) $v_k = u_k \circ v_i,\ i < k$, *where $u_k : B_k \to \mathbb{R}$ is a univariate intrinsic function,*

and $f(\mathbf{z}) = v_m(\mathbf{z})$. A vector-valued function $\mathbf{f}$ is factorable if every component f_i is factorable.

The functions f, $\mathbf{g}$, and $\mathbf{h}$ considered in this paper are assumed to be factorable. Such an assumption is not very restrictive since this includes almost any function that can be represented finitely on a computer. McCormick's [18] relaxation technique computes convex and concave relaxations of factorable functions by recursively applying simple rules for relaxing binary addition, binary multiplication, and univariate composition with univariate intrinsic functions.

DEFINITION 2.16 (Composite relaxations: $\mathbf{u}_{\mathscr{G}}, \mathbf{o}_{\mathscr{G}}$) *Let $D \subset \mathbb{R}^{n_x}$, $Z \in \mathbb{I}D$, and $P \in \mathbb{IR}^{n_p}$. Let $\mathscr{G} : D \times P \to \mathbb{R}^{n_x}$. The functions $\mathbf{u}_{\mathscr{G}}, \mathbf{o}_{\mathscr{G}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times P \to \mathbb{R}^{n_x}$ are called* composite relaxations *of $\mathscr{G}$ on $Z \times P$ if for any $\boldsymbol{\psi}^c, \boldsymbol{\psi}^C : P \to \mathbb{R}^{n_x}$, the functions $\mathbf{u}_{\mathscr{G}}(\boldsymbol{\psi}^c(\cdot), \boldsymbol{\psi}^C(\cdot), \cdot)$ and $\mathbf{o}_{\mathscr{G}}(\boldsymbol{\psi}^c(\cdot), \boldsymbol{\psi}^C(\cdot), \cdot)$ are, respectively, convex and concave relaxations of $\mathscr{G}(\mathbf{q}(\cdot), \cdot)$ on P for any function $\mathbf{q} : P \to Z$ and any pair of convex and concave relaxations $(\boldsymbol{\psi}^c, \boldsymbol{\psi}^C)$ of $\mathbf{q}$ on P.*

Provided that $\mathscr{G}$ is factorable, functions $\mathbf{u}_{\mathscr{G}}$ and $\mathbf{o}_{\mathscr{G}}$ satisfying the previous definition can be computed using generalized McCormick relaxations as described in [31]. By the properties of generalized McCormick relaxations, the functions $\mathbf{u}_{\mathscr{G}}$ and $\mathbf{o}_{\mathscr{G}}$ are continuous on $\mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times P$.

Remark 1 Strictly speaking, by the definition of generalized McCormick relaxations and the definition of composite relaxations given in [31], the bounding information (i.e. $Z \times P$ in Definition 2.16) is required and should be taken as explicit arguments of $\mathbf{u}_{\mathscr{G}}$ and $\mathbf{o}_{\mathscr{G}}$. However, for notational clarity in this work, the bounding information will not be passed as arguments of the composite relaxations and instead will be stated explicitly wherever composite relaxations are used.

Remark 2 More generally, composite relaxations for any arbitrary function $\mathscr{G}(\mathbf{v}(\cdot), \mathbf{w}(\cdot), \ldots, \mathbf{z}(\cdot), \cdot)$, on P, taking arbitrarily many functions as arguments, can be constructed in an analogous manner. Also, the inner functions need not be vector valued, but can be matrix valued, by treating each column vector of the matrix-valued function as a vector-valued function and applying the above definition.

2.4 *Subgradients*

Since McCormick relaxations are potentially nondifferentiable, subgradients provide useful information to a nonsmooth optimization algorithm or can be used to compute affine relaxations of the functions. The rules for calculating subgradients of McCormick relaxations and corresponding affine relaxations are thoroughly discussed in [19].

DEFINITION 2.17 (Subgradients) *Let $Z \subset \mathbb{R}^n$ be a nonempty convex set, $f^c : Z \to \mathbb{R}$ be convex and $f^C : Z \to \mathbb{R}$ be concave. A vector-valued function $\mathbf{s}_f^c : Z \to \mathbb{R}^n$ is called a subgradient of f^c on Z if for every $\bar{\mathbf{z}} \in Z$, $f^c(\mathbf{z}) \geq f^c(\bar{\mathbf{z}}) + (\mathbf{s}_f^c(\bar{\mathbf{z}}))^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}})$, $\forall \mathbf{z} \in Z$. Likewise, a vector-valued function $\mathbf{s}_f^C : Z \to \mathbb{R}^n$ is called a subgradient of f^C on Z if for every $\bar{\mathbf{z}} \in Z$, $f^C(\mathbf{z}) \leq f^C(\bar{\mathbf{z}}) + (\mathbf{s}_f^C(\bar{\mathbf{z}}))^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}})$, $\forall \mathbf{z} \in Z$.*

Remark 3 Subgradients are not unique in general. The procedures in [19] compute a single element of the subdifferential, therefore the subgradient functions above are well defined. Subgradients of vector-valued functions $\mathbf{f}^c, \mathbf{f}^C : Z \to \mathbb{R}^m$, convex and concave, respectively, will be matrix-valued functions denoted $\boldsymbol{\sigma}_{\mathbf{f}}^c, \boldsymbol{\sigma}_{\mathbf{f}}^C : Z \to \mathbb{R}^{n \times m}$. Furthermore, subgradients of matrix-valued functions $\mathbf{F}^c, \mathbf{F}^C : Z \to \mathbb{R}^{m \times m}$, convex and concave, respectively, will be 3^{rd}-order tensor-valued functions denoted $\hat{\boldsymbol{\sigma}}_{\mathbf{F}}^c, \hat{\boldsymbol{\sigma}}_{\mathbf{F}}^C : Z \to \mathbb{R}^{n \times m \times m}$.

DEFINITION 2.18 (Affine relaxations) *Let $Z \subset \mathbb{R}^n$ be a nonempty convex set and define $\mathbf{f} : Z \to \mathbb{R}^n$. The functions $\mathbf{f}^a, \mathbf{f}^A : Z \to \mathbb{R}^n$ are called* affine relaxations *of $\mathbf{f}$ if $\mathbf{f}^a(\mathbf{z}) \leq \mathbf{f}(\mathbf{z}) \leq \mathbf{f}^A(\mathbf{z}), \forall \mathbf{z} \in Z$, and $\mathbf{f}^a$ and $\mathbf{f}^A$ are affine on Z.*

In the same notation as the above definition, a natural choice of affine relaxations is given by

$$\mathbf{f}^a(\mathbf{z}) = \mathbf{f}^c(\bar{\mathbf{z}}) + (\boldsymbol{\sigma}_f^c(\bar{\mathbf{z}}))^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}}) \quad \text{and} \quad \mathbf{f}^A(\mathbf{z}) = \mathbf{f}^C(\bar{\mathbf{z}}) + (\boldsymbol{\sigma}_f^C(\bar{\mathbf{z}}))^{\mathrm{T}}(\mathbf{z} - \bar{\mathbf{z}}).$$

DEFINITION 2.19 (Composite subgradients: $\mathcal{S}_{u_{\mathscr{G}}}, \mathcal{S}_{o_{\mathscr{G}}}$) *Let $D \subset \mathbb{R}^{n_x}$, $P \in \mathbb{IR}^{n_p}$, and $Z \in \mathbb{I}D$. Let $\mathbf{q} : P \to Z$ and $\mathscr{G} : D \times P \to \mathbb{R}^{n_x}$. Let $\mathbf{u}_{\mathscr{G}}, \mathbf{o}_{\mathscr{G}}$ be composite relaxations of $\mathscr{G}$ on $Z \times P$. The functions $\mathcal{S}_{u_{\mathscr{G}}}, \mathcal{S}_{o_{\mathscr{G}}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p \times n_x} \times \mathbb{R}^{n_p \times n_x} \times P \to \mathbb{R}^{n_p \times n_x}$ are called* composite subgradients *of $\mathbf{u}_{\mathscr{G}}$ and $\mathbf{o}_{\mathscr{G}}$ on $Z \times P$, respectively, if for any $\boldsymbol{\psi}^c, \boldsymbol{\psi}^C : P \to \mathbb{R}^{n_x}$ and $\boldsymbol{\sigma}_{\psi}^c : \boldsymbol{\sigma}_{\psi}^C : P \to \mathbb{R}^{n_p \times n_x}$, the functions $\mathcal{S}_{u_{\mathscr{G}}}(\boldsymbol{\psi}^c(\cdot), \boldsymbol{\psi}^C(\cdot), \boldsymbol{\sigma}_{\psi}^c(\cdot), \boldsymbol{\sigma}_{\psi}^C(\cdot), \cdot)$, and $\mathcal{S}_{o_{\mathscr{G}}}(\boldsymbol{\psi}^c(\cdot), \boldsymbol{\psi}^C(\cdot), \boldsymbol{\sigma}_{\psi}^c(\cdot), \boldsymbol{\sigma}_{\psi}^C(\cdot), \cdot)$ are, respectively, subgradients of $\mathbf{u}_{\mathscr{G}}(\boldsymbol{\psi}^c(\cdot), \boldsymbol{\psi}^C(\cdot), \cdot)$ and $\mathbf{o}_{\mathscr{G}}(\boldsymbol{\psi}^c(\cdot), \boldsymbol{\psi}^C(\cdot), \cdot)$, provided $\boldsymbol{\psi}^c$ and $\boldsymbol{\psi}^C$ are, respectively, convex and concave relaxations of $\mathbf{q}$ on P and $\boldsymbol{\sigma}_{\psi}^c$ and $\boldsymbol{\sigma}_{\psi}^C$ are, respectively, subgradients of $\boldsymbol{\psi}^c$ and $\boldsymbol{\psi}^C$ on P.*

Remark 4 Similar to composite relaxations, composite subgradients of convex and concave relaxations of any $\mathscr{G}(\mathbf{v}(\cdot), \mathbf{q}(\cdot), \ldots, \mathbf{z}(\cdot), \cdot)$ on P, taking arbitrarily many functions as arguments, can be constructed analogously to the case considered in Definition 2.19. Again, the inner functions need not be vector-valued, but can be matrix-valued, by treating each column vector of the matrix-valued function as a vector-valued function and applying the above definition. As per Remark 3, subgradients of a matrix-valued function will be third-order tensors.

3. Relaxation of implicit functions

This section contains new developments regarding relaxations of implicit functions. Two different methods for constructing relaxations of implicit functions will be discussed. The first technique

(Sections 3.1 and 3.2) is directly relaxing fixed-point iterations that are used to approximate solutions of systems of equations, as in [31]. This method, along with new results are discussed in detail in Section 3.1. This approach can be quite limited, however, and shortcomings of this method are discussed in Section 3.2. The second technique (Sections 3.3 and 3.4), referred to as Relaxations of Solutions of Parametric Systems, circumvents the shortcomings of the first method by relaxing the actual implicit functions themselves, without reference to an associated fixed-point iteration, and is thus more broadly applicable. The case of parametric linear systems is discussed in Section 3.3. In Section 3.4, the case of parametric nonlinear systems is discussed.

3.1 *Direct relaxation of fixed-point iterations*

Consider the system of equations in (3). Let the (factorable) function $\boldsymbol{\phi} : D_x \times D_p \to \mathbb{R}^{n_x}$ be an algebraic rearrangement of $\mathbf{h}$ such that $\mathbf{h}(\mathbf{z}, \mathbf{p}) = \mathbf{z} - \boldsymbol{\phi}(\mathbf{z}, \mathbf{p}) = \mathbf{0} \Leftrightarrow \mathbf{z} = \boldsymbol{\phi}(\mathbf{z}, \mathbf{p})$ and $D_x \times D_p \subset D_y$. For example, consider $h(z, p) = z - \sin(z + p) = 0 \Leftrightarrow z = \sin(z + p)$ or $h(z, p) = z^2 + pz + C = 0 \Leftrightarrow z = -(z^2 + C)/p$.

Assumption 3.1 There exists $\mathbf{x} : P \to \mathbb{R}^{n_x}$ such that $\mathbf{x}(\mathbf{p}) = \boldsymbol{\phi}(\mathbf{x}(\mathbf{p}), \mathbf{p}), \forall \mathbf{p} \in P$, and an interval $[\mathbf{x}^L, \mathbf{x}^U] \equiv X \in \mathbb{IR}^{n_x}$ is known such that $\mathbf{x}(P) \subset X$ and $\mathbf{x}(\mathbf{p})$ is unique in X for all $\mathbf{p} \in P$.

The parametric extension of the well-known interval-Newton method, which is discussed in [10,24,34], exhibits the theoretical capability of finding an X satisfying this assumption. Finding such an X is really a precursor to calculating relaxations since, for the purposes of this paper, it is desired to relax a single implicit function.

In [31], the authors consider the computation of relaxations of $\mathbf{x}^k : P \to \mathbb{R}^{n_x}$, the approximations of $\mathbf{x}$, defined by the fixed-point iteration:

$$\mathbf{x}^{k+1}(\mathbf{p}) := \boldsymbol{\phi}(\mathbf{x}^k(\mathbf{p}), \mathbf{p}), \quad \forall\, \mathbf{p} \in P. \tag{8}$$

If $\boldsymbol{\phi}(\cdot, \mathbf{p})$ is a contraction mapping on X for every $\mathbf{p} \in P$, then this iteration is referred to as a successive-substitution fixed-point iteration. Under this assumption, it can be shown that $\{\mathbf{x}^k\} \to \mathbf{x}$ so that this method provides relaxations of arbitrarily good approximations of $\mathbf{x}$. However, this result is rather weak in that it does not provide us with guaranteed valid relaxations of the implicit function $\mathbf{x}$ upon finite termination. In contrast, the following result provides sequences, $\{\mathbf{x}^{k,c}\}$ and $\{\mathbf{x}^{k,C}\}$, such that $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$ are relaxations of $\mathbf{x}$ on P, for every $k \in \mathbb{N}$. Moreover, this result does not require contractivity of $\boldsymbol{\phi}$ on X. Thus, although approximations of the value of $\mathbf{x}$ may not even be available, valid relaxations of $\mathbf{x}$ are readily calculable.

DEFINITION 3.2 ($\bar{\mathbf{u}}_\phi, \bar{\mathbf{o}}_\phi$) *Let $\mathbf{u}_\phi, \mathbf{o}_\phi$ be composite relaxations of $\boldsymbol{\phi}$ on $X \times P$. The functions $\bar{\mathbf{u}}_\phi, \bar{\mathbf{o}}_\phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times P \to \mathbb{R}^{n_x}$ will be defined as*

$$\bar{\mathbf{u}}_\phi(\mathbf{z}^c, \mathbf{z}^C, \mathbf{p}) \equiv \max\{\mathbf{z}^c, \mathbf{u}_\phi(\mathbf{z}^c, \mathbf{z}^C, \mathbf{p})\},$$
$$\bar{\mathbf{o}}_\phi(\mathbf{z}^c, \mathbf{z}^C, \mathbf{p}) \equiv \min\{\mathbf{z}^C, \mathbf{o}_\phi(\mathbf{z}^c, \mathbf{z}^C, \mathbf{p})\},$$

$\forall(\mathbf{z}^c, \mathbf{z}^C, \mathbf{p}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times P$ with the max / min *operations applied componentwise.*

DEFINITION 3.3 ($\mathcal{S}_{\bar{\mathbf{u}}_\phi}, \mathcal{S}_{\bar{\mathbf{o}}_\phi}$) *Let $\mathbf{u}_\phi, \mathbf{o}_\phi$ be composite relaxations of $\boldsymbol{\phi}$ on $X \times P$. Let $\mathcal{S}_{u_\phi}, \mathcal{S}_{o_\phi}$ be composite subgradients of $\mathbf{u}_\phi$ and $\mathbf{o}_\phi$ on $X \times P$, respectively. The functions $\mathcal{S}_{\bar{\mathbf{u}}_\phi}, \mathcal{S}_{\bar{\mathbf{o}}_\phi} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times$*

$\mathbb{R}^{n_p \times n_x} \times \mathbb{R}^{n_p \times n_x} \times P \to \mathbb{R}^{n_p \times n_x}$ *will be defined as*

$$\mathcal{S}_{\bar{\mathbf{u}}_\phi}(\mathbf{z}^c, \mathbf{z}^C, \boldsymbol{\sigma}_\mathbf{z}^c, \boldsymbol{\sigma}_\mathbf{z}^C, \bar{\mathbf{p}}) = \begin{cases} \boldsymbol{\sigma}_\mathbf{z}^c & \text{if } \bar{\mathbf{u}}_\phi(\mathbf{z}^c, \mathbf{z}^C, \bar{\mathbf{p}}) = \mathbf{z}^c \\ \mathcal{S}_{\mathbf{u}_\phi}(\mathbf{z}^c, \mathbf{z}^C, \boldsymbol{\sigma}_\mathbf{z}^c, \boldsymbol{\sigma}_\mathbf{z}^C, \bar{\mathbf{p}}) & \text{otherwise} \end{cases}$$

$$\mathcal{S}_{\bar{\mathbf{o}}_\phi}(\mathbf{z}^c, \mathbf{z}^C, \boldsymbol{\sigma}_\mathbf{z}^c, \boldsymbol{\sigma}_\mathbf{z}^C, \bar{\mathbf{p}}) = \begin{cases} \boldsymbol{\sigma}_\mathbf{z}^C & \text{if } \bar{\mathbf{o}}_\phi(\mathbf{z}^c, \mathbf{z}^C, \bar{\mathbf{p}}) = \mathbf{z}^C \\ \mathcal{S}_{\mathbf{o}_\phi}(\mathbf{z}^c, \mathbf{z}^C, \boldsymbol{\sigma}_\mathbf{z}^c, \boldsymbol{\sigma}_\mathbf{z}^C, \bar{\mathbf{p}}) & \text{otherwise} \end{cases}$$

$\forall (\mathbf{z}^c, \mathbf{z}^C, \boldsymbol{\sigma}_\mathbf{z}^c, \boldsymbol{\sigma}_\mathbf{z}^C, \bar{\mathbf{p}}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p \times n_x} \times \mathbb{R}^{n_p \times n_x} \times P$.

It should be noted that the functions $\mathcal{S}_{\bar{u}_\phi}$ and $\mathcal{S}_{\bar{o}_\phi}$ define composite subgradients of $\bar{\mathbf{u}}_\phi$ and $\bar{\mathbf{o}}_\phi$ on $X \times P$, respectively.

THEOREM 3.4 *Let* $\mathbf{x}^{0,c}, \mathbf{x}^{0,C} : P \to \mathbb{R}^{n_x}$ *be defined by* $\mathbf{x}^{0,c}(\mathbf{p}) = \mathbf{x}^L$ *and* $\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{x}^U$ *for all* $\mathbf{p} \in P$. *Then the elements of the sequences* $\{\mathbf{x}^{k,c}\}$ *and* $\{\mathbf{x}^{k,C}\}$ *defined by* $\mathbf{x}^{k+1,c}(\cdot) = \bar{\mathbf{u}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ *and* $\mathbf{x}^{k+1,C}(\cdot) = \bar{\mathbf{o}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ *are convex and concave relaxations of* $\mathbf{x}$ *on* P, *respectively, for every* $k \in \mathbb{N}$.

Proof $\mathbf{x}^{0,c}$ and $\mathbf{x}^{0,C}$ are trivially convex and concave relaxations of $\mathbf{x}$ on P, respectively. Suppose this is true of $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$ for some $k \geq 0$. By Definition 2.16, $\mathbf{u}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathbf{o}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ are also relaxations of $\mathbf{x}(\cdot) = \boldsymbol{\phi}(\mathbf{x}(\cdot), \cdot)$ on P. Since the maximum of two convex functions is convex and the minimum of two concave functions is concave, $\mathbf{x}^{k+1,c}(\cdot) = \bar{\mathbf{u}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathbf{x}^{k+1,C}(\cdot) = \bar{\mathbf{o}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ are convex and concave relaxations of $\mathbf{x}(\cdot) = \boldsymbol{\phi}(\mathbf{x}(\cdot), \cdot)$ on P, respectively. Induction completes the proof. ■

THEOREM 3.5 *Let* $\mathbf{x}^{0,c}, \mathbf{x}^{0,C} : P \to \mathbb{R}^{n_x}$ *be defined by* $\mathbf{x}^{0,c}(\mathbf{p}) = \mathbf{x}^L$ *and* $\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{x}^U$, *for all* $\mathbf{p} \in P$. *Let* $\boldsymbol{\sigma}_\mathbf{x}^{0,c}(\mathbf{p}) = \boldsymbol{\sigma}_\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{0}$, *for all* $\mathbf{p} \in P$. *Let relaxations of* $\mathbf{x}$ *on* P *be given by* $\mathbf{x}^{k+1,c}(\cdot) = \bar{\mathbf{u}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ *and* $\mathbf{x}^{k+1,C}(\cdot) = \bar{\mathbf{o}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$, $k \in \mathbb{N}$. *Then the sequences* $\{\boldsymbol{\sigma}_\mathbf{x}^{k,c}\}$ *and* $\{\boldsymbol{\sigma}_\mathbf{x}^{k,C}\}$ *defined by*

$$\boldsymbol{\sigma}_\mathbf{x}^{k+1,c}(\cdot) := \mathcal{S}_{\bar{\mathbf{u}}_\phi}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\boldsymbol{\sigma}_\mathbf{x}^{k+1,C}(\cdot) := \mathcal{S}_{\bar{\mathbf{o}}_\phi}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)$$

are, respectively, subgradients of $\mathbf{x}^{k+1,c}$ *and* $\mathbf{x}^{k+1,C}$ *on* P *for* $k \in \mathbb{N}$.

Proof From the hypothesis, $\mathbf{x}^{0,c}$ and $\mathbf{x}^{0,C}$ are (constant) convex and concave relaxations of $\mathbf{x}$ on P, respectively, and $\boldsymbol{\sigma}_\mathbf{x}^{0,c} = \boldsymbol{\sigma}_\mathbf{x}^{0,C} = \mathbf{0}$ are subgradients of $\mathbf{x}^{0,c}$ and $\mathbf{x}^{0,C}$ on P, respectively. Suppose this holds for $k \in \mathbb{N}$. Then we have $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$, convex and concave relaxations of $\mathbf{x}$ on P, respectively, and $\boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot)$ and $\boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot)$, subgradients of $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$ on P, respectively. By the definition of the composite subgradient (Definition 2.19), subgradients of $\mathbf{u}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathbf{o}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ on P are given by $\mathcal{S}_{\mathbf{u}_\phi}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathcal{S}_{\mathbf{o}_\phi}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)$, respectively, and by Definition 3.3, $\mathcal{S}_{\bar{\mathbf{u}}_\phi}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathcal{S}_{\bar{\mathbf{o}}_\phi}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)$ are subgradients of

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot) = \max\{\mathbf{x}^{k,c}(\cdot), \mathbf{u}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)\},$$
$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot) = \min\{\mathbf{x}^{k,C}(\cdot), \mathbf{o}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)\}$$

on P, respectively. Induction completes the proof. ■

It is of great interest to understand when this procedure for calculating relaxations works well. Although improvement on the bounds cannot be guaranteed in general, one can find cases when improvement is definitely not possible; thus providing a necessary condition for improvement.

THEOREM 3.6 *Let* $\{\mathbf{x}^k\}$ *be a sequence generated by the fixed-point iteration* (8) *starting from* $\mathbf{x}^0(\mathbf{p}) \in X$, $\forall \mathbf{p} \in P$. *If* $\mathbf{x}^k(\mathbf{p}) \notin X$ *for some* $\mathbf{p} \in P$, $k \in \mathbb{N}$, *then the sequences* $\{\mathbf{x}^{k,c}\}$ *and* $\{\mathbf{x}^{k,C}\}$ *from Theorem* 3.4 *are such that there exists a* $\mathbf{p} \in P$ *such that* $\mathbf{x}^{k,c}(\mathbf{p}) = \mathbf{x}^L$ *or* $\mathbf{x}^{k,C}(\mathbf{p}) = \mathbf{x}^U$ *for every* $k \in \mathbb{N}$.

Proof By hypothesis, $\mathbf{x}^0(\mathbf{p}) \in X$ for every $\mathbf{p} \in P$ and $\mathbf{x}^{0,c} = \mathbf{x}^L$ and $\mathbf{x}^{0,C} = \mathbf{x}^U$. Therefore $\mathbf{x}^{0,c}$ and $\mathbf{x}^{0,C}$ are convex and concave relaxations of $\mathbf{x}^0$ on P, respectively. Suppose this is true for $(K-1) \in \mathbb{N}$ where K is the iteration in which $\mathbf{x}^K(\bar{\mathbf{p}}) \notin X$ such that $\mathbf{x}^k(\bar{\mathbf{p}}) \in X, \forall k < K$ for some $\bar{\mathbf{p}} \in P$. Then by Definition 2.16 and Theorem 3.4, $\mathbf{u}_\phi(\mathbf{x}^{K-1,c}(\mathbf{p}), \mathbf{x}^{K-1,C}(\mathbf{p}), \mathbf{p}) \leq \boldsymbol{\phi}(\mathbf{x}^{K-1}(\mathbf{p}), \mathbf{p}) \leq \mathbf{o}_\phi(\mathbf{x}^{K-1,c}(\mathbf{p}), \mathbf{x}^{K-1,C}(\mathbf{p}), \mathbf{p})$ for every $\mathbf{p} \in P$ (noting $\mathbf{x}^{K-1}(\mathbf{p}) \in X, \forall \mathbf{p} \in P$). Since $\mathbf{x}^K(\mathbf{p}) = \boldsymbol{\phi}(\mathbf{x}^{K-1}(\mathbf{p}), \mathbf{p}), \forall \mathbf{p} \in P$, this implies $\mathbf{u}_\phi(\mathbf{x}^{K-1,c}(\mathbf{p}), \mathbf{x}^{K-1,C}(\mathbf{p}), \mathbf{p}) \leq \mathbf{x}^K(\mathbf{p}) \leq \mathbf{o}_\phi(\mathbf{x}^{K-1,c}(\mathbf{p}), \mathbf{x}^{K-1,C}(\mathbf{p}), \mathbf{p}), \forall \mathbf{p} \in P$. However, since $\mathbf{x}^K(\bar{\mathbf{p}}) \notin X$, it follows that $\mathbf{u}_\phi(\mathbf{x}^{K-1,c}(\bar{\mathbf{p}}), \mathbf{x}^{K-1,C}(\bar{\mathbf{p}}), \bar{\mathbf{p}}) < \mathbf{x}^L$ or $\mathbf{o}_\phi(\mathbf{x}^{K-1,c}(\bar{\mathbf{p}}), \mathbf{x}^{K-1,C}(\bar{\mathbf{p}}), \bar{\mathbf{p}}) > \mathbf{x}^U$. Therefore $\bar{\mathbf{u}}_\phi(\mathbf{x}^{K-1,c}(\bar{\mathbf{p}}), \mathbf{x}^{K-1,C}(\bar{\mathbf{p}}), \bar{\mathbf{p}}) = \mathbf{x}^L$ or $\bar{\mathbf{o}}_\phi(\mathbf{x}^{K-1,c}(\bar{\mathbf{p}}), \mathbf{x}^{K-1,C}(\bar{\mathbf{p}}), \bar{\mathbf{p}}) = \mathbf{x}^U$. Additionally, since $\mathbf{u}_\phi$ and $\mathbf{o}_\phi$ are composite relaxations of $\boldsymbol{\phi}$ on $X \times P$, and will therefore always bound $\boldsymbol{\phi}$, since ϕ maps $X \times P$ outside of X, this implies the result holds for every iteration $k < K$ as well. ■

3.2 *Direct relaxation of Newton-type iterations*

According to Theorem 3.6, the property that $\boldsymbol{\phi}$ maps $X \times P$ into X is desirable in order to calculate relaxations that are potential improvements on the original bounds of X. This property will be exhibited by any $\boldsymbol{\phi}$ that is a contraction mapping. Consider the system of Equations (3) and now suppose that $\mathbf{h}$ cannot be rearranged algebraically as in the previous section, such that (8) is contractive. Thus, $\mathbf{h}$ will be a member of a more general class of functions. The following result guarantees that a different form of fixed-point iteration can still be constructed from any such system and under some other fixed-point results, may be guaranteed to be contractive. However, as will be shown in this section, the fact that $\boldsymbol{\phi}$ is contractive is not enough to calculate relaxations of $\mathbf{x}$ that are guaranteed to be refinements on the bounds of X using the method of Section 3.1. Although, this property is a necessary condition.

PROPOSITION 3.7 *For any function* $\mathbf{h} : A \subset \mathbb{R}^n \to \mathbb{R}^n$, *there exists* $\boldsymbol{\phi} : A \to \mathbb{R}^n$ *such that* $\boldsymbol{\phi}(\mathbf{z}) = \mathbf{z}$ *if and only if* $\mathbf{h}(\mathbf{z}) = \mathbf{0}$.

Proof Proof can be found in [34]. ■

By the previous proposition, the function $\boldsymbol{\phi} : X \times P \to \mathbb{R}^{n_x}$ can be defined as

$$\boldsymbol{\phi}(\mathbf{z},\mathbf{p}) \equiv \mathbf{z} - \mathbf{Y}(\mathbf{z}, \mathbf{p})\mathbf{h}(\mathbf{z}, \mathbf{p}) \tag{9}$$

with $\mathbf{Y}(\mathbf{z}, \mathbf{p}) \in \mathbb{R}^{n_x \times n_x}$ being nonsingular for all $(\mathbf{z}, \mathbf{p}) \in X \times P$. Then

$$\mathbf{x}^{k+1}(\mathbf{p}) := \boldsymbol{\phi}(\mathbf{x}^k(\mathbf{p}), \mathbf{p}) \tag{10}$$

is a fixed-point iteration. Thus, the method of Section 3.1 can still, in principle, be used to construct relaxations of $\mathbf{x}$ on P. However, the following result shows that the relaxations of $\mathbf{x}$ constructed in this way cannot be tighter than the bounds $\mathbf{x}^L$ and $\mathbf{x}^U$.

THEOREM 3.8 *Let $\boldsymbol{\phi}$ be defined as in (9) and suppose Assumption 3.1 holds. Let $\mathbf{x}^{0,c}, \mathbf{x}^{0,C} : P \to \mathbb{R}^{n_x}$ be defined by $\mathbf{x}^{0,c}(\mathbf{p}) = \mathbf{x}^L$ and $\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{x}^U$ for all $\mathbf{p} \in P$. Let $\mathbf{u}_\phi$ and $\mathbf{o}_\phi$ be composite relaxations of $\boldsymbol{\phi}$ on $X \times P$ and let $\bar{\mathbf{u}}_\phi$ and $\bar{\mathbf{o}}_\phi$ be defined as in Definition 3.2. Let $\mathbf{x}^{k,c}, \mathbf{x}^{k,C} : P \to X$ be defined by $\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\phi(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$. Then $\mathbf{x}^{k,c}(\mathbf{p}) = \mathbf{x}^L$ and $\mathbf{x}^{k,C}(\mathbf{p}) = \mathbf{x}^U$ for every $\mathbf{p} \in P$ and all $k \in \mathbb{N}$.*

Proof Let $\mathbf{f}(\mathbf{z}, \mathbf{p}) = -\mathbf{Y(z,p)h}(\mathbf{z}, \mathbf{p})$. By the rules of McCormick [31] relaxations, $\mathbf{u}_\phi$ and $\mathbf{o}_\phi$ can be written as

$$\mathbf{u}_\phi(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}) = \mathbf{x}^{k,c}(\mathbf{p}) + \mathbf{u}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}),$$
$$\mathbf{o}_\phi(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}) = \mathbf{x}^{k,C}(\mathbf{p}) + \mathbf{o}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}),$$

where, respectively, $\mathbf{u}_\mathbf{f}$ and $\mathbf{o}_\mathbf{f}$ are composite relaxations of $\mathbf{f}$ on $X \times P$. By Definition 2.16, $\mathbf{u}_\mathbf{f}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ and $\mathbf{o}_\mathbf{f}(\mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$ are convex and concave relaxations of $\mathbf{f}(\mathbf{x}(\cdot), \cdot)$ on P, respectively, for every $k \in \mathbb{N}$. By definition, $\mathbf{f}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \mathbf{0}, \forall \mathbf{p} \in P$. Thus

$$\mathbf{u}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}) \le \mathbf{0} \le \mathbf{o}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p})$$

hold for every $\mathbf{p} \in P$ for every $k \ge 0$. Note that $\mathbf{x}^{0,c}(\mathbf{p}) = \mathbf{x}^L$ and $\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{x}^U$. Suppose the same is true of $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$, respectively. Then,

$$\mathbf{x}^{k,c}(\mathbf{p}) + \mathbf{u}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}) \le \mathbf{x}^{k,c}(\mathbf{p}) = \mathbf{x}^L,$$
$$\mathbf{x}^{k,C}(\mathbf{p}) + \mathbf{o}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p}) \ge \mathbf{x}^{k,C}(\mathbf{p}) = \mathbf{x}^U.$$

By Definition 3.2, we have

$$\mathbf{x}^{k+1,c}(\mathbf{p}) := \max\{\mathbf{x}^{k,c}, \mathbf{x}^{k,c}(\mathbf{p}) + \mathbf{u}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p})\} = \mathbf{x}^L,$$
$$\mathbf{x}^{k+1,C}(\mathbf{p}) := \min\{\mathbf{x}^{k,C}, \mathbf{x}^{k,C}(\mathbf{p}) + \mathbf{o}_\mathbf{f}(\mathbf{x}^{k,c}(\mathbf{p}), \mathbf{x}^{k,C}(\mathbf{p}), \mathbf{p})\} = \mathbf{x}^U.$$

Induction completes the proof. ■

The importance of the above theorem is that the convex and concave relaxations of the generic Newton-type form (9), discussed in Proposition 3.7, can be no tighter than the original bounds given by X, and will in fact be fixed at these bounds. For those readers that are familiar with the interval-Newton method [24], this result is analogous to the reason why one cannot simply take an interval extension of the Newton iteration and improve the initial bounds on a locally unique solution. This result motivates the need for a different technique for calculating valid convex and concave relaxations of $\mathbf{x}$. Again, it should be noted that fixed-point iterations of different forms, such as the successive-substitution iteration discussed above and in [31], may not have the same problem, per Theorem 3.4, so long as $\boldsymbol{\phi}$ maps $X \times P$ into X.

The next two sections describe a different method which is capable of constructing relaxations of $\mathbf{x}$ on P that are potentially refinements of the bounds given by X, when no successive substitution rearrangement for $\mathbf{h}$ exists that is a contraction mapping. First, the method is developed for parametric linear systems in Section 3.3. The extension to parametric nonlinear systems is developed in Section 3.4.

3.3 *Relaxations of solutions of parametric linear systems*

Consider the parametric linear system:

$$\mathbf{A}(\mathbf{p})\mathbf{z} = \mathbf{b}(\mathbf{p}), \tag{11}$$

with $\mathbf{A} : P \to D_a \subset \mathbb{R}^{n_x \times n_x}$ and $\mathbf{b} : P \to D_b \subset \mathbb{R}^{n_x}$ factorable, $\mathbf{z} \in \mathbb{R}^{n_x}$, and $\mathbf{p} \in P$.

Assumption 3.9

(1) There exists $\boldsymbol{\delta} : P \to \mathbb{R}^{n_x}$ such that $\mathbf{A}(\mathbf{p})\boldsymbol{\delta}(\mathbf{p}) = \mathbf{b}(\mathbf{p}), \forall \mathbf{p} \in P$, and an interval $[\boldsymbol{\delta}^L, \boldsymbol{\delta}^U] \equiv \Delta \in \mathbb{IR}^{n_x}$ is available such that $\boldsymbol{\delta}(P) \subset \Delta$ and $\boldsymbol{\delta}(\mathbf{p})$ is unique in Δ for every $\mathbf{p} \in P$.
(2) Intervals $A \in \mathbb{I}D_a$ and $B \in \mathbb{I}D_b$ are known such that $\mathbf{A}(P) \subset A$, $\mathbf{b}(P) \subset B$, and $0 \notin A_{ii}$ for all i.

Since $\mathbf{A}$ and $\mathbf{b}$ are factorable, the intervals A and B are easily calculable using interval analysis, e.g. by calculating their natural interval extensions. The set Δ may be computed using a parametric interval linear solver such as that in [27]. The assumption that $0 \notin A_{ii}$, $\forall i$ implies that $a_{ii}(\mathbf{p}) \neq 0$ for all $\mathbf{p} \in P$. However, this can be relaxed by assuming that there exists a preconditioning matrix $\mathbf{Y} \in \mathbb{R}^{n_x \times n_x}$ such that the diagonal elements of $\mathbf{Y}A$ do not enclose 0 and thus the product $\mathbf{YA}(\mathbf{p})$ has nonzero diagonal elements for every $\mathbf{p} \in P$. In [24], various results on the relationship between $\mathbf{Y}$, A, and $\mathbf{A}$ are discussed. The key result of this section offers a way of calculating relaxations of solutions to parametric linear systems. To begin, the solution $\boldsymbol{\delta}$ will be characterized in semi-explicit form.

DEFINITION 3.10 (**f**) *Define the function* $\mathbf{f} : D_b \times D_a \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ *such that* $\mathbf{f}(\tilde{\mathbf{b}}, \tilde{\mathbf{A}}, \tilde{\boldsymbol{\delta}}) = \tilde{\boldsymbol{\delta}}^*$, *where the ith component of* $\tilde{\boldsymbol{\delta}}^*$ *is given by the loop*:

$$\texttt{for} \quad i = 1, \ldots, n_x \ \texttt{do}$$
$$\tilde{\delta}_i^* := \frac{(\tilde{b}_i - \sum_{j<i} \tilde{a}_{ij}\tilde{\delta}_j^* - \sum_{j>i} \tilde{a}_{ij}\tilde{\delta}_j)}{\tilde{a}_{ii}} \tag{12}$$
$$\texttt{end}$$

where $\tilde{a}_{ij}$ *is the* (i,j)*th element of* $\tilde{\mathbf{A}}$, $\tilde{b}_i$ *is the ith component of* $\tilde{\mathbf{b}}$, *and* $\tilde{\delta}_i$ *is the ith component of* $\tilde{\boldsymbol{\delta}}$.

LEMMA 3.11 *Suppose Assumption 3.9 holds. Then* $\boldsymbol{\delta}(\mathbf{p}) = \mathbf{f}(\mathbf{b}(\mathbf{p}), \mathbf{A}(\mathbf{p}), \boldsymbol{\delta}(\mathbf{p}))$ *for every* $\mathbf{p} \in P$, *i.e.* $\boldsymbol{\delta}(\mathbf{p})$ *is a fixed-point of* $\mathbf{f}(\mathbf{b}(\mathbf{p}), \mathbf{A}(\mathbf{p}), \cdot)$ *for every* $\mathbf{p} \in P$.

Proof By hypothesis, $\mathbf{A}(\mathbf{p})\boldsymbol{\delta}(\mathbf{p}) = \mathbf{b}(\mathbf{p})$ holds and the ith equation can be expressed as

$$\sum_{j=1}^{n_x} a_{ij}(\mathbf{p})\delta_j(\mathbf{p}) = b_i(\mathbf{p}), \quad \forall \mathbf{p} \in P.$$

Or, equivalently written

$$a_{ii}(\mathbf{p})\delta_i(\mathbf{p}) + \sum_{j<i} a_{ij}(\mathbf{p})\delta_j(\mathbf{p}) + \sum_{j>i} a_{ij}(\mathbf{p})\delta_j(\mathbf{p}) = b_i(\mathbf{p}), \quad \forall \mathbf{p} \in P.$$

Solving for δ_i:

$$\delta_i(\mathbf{p}) = \frac{(b_i(\mathbf{p}) - \sum_{j<i} a_{ij}(\mathbf{p})\delta_j(\mathbf{p}) - \sum_{j>i} a_{ij}(\mathbf{p})\delta_j(\mathbf{p}))}{a_{ii}(\mathbf{p})}, \quad \forall \mathbf{p} \in P.$$

It immediately follows that

$$f_1(\mathbf{b}(\mathbf{p}), \mathbf{A}(\mathbf{p}), \boldsymbol{\delta}(\mathbf{p})) = \delta_1^*(\mathbf{p}) = \frac{(b_1(\mathbf{p}) - \sum_{j>1} a_{1j}(\mathbf{p})\delta_j(\mathbf{p}))}{a_{11}(\mathbf{p})} = \delta_1(\mathbf{p}).$$

Suppose $\delta_k = \delta_k^*$ holds for $k < n_x$. Then

$$f_{k+1}(\mathbf{b}(\mathbf{p}),\mathbf{A}(\mathbf{p}),\boldsymbol{\delta}(\mathbf{p})) = \delta_{k+1}^*(\mathbf{p}) = \frac{(b_{k+1}(\mathbf{p}) - \sum_{j<k+1} a_{ij}(\mathbf{p})\delta_j^* - \sum_{j>k+1} a_{ij}(\mathbf{p})\delta_j)}{a_{ii}(\mathbf{p})}$$
$$= \frac{(b_{k+1}(\mathbf{p}) - \sum_{j<k+1} a_{ij}(\mathbf{p})\delta_j - \sum_{j>k+1} a_{ij}(\mathbf{p})\delta_j)}{a_{ii}(\mathbf{p})} = \delta_{k+1}(\mathbf{p}).$$

Induction completes the proof. ■

Using the characterization of the implicit function $\boldsymbol{\delta}$ provided by Lemma 3.11, convex and concave relaxations of $\boldsymbol{\delta}$ on P can be computed by iteratively refining the bounds $\boldsymbol{\delta}^L$ and $\boldsymbol{\delta}^U$.

THEOREM 3.12 (Relaxations of parametric linear systems) *Let $\mathbf{A}^c, \mathbf{A}^C : P \to \mathbb{R}^{n_x \times n_x}$ be convex and concave relaxations of $\mathbf{A}$ on P, respectively, and let $\mathbf{b}^c, \mathbf{b}^C : P \to \mathbb{R}^{n_x}$ be convex and concave relaxations of $\mathbf{b}$ on P, respectively. Let $\mathbf{u_f}$ and $\mathbf{o_f}$ be composite relaxations of $\mathbf{f}$ on $B \times A \times \Delta \times P$. Let $\boldsymbol{\delta}^{0,c}, \boldsymbol{\delta}^{0,C} : P \to \mathbb{R}^{n_x}$ be defined by $\boldsymbol{\delta}^{0,c}(\mathbf{p}) = \boldsymbol{\delta}^L$ and $\boldsymbol{\delta}^{0,C}(\mathbf{p}) = \boldsymbol{\delta}^U$ for all $\mathbf{p} \in P$. Then the sequences $\{\boldsymbol{\delta}^{k,c}\}$ and $\{\boldsymbol{\delta}^{k,C}\}$ defined by the iteration*

$$\boldsymbol{\delta}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\mathbf{f}(\mathbf{b}^c(\cdot),\mathbf{b}^C(\cdot),\mathbf{A}^c(\cdot),\mathbf{A}^C(\cdot),\boldsymbol{\delta}^{k,c}(\cdot),\boldsymbol{\delta}^{k,C}(\cdot)),$$
$$\boldsymbol{\delta}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\mathbf{f}(\mathbf{b}^c(\cdot),\mathbf{b}^C(\cdot),\mathbf{A}^c(\cdot),\mathbf{A}^C(\cdot),\boldsymbol{\delta}^{k,c}(\cdot),\boldsymbol{\delta}^{k,C}(\cdot)),$$

are convex and concave relaxations of $\boldsymbol{\delta}$ on P, respectively, for every $k \in \mathbb{N}$, with $\bar{\mathbf{u}}_\mathbf{f}, \bar{\mathbf{o}}_\mathbf{f}$ defined analogously to Definition 3.2.

Proof $\boldsymbol{\delta}^{0,c}$ and $\boldsymbol{\delta}^{0,C}$ are trivially convex and concave relaxations of $\boldsymbol{\delta}$ on P. Suppose this holds for $k \geq 0$. Then $\boldsymbol{\delta}^{k,c}$ and $\boldsymbol{\delta}^{k,C}$ are relaxations of $\boldsymbol{\delta}$ on P. By Definition 2.16

$$\mathbf{u}_\mathbf{f}(\mathbf{b}^c(\cdot),\mathbf{b}^C(\cdot),\mathbf{A}^c(\cdot),\mathbf{A}^C(\cdot),\boldsymbol{\delta}^{k,c}(\cdot),\boldsymbol{\delta}^{k,C}(\cdot)),$$
$$\mathbf{o}_\mathbf{f}(\mathbf{b}^c(\cdot),\mathbf{b}^C(\cdot),\mathbf{A}^c(\cdot),\mathbf{A}^C(\cdot),\boldsymbol{\delta}^{k,c}(\cdot),\boldsymbol{\delta}^{k,C}(\cdot)),$$

are convex and concave relaxations of $\mathbf{f}(\mathbf{b}(\cdot),\mathbf{A}(\cdot),\boldsymbol{\delta}(\cdot))$ on P, respectively. By Lemma 3.11, $\boldsymbol{\delta}(\cdot) = \mathbf{f}(\mathbf{b}(\cdot),\mathbf{A}(\cdot),\boldsymbol{\delta}(\cdot))$, and hence these are also relaxations of $\boldsymbol{\delta}$ on P. Since the maximum of two convex functions is convex and the minimum of two concave functions is concave,

$$\boldsymbol{\delta}^{k+1,c}(\cdot) = \bar{\mathbf{u}}_\mathbf{f}(\mathbf{b}^c(\cdot),\mathbf{b}^C(\cdot),\mathbf{A}^c(\cdot),\mathbf{A}^C(\cdot),\boldsymbol{\delta}^{k,c}(\cdot),\boldsymbol{\delta}^{k,C}(\cdot)),$$
$$\boldsymbol{\delta}^{k+1,C}(\cdot) = \bar{\mathbf{o}}_\mathbf{f}(\mathbf{b}^c(\cdot),\mathbf{b}^C(\cdot),\mathbf{A}^c(\cdot),\mathbf{A}^C(\cdot),\boldsymbol{\delta}^{k,c}(\cdot),\boldsymbol{\delta}^{k,C}(\cdot)),$$

are convex and concave relaxations of $\boldsymbol{\delta}$ on P, respectively. Induction completes the proof. ■

Remark 5 The definition of $\mathbf{f}$ does not have explicit dependence on $\mathbf{p}$, however, this is just a special case of the general form (8). Therefore $\mathbf{u_f}$ and $\mathbf{o_f}$ are said to be composite relaxations of $\mathbf{f}$ on $B \times A \times \Delta \times P$, which is consistent with the definition of composite relaxations (Definition 2.16).

It should be noted that the functions $\boldsymbol{\delta}^{k,c}$ and $\boldsymbol{\delta}^{k,C}$ can be no worse than the original bounds. Thus, Theorem 3.12 offers an efficient procedure for constructing relaxations of solutions to parametric linear systems that may be, potentially significant, refinements of the original bounds. It should also be mentioned that because of how $\mathbf{f}$ is defined, each component i makes use of information from the previous $j < i$ updated components. It is said that $\mathbf{f}$ is evaluated in a sequential componentwise manner. Similarly, relaxations of $\mathbf{f}$ are calculated in a sequential componentwise manner. What this

amounts to is the sequential componentwise refinement of relaxations of δ_j making use of the newly calculated refinements of the previous components ($i < j$). This procedure is analogous to how the Gauss–Seidel method propagates the newly calculated ($i < j$) information forward to ($j > i$) components to get better approximations of the solution and potentially speed up convergence. Subgradients of these relaxations can also be calculated.

THEOREM 3.13 *Let $\mathbf{A}^c, \mathbf{A}^C : P \to \mathbb{R}^{n_x \times n_x}$ be convex and concave relaxations of $\mathbf{A}$ on P, respectively, and let $\mathbf{b}^c, \mathbf{b}^C : P \to \mathbb{R}^{n_x}$ be convex and concave relaxations of $\mathbf{b}$ on P, respectively. Let $\boldsymbol{\delta}^{0,c}, \boldsymbol{\delta}^{0,C} : P \to \mathbb{R}^{n_x}$ be defined by $\boldsymbol{\delta}^{0,c}(\mathbf{p}) = \boldsymbol{\delta}^L$ and $\boldsymbol{\delta}^{0,C}(\mathbf{p}) = \boldsymbol{\delta}^U$ for all $\mathbf{p} \in P$ and $\boldsymbol{\sigma}_\delta^{0,c}(\mathbf{p}) = \boldsymbol{\sigma}_\delta^{0,C}(\mathbf{p}) = \mathbf{0}$, for all $\mathbf{p} \in P$. Let $\hat{\boldsymbol{\sigma}}_{\mathbf{A}}^c, \hat{\boldsymbol{\sigma}}_{\mathbf{A}}^C : P \to \mathbb{R}^{n_p \times n_x \times n_x}$ be subgradients of $\mathbf{A}^c, \mathbf{A}^C$ on P, respectively. Similarly, let $\boldsymbol{\sigma}_{\mathbf{b}}^c, \boldsymbol{\sigma}_{\mathbf{b}}^C : P \to \mathbb{R}^{n_p \times n_x}$ be subgradients of $\mathbf{b}^c, \mathbf{b}^C$ on P, respectively. Let relaxations of $\boldsymbol{\delta}$, $(\boldsymbol{\delta}^{k,c}, \boldsymbol{\delta}^{k,C})$, be given by Theorem 3.12. Let $\mathcal{S}_{\mathbf{u_f}}, \mathcal{S}_{\mathbf{o_f}}$ be composite subgradients of $\mathbf{u_f}$ and $\mathbf{o_f}$ on $B \times A \times \Delta \times P$, respectively. Then the sequences $\{\boldsymbol{\sigma}_\delta^{k+1,c}\}$ and $\{\boldsymbol{\sigma}_\delta^{k+1,C}\}$ defined by*

$$\begin{aligned}\boldsymbol{\sigma}_\delta^{k+1,c}(\cdot) := \mathcal{S}_{\bar{\mathbf{u}}_\mathbf{f}}(&\mathbf{b}^c(\cdot), \mathbf{b}^C(\cdot), \boldsymbol{\sigma}_{\mathbf{b}}^c(\cdot), \boldsymbol{\sigma}_{\mathbf{b}}^C(\cdot), \mathbf{A}^c(\cdot), \mathbf{A}^C(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{A}}^c(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{A}}^C(\cdot), \boldsymbol{\delta}^{k,c}(\cdot), \boldsymbol{\delta}^{k,C}(\cdot), \boldsymbol{\sigma}_\delta^{k,c}(\cdot),\\ &\boldsymbol{\sigma}_\delta^{k,C}(\cdot)),\\ \boldsymbol{\sigma}_\delta^{k+1,C}(\cdot) := \mathcal{S}_{\bar{\mathbf{o}}_\mathbf{f}}(&\mathbf{b}^c(\cdot), \mathbf{b}^C(\cdot), \boldsymbol{\sigma}_{\mathbf{b}}^c(\cdot), \boldsymbol{\sigma}_{\mathbf{b}}^C(\cdot), \mathbf{A}^c(\cdot), \mathbf{A}^C(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{A}}^c(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{A}}^C(\cdot), \boldsymbol{\delta}^{k,c}(\cdot), \boldsymbol{\delta}^{k,C}(\cdot), \boldsymbol{\sigma}_\delta^{k,c}(\cdot),\\ &\boldsymbol{\sigma}_\delta^{k,C}(\cdot))\end{aligned}$$

are subgradients of $\boldsymbol{\delta}^{k+1,c}$ and $\boldsymbol{\delta}^{k+1,C}$ on P, respectively, with $\mathcal{S}_{\bar{\mathbf{u}}_\mathbf{f}}$ and $\mathcal{S}_{\bar{\mathbf{o}}_\mathbf{f}}$ defined analogously to Definition 3.5.

Proof The proof is analogous to that for Theorem 3.5. ■

3.4 *Relaxations of solutions of parametric nonlinear systems*

As in Section 3.2, the general form of $\mathbf{h}$ will be considered such that $\mathbf{h}$ cannot be rearranged algebraically as in Section 3.1.

Assumption 3.14

(1) There exists $\mathbf{x} : P \to D_x$ such that $\mathbf{h}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$, $\forall \mathbf{p} \in P$, and an interval $X \equiv [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{I}D_x$ is available such that $\mathbf{x}(P) \subset X$ and $\mathbf{x}(\mathbf{p})$ is unique in X for all $\mathbf{p} \in P$.
(2) Derivative information $\nabla_\mathbf{x} h_i$, $i = 1, \ldots, n_x$ is available and is factorable, say by automatic differentiation [2,8].
(3) A matrix $\mathbf{Y} \in \mathbb{R}^{n_x \times n_x}$ is known such that $M \equiv \mathbf{Y} J_\mathbf{x} H(X, P)$ satisfies $0 \notin M_{ii}$ for all i, where $J_\mathbf{x} H$ is an inclusion monotonic interval extension of $\mathbf{J_x h}$ on $X \times P$.

The matrix M can be calculated by taking natural interval extensions [20,24]. Furthermore, parametric interval-Newton methods [9,10,24,34] offer a way to calculate X satisfying Assumption 3.14. The matrix $\mathbf{Y}$ is simply a *preconditioning matrix* and has been the topic of many articles. Specifically, Kearfott [16] discusses the application to interval-Newton methods. A frequently valid choice is $\mathbf{Y} = [m(J_\mathbf{x} H(X, P))]^{-1}$, which is popular due to its relatively efficient computation. As in Section 3.3, we begin by characterizing $\mathbf{x}$ in semi-explicit form.

LEMMA 3.15 *Choose any* $\mathbf{z} : P \to \mathbb{R}^{n_x}$ *such that* $\mathbf{z}(P) \subset X$. *There exists a matrix-valued function* $\mathbf{M} : P \to M$ *such that*

$$-\mathbf{Y}\mathbf{h}(\mathbf{z}(\mathbf{p}), \mathbf{p}) = \mathbf{M}(\mathbf{p})(\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})), \quad \forall\, \mathbf{p} \in P$$

with $M \equiv \mathbf{Y} J_{\mathbf{x}} H(X, P)$.

Proof From the parametric mean-value Theorem 2.5, there exists a function $\mathbf{y}^i : P \to X$ such that

$$h_i(\mathbf{x}(\mathbf{p}), \mathbf{p}) - h_i(\mathbf{z}(\mathbf{p}), \mathbf{p}) = \nabla_{\mathbf{x}} h_i(\mathbf{y}^i(\mathbf{p}), \mathbf{p})^{\mathrm{T}} (\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})), \quad \forall\, \mathbf{p} \in P$$

for the ith component of $\mathbf{h}$. Writing the mean-value form for $i = 1, \ldots, n_x$, and noticing that $h_i(\mathbf{x}(\mathbf{p}), \mathbf{p}) = 0$ for all i, we get

$$-\mathbf{h}(\mathbf{z}(\mathbf{p}), \mathbf{p}) = \begin{bmatrix} \nabla_{\mathbf{x}} h_1(\mathbf{y}^1(\mathbf{p}), \mathbf{p})^{\mathrm{T}} \\ \nabla_{\mathbf{x}} h_2(\mathbf{y}^2(\mathbf{p}), \mathbf{p})^{\mathrm{T}} \\ \vdots \\ \nabla_{\mathbf{x}} h_{n_x}(\mathbf{y}^{n_x}(\mathbf{p}), \mathbf{p})^{\mathrm{T}} \end{bmatrix} (\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})), \quad \forall\, \mathbf{p} \in P.$$

Multiplying both sides by $\mathbf{Y}$, we get

$$-\mathbf{Y}\mathbf{h}(\mathbf{z}(\mathbf{p}), \mathbf{p}) = \mathbf{Y} \begin{bmatrix} \nabla_{\mathbf{x}} h_1(\mathbf{y}^1(\mathbf{p}), \mathbf{p})^{\mathrm{T}} \\ \nabla_{\mathbf{x}} h_2(\mathbf{y}^2(\mathbf{p}), \mathbf{p})^{\mathrm{T}} \\ \vdots \\ \nabla_{\mathbf{x}} h_{n_x}(\mathbf{y}^{n_x}(\mathbf{p}), \mathbf{p})^{\mathrm{T}} \end{bmatrix} (\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})), \quad \forall\, \mathbf{p} \in P.$$

Let $\mathbf{B} : X \times X \times \cdots \times X \times P \to \mathbb{R}^{n_x \times n_x}$ be defined so that

$$\mathbf{M}(\cdot) = \mathbf{B}(\mathbf{y}^1(\cdot), \mathbf{y}^2(\cdot), \ldots, \mathbf{y}^{n_x}(\cdot), \cdot) \equiv \mathbf{Y} \begin{bmatrix} \nabla_{\mathbf{x}} h_1(\mathbf{y}^1(\cdot), \cdot)^{\mathrm{T}} \\ \nabla_{\mathbf{x}} h_2(\mathbf{y}^2(\cdot), \cdot)^{\mathrm{T}} \\ \vdots \\ \nabla_{\mathbf{x}} h_{n_x}(\mathbf{y}^{n_x}(\cdot), \cdot)^{\mathrm{T}} \end{bmatrix}.$$

By Assumption 3.14(3), there exists a matrix $\mathbf{Y}$ so that $M \equiv \mathbf{Y} J_{\mathbf{x}} H(X, P)$ is such that $0 \notin M_{ii}$. Since $\mathbf{y}^i(P) \subset X$ holds and the image $\mathbf{B}(X, X, \ldots, X, P)$ is such that $\mathbf{B}(X, X, \ldots, X, P) \subset \mathbf{Y} J_{\mathbf{x}} H(X, P)$ holds, then $\mathbf{M}(P) \subset M$. ■

It is important to notice that, for the purposes of this paper, $\mathbf{M}$ need not be calculated explicitly. However, it is required that convex and concave relaxations of $\mathbf{M}$ on P can be calculated. This is fortuitous since it is easier to relax $\mathbf{M}$ than calculate $\mathbf{M}$ explicitly.

LEMMA 3.16 *Let* $\mathbf{z}$, $\mathbf{M}$, *and* $\mathbf{B}$ *be as in Lemma 3.15. Let* $\mathbf{u}_{\mathbf{B}}, \mathbf{o}_{\mathbf{B}}$ *be composite relaxations of* $\mathbf{B}$ *on* $X \times X \times \cdots \times X \times X \times P$. *Let* $\mathbf{x}^c, \mathbf{x}^C : P \to \mathbb{R}^{n_x}$ *be convex and concave relaxations of* $\mathbf{x}$ *on* P, *respectively, such that* $\mathbf{x}^c(\mathbf{p}) \leq \mathbf{z}(\mathbf{p}) \leq \mathbf{x}^C(\mathbf{p})$, $\forall \mathbf{p} \in P$. *Then the functions*

$$\mathbf{M}^c(\cdot) \equiv \mathbf{u}_{\mathbf{B}}(\mathbf{x}^c(\cdot), \mathbf{x}^C(\cdot), \ldots, \mathbf{x}^c(\cdot), \mathbf{x}^C(\cdot), \cdot)$$
$$\mathbf{M}^C(\cdot) \equiv \mathbf{o}_{\mathbf{B}}(\mathbf{x}^c(\cdot), \mathbf{x}^C(\cdot), \ldots, \mathbf{x}^c(\cdot), \mathbf{x}^C(\cdot), \cdot),$$

are convex and concave relaxations of $\mathbf{M}$ *on* P, *respectively.*

Proof By Assumption 3.14(2), $\nabla_{\mathbf{x}} h_i$, $i = 1, \ldots, n_x$, is available and factorable. We know that for each $\mathbf{p} \in P$ and all $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_x$, either $x_j(\mathbf{p}) \leq y_j^i(\mathbf{p}) \leq z_j(\mathbf{p})$ or $z_j(\mathbf{p}) \leq y_j^i(\mathbf{p}) \leq x_j(\mathbf{p})$. Also, we have valid relaxations such that $\mathbf{x}^c(\mathbf{p}) \leq \mathbf{z}(\mathbf{p}) \leq \mathbf{x}^C(\mathbf{p})$, $\forall \mathbf{p} \in P$. Thus, it is clear $\mathbf{x}^c(\mathbf{p}) \leq \mathbf{y}^i(\mathbf{p}) \leq \mathbf{x}^C(\mathbf{p})$, $\forall \mathbf{p} \in P$ and $i = 1, \ldots, n_x$. Since $\mathbf{x}^c$ and $\mathbf{x}^C$ are convex and concave relaxations of $\mathbf{y}^i$ on P for $i = 1, \ldots, n_x$ by Definition 2.16, and $\mathbf{u_B}$ and $\mathbf{o_B}$ are composite relaxations of $\mathbf{B}$ on $X \times X \times \cdots \times X \times X \times P$, it follows directly that $\mathbf{M}^c$ and $\mathbf{M}^C$ are valid convex and concave relaxations of $\mathbf{M}$ on P. ■

Two different techniques for constructing relaxations of solutions of parametric nonlinear systems, that rely on the above results, will now be presented along with very general composite relaxation results. The complete results and procedures regarding constructing relaxations of solutions of parametric nonlinear systems will then be presented.

DEFINITION 3.17 ($\boldsymbol{\psi}$) *Let* $\mathbf{b} : X \times P \to \mathbb{R}^{n_x}$ *such that* $\mathbf{b} \equiv \mathbf{Yh}$. *Define the function* $\boldsymbol{\psi} : X \times M \times X \times P \to \mathbb{R}^{n_x}$ *such that* $\forall (\tilde{\mathbf{z}}, \tilde{\mathbf{M}}, \tilde{\mathbf{x}}, \mathbf{p}) \in X \times M \times X \times P$, $\boldsymbol{\psi}(\tilde{\mathbf{z}}, \tilde{\mathbf{M}}, \tilde{\mathbf{x}}, \mathbf{p}) = \tilde{\mathbf{x}}^*$, *where the ith component of* $\tilde{\mathbf{x}}^*$ *is given by the loop*:

$$\texttt{for} \quad i = 1, \ldots, n_x \ \texttt{do}$$

$$\tilde{x}_i^* := \tilde{z}_i - \frac{(b_i(\tilde{\mathbf{z}}, \mathbf{p}) + \sum_{j<i} \tilde{m}_{ij}(\tilde{x}_j^* - \tilde{z}_j) + \sum_{j>i} \tilde{m}_{ij}(\tilde{x}_j - \tilde{z}_j))}{\tilde{m}_{ii}}, \tag{13}$$

$$\texttt{end.}$$

The reader should note that this is simply a formal definition of a single iteration of the parametric version of the Gauss–Seidel method if $\tilde{\mathbf{M}}$ was taken as the Jacobian matrix. If $\tilde{\mathbf{M}}$ was taken to be $\mathbf{M}$ as in Lemma 3.15, this is a semi-explicit characterization of the implicit function $\mathbf{x}$ through its mean-value form. This characterization is very closely related to the function $\mathbf{f}$ from the linear systems section above. The following result shows that if relaxations of $\mathbf{x}$ are known, they can be refined. Later, the full method, that is practical computationally, for refining relaxations of $\mathbf{x}$ iteratively is presented which relies on this result.

THEOREM 3.18 *Let* $\mathbf{z}$ *and* $\mathbf{M}$ *be as in Lemma 3.15. Let* $\mathbf{M}^c, \mathbf{M}^C : P \to \mathbb{R}^{n_x \times n_x}$ *be relaxations of* $\mathbf{M}$ *on* P, *let* $\mathbf{x}^{k,c}, \mathbf{x}^{k,C} : P \to \mathbb{R}^{n_x}$ *be relaxations of* $\mathbf{x}$ *on* P, *and let* $\mathbf{z}^c, \mathbf{z}^C : P \to \mathbb{R}^{n_x}$ *be relaxations of* $\mathbf{z}$ *on* P. *Let* $\mathbf{u}_\psi$ *and* $\mathbf{o}_\psi$ *be composite relaxations of* $\boldsymbol{\psi}$ *on* $X \times M \times X \times P$. *Then convex and concave relaxations of* $\mathbf{x}$ *on* P *are given by*

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$

respectively, with $\bar{\mathbf{u}}_\psi$ *and* $\bar{\mathbf{o}}_\psi$ *defined analogously to Definition 3.2.*

Proof Similar to the linear systems result above, we will show that $\mathbf{x}$ is a fixed-point of $\boldsymbol{\psi}$. By Lemma 3.15

$$\mathbf{M}(\mathbf{p})(\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})) = -\mathbf{Yh}(\mathbf{z}(\mathbf{p}), \mathbf{p}), \quad \forall \mathbf{p} \in P,$$

and $0 \notin M_{ii} \supset m_{ii}(P)$, $\forall i$. Now, it is clear that, for $i = 1, \ldots, n_x$, we can write

$$x_i(\mathbf{p}) = z_i(\mathbf{p}) - \frac{(b_i(\mathbf{z}(\mathbf{p}), \mathbf{p}) + \sum_{j<i} m_{ij}(\mathbf{p})(x_j(\mathbf{p}) - z_j(\mathbf{p})) + \sum_{j>i} m_{ij}(\mathbf{p})(x_j(\mathbf{p}) - z_j(\mathbf{p})))}{m_{ii}(\mathbf{p})}$$

with $\mathbf{b} = \mathbf{Yh}$. It immediately follows that

$$\psi_1(\mathbf{z}(\mathbf{p}), \mathbf{M}(\mathbf{p}), \mathbf{x}(\mathbf{p}), \mathbf{p}) = x_1^*(\mathbf{p}) = z_1(\mathbf{p}) - \frac{(b_1(\mathbf{z}(\mathbf{p}), \mathbf{p}) + \sum_{j>1} m_{1j}(\mathbf{p})(x_j(\mathbf{p}) - z_j(\mathbf{p})))}{m_{11}(\mathbf{p})}$$
$$= x_1(\mathbf{p}).$$

Similar to the proof of Lemma 3.11, using induction, $x_i(\mathbf{p}) = \psi_i(\mathbf{z}(\mathbf{p}), \mathbf{M}(\mathbf{p}), \mathbf{x}(\mathbf{p}), \mathbf{p}) = x_i^*$, $\forall i$. Therefore $\mathbf{x}$ is a fixed-point of $\boldsymbol{\psi}$ for every $\mathbf{p} \in P$. From the hypothesis and Definition 2.16, it follows that

$$\mathbf{u}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\mathbf{o}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$$

are relaxations of $\boldsymbol{\psi}(\mathbf{z}(\cdot), \mathbf{M}(\cdot), \mathbf{x}(\cdot), \cdot)$ on P that are also relaxations of $\mathbf{x}(\cdot) = \boldsymbol{\psi}(\mathbf{z}(\cdot), \mathbf{M}(\cdot), \mathbf{x}(\cdot), \cdot)$ on P. It immediately follows that

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$

are convex and concave relaxations of $\mathbf{x}$ on P, respectively. ■

As in Section 3.3, the sequential componentwise refinement of relaxations of $\mathbf{x}$ enable the calculations of subsequent components ($j > i$) to make use of the newly calculated refinements of the previous components ($j < i$).

THEOREM 3.19 *Let $\mathbf{z}$ and $\mathbf{M}$ be as in Lemma 3.15. Let $\mathbf{M}^c, \mathbf{M}^C : P \to \mathbb{R}^{n_x \times n_x}$ be relaxations of $\mathbf{M}$ on P, let $\mathbf{x}^{k,c}, \mathbf{x}^{k,C} : P \to \mathbb{R}^{n_x}$ be relaxations of $\mathbf{x}$ on P, and let $\mathbf{z}^c, \mathbf{z}^C : P \to \mathbb{R}^{n_x}$ be relaxations of $\mathbf{z}$ on P. Let $\hat{\boldsymbol{\sigma}}_{\mathbf{M}}^c, \hat{\boldsymbol{\sigma}}_{\mathbf{M}}^C : P \to \mathbb{R}^{n_p \times n_x \times n_x}$ be subgradients of $\mathbf{M}^c$ and $\mathbf{M}^C$ on P, respectively, let $\boldsymbol{\sigma}_{\mathbf{x}}^{k,c}, \boldsymbol{\sigma}_{\mathbf{x}}^{k,C} : P \to \mathbb{R}^{n_p \times n_x}$ be subgradients of $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$ on P, respectively, and let $\boldsymbol{\sigma}_{\mathbf{z}}^c, \boldsymbol{\sigma}_{\mathbf{z}}^C : P \to \mathbb{R}^{n_p \times n_x}$ be subgradients of $\mathbf{z}^c$ and $\mathbf{z}^C$ on P, respectively. Let $\mathcal{S}_{\mathbf{u}_\psi}$ and $\mathcal{S}_{\mathbf{o}_\psi}$ be composite subgradients of $\mathbf{u}_\psi$ and $\mathbf{o}_\psi$ on $X \times M \times X \times P$, respectively. Then we have*

$$\boldsymbol{\sigma}_{\mathbf{x}}^{k+1,c}(\cdot) := \mathcal{S}_{\bar{\mathbf{u}}_\psi}(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \boldsymbol{\sigma}_{\mathbf{z}}^c(\cdot), \boldsymbol{\sigma}_{\mathbf{z}}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{M}}^c(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{M}}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_{\mathbf{x}}^{k,c}(\cdot),$$
$$\boldsymbol{\sigma}_{\mathbf{x}}^{k,C}(\cdot), \cdot),$$
$$\boldsymbol{\sigma}_{\mathbf{x}}^{k,C}(\cdot) := \mathcal{S}_{\bar{\mathbf{o}}_\psi}(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \boldsymbol{\sigma}_{\mathbf{z}}^c(\cdot), \boldsymbol{\sigma}_{\mathbf{z}}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{M}}^c(\cdot), \hat{\boldsymbol{\sigma}}_{\mathbf{M}}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_{\mathbf{x}}^{k,c}(\cdot),$$
$$\boldsymbol{\sigma}_{\mathbf{x}}^{k,C}(\cdot), \cdot)$$

are subgradients of

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$$
$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\psi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)$$

on P, with $\mathcal{S}_{\bar{\mathbf{u}}_\mathbf{f}}$ and $\mathcal{S}_{\bar{\mathbf{o}}_\mathbf{f}}$ defined analogously to Definition 3.3.

Proof The proof is analogous to that for Theorem 3.5. ■

A second technique for constructing relaxations of solutions of parametric nonlinear systems will now be presented.

DEFINITION 3.20 ($\boldsymbol{\chi}$) *The function* $\boldsymbol{\chi} : X \times M \times X \times P \to \mathbb{R}^{n_x}$ *will be defined as*

$$\boldsymbol{\chi}(\tilde{\mathbf{z}}, \tilde{\mathbf{M}}, \tilde{\mathbf{x}}, \mathbf{p}) \equiv \tilde{\mathbf{z}} - \mathbf{Yh}(\tilde{\mathbf{z}}, \mathbf{p}) + (\mathbf{I} - \tilde{\mathbf{M}})(\tilde{\mathbf{x}} - \tilde{\mathbf{z}}), \tag{14}$$

$\forall(\tilde{\mathbf{z}}, \tilde{\mathbf{M}}, \tilde{\mathbf{x}}, \mathbf{p}) \in X \times M \times X \times P$.

THEOREM 3.21 *Let* $\mathbf{z}$ *and* $\mathbf{M}$ *be as in Lemma 3.15. Let* $\mathbf{M}^c, \mathbf{M}^C : P \to \mathbb{R}^{n_x \times n_x}$ *be relaxations of* $\mathbf{M}$ *on* P*, let* $\mathbf{x}^{k,c}, \mathbf{x}^{k,C} : P \to \mathbb{R}^{n_x}$ *be relaxations of* $\mathbf{x}$ *on* P*, and let* $\mathbf{z}^c, \mathbf{z}^C : P \to \mathbb{R}^{n_x}$ *be relaxations of* $\mathbf{z}$ *on* P*. Let* $\mathbf{u}_\chi$ *and* $\mathbf{o}_\chi$ *be composite relaxations of* $\boldsymbol{\chi}$ *on* $X \times M \times X \times P$*. Then convex and concave relaxations of* $\mathbf{x}$ *on* P *are given by*

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\chi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\chi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$

respectively, with $\bar{\mathbf{u}}_\chi$ *and* $\bar{\mathbf{o}}_\chi$ *defined analogously to Definition 3.2.*

Proof First, we will show that $\mathbf{x}$ is a fixed-point of $\boldsymbol{\chi}$. By Proposition 3.7, we can write $\boldsymbol{\phi}(\mathbf{w}, \mathbf{p}) = \mathbf{w} - \mathbf{Yh}(\mathbf{w}, \mathbf{p})$ so that $\boldsymbol{\phi}(\mathbf{w}, \mathbf{p}) = \mathbf{w} \Leftrightarrow \mathbf{h}(\mathbf{w}, \mathbf{p}) = \mathbf{0}$. Now,

$$\begin{aligned}\boldsymbol{\phi}(\mathbf{x}(\mathbf{p}), \mathbf{p}) &= \boldsymbol{\phi}(\mathbf{x}(\mathbf{p}), \mathbf{p}) + \boldsymbol{\phi}(\mathbf{z}(\mathbf{p}), \mathbf{p}) - \boldsymbol{\phi}(\mathbf{z}(\mathbf{p}), \mathbf{p}),\\ &= \mathbf{x}(\mathbf{p}) - \mathbf{Yh}(\mathbf{x}(\mathbf{p}), \mathbf{p}) + \mathbf{z}(\mathbf{p}) - \mathbf{Yh}(\mathbf{z}(\mathbf{p}), \mathbf{p}) - \mathbf{z}(\mathbf{p}) + \mathbf{Yh}(\mathbf{z}(\mathbf{p}), \mathbf{p}),\\ &= \mathbf{z}(\mathbf{p}) - \mathbf{Yh}(\mathbf{z}(\mathbf{p}), \mathbf{p}) + (\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})) - \mathbf{Y}(\mathbf{h}(\mathbf{x}(\mathbf{p}), \mathbf{p}) - \mathbf{h}(\mathbf{z}(\mathbf{p}), \mathbf{p})),\end{aligned}$$

for all $\mathbf{p} \in P$. From the definition of $\mathbf{M}$ and $\mathbf{z}$, $\mathbf{Y}(\mathbf{h}(\mathbf{x}(\mathbf{p}), \mathbf{p}) - \mathbf{h}(\mathbf{z}(\mathbf{p}), \mathbf{p})) = \mathbf{M}(\mathbf{p})(\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p}))$ holds. Substituting in we get

$$\begin{aligned}\boldsymbol{\phi}(\mathbf{x}(\mathbf{p}), \mathbf{p}) &= \mathbf{z}(\mathbf{p}) - \mathbf{Yh}(\mathbf{z}(\mathbf{p}), \mathbf{p}) + (\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})) - \mathbf{M}(\mathbf{p})(\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})),\\ &= \mathbf{z}(\mathbf{p}) - \mathbf{Yh}(\mathbf{z}(\mathbf{p}), \mathbf{p}) + (\mathbf{I} - \mathbf{M}(\mathbf{p}))(\mathbf{x}(\mathbf{p}) - \mathbf{z}(\mathbf{p})),\\ &= \boldsymbol{\chi}(\mathbf{z}(\mathbf{p}), \mathbf{M}(\mathbf{p}), \mathbf{x}(\mathbf{p})).\end{aligned}$$

Thus, $\mathbf{x}(\mathbf{p}) = \boldsymbol{\phi}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \boldsymbol{\chi}(\mathbf{z}(\mathbf{p}), \mathbf{M}(\mathbf{p}), \mathbf{x}(\mathbf{p}))$. From the hypothesis and by Definition 2.16, it follows that

$$\mathbf{u}_\chi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\mathbf{o}_\chi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$

are relaxations of $\boldsymbol{\chi}(\mathbf{z}(\cdot), \mathbf{M}(\cdot), \mathbf{x}(\cdot), \cdot)$ on P that are also relaxations of $\mathbf{x}(\cdot) = \boldsymbol{\chi}(\mathbf{z}(\cdot), \mathbf{M}(\cdot), \mathbf{x}(\cdot), \cdot)$ on P. It immediately follows that

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_\chi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$
$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_\chi(\mathbf{z}^c(\cdot), \mathbf{z}^C(\cdot), \mathbf{M}^c(\cdot), \mathbf{M}^C(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot),$$

are convex and concave relaxations of $\mathbf{x}$ on P, respectively. ■

Remark 6 Similar to how $\boldsymbol{\psi}$, from above, and $\mathbf{f}$ from Section 3.3 were defined, it is easy to rearrange $\boldsymbol{\chi}$ to be calculated in a sequential componentwise fashion.

Remark 7 The subgradient result for $\boldsymbol{\psi}$, Theorem 3.19, trivially holds with $\boldsymbol{\psi}$ replaced by $\boldsymbol{\chi}$.

One hypothesis that the above results rely upon is the existence of an appropriate function $\mathbf{z} : P \to X$, for which relaxations are readily available. Without such a function, convex and concave relaxations of $\mathbf{x}$ that are potential improvements on the initial bounds cannot be calculated. This issue is addressed next.

DEFINITION 3.22 ($\mathbf{z}$) *Let $\mathbf{x}^a, \mathbf{x}^A : P \to \mathbb{R}^{n_x}$ be any affine relaxations of $\mathbf{x}$ on P, respectively. For some $\lambda \in [0,1]$ define the function $\mathbf{z} : P \to \mathbb{R}^{n_x}$ with the following procedure*:

$$
\begin{aligned}
&\texttt{for}\quad i = 1, \ldots, n_x \ \texttt{do}\\
&\quad \xi_i(\cdot) := \lambda x_i^a(\cdot) + (1-\lambda) x_i^A(\cdot)\\
&\quad \Xi_i := [\xi_i^L, \xi_i^U] = \left[\min_{\mathbf{p}\in P} \xi_i(\mathbf{p}), \max_{\mathbf{p}\in P} \xi_i(\mathbf{p}) \right]\\
&\quad \texttt{if}\ \xi_i^L < x_i^L\ \texttt{then}\\
&\qquad \hat{x}_i^a(\cdot) := x_i^L,\ \texttt{else}\ \hat{x}_i^a(\cdot) := x_i^a(\cdot)\\
&\quad \texttt{if}\ \xi_i^U > x_i^U\ \texttt{then}\\
&\qquad \hat{x}_i^A(\cdot) := x_i^U,\ \texttt{else}\ \hat{x}_i^A(\cdot) := x_i^A(\cdot)\\
&\quad z_i(\cdot) := \lambda \hat{x}_i^a(\cdot) + (1-\lambda)\hat{x}_i^A(\cdot)\\
&\texttt{end}
\end{aligned}
$$

It should be noted that the interval $\Xi_i = [\min_{\mathbf{p}\in P} \xi(\mathbf{p}), \max_{\mathbf{p}\in P} \xi(\mathbf{p})]$ can be calculated easily and efficiently for each i using interval analysis. Also, defining $\mathbf{z}$ to be affine is important because affine functions are trivially convex *and* concave, so that the calculation of valid relaxations is trivial.

LEMMA 3.23 *Suppose $\mathbf{x}^a, \mathbf{x}^A : P \to \mathbb{R}^{n_x}$ are any affine relaxations of $\mathbf{x}$ on P. Then the function $\mathbf{z} : P \to \mathbb{R}^{n_x}$, defined in Definition 3.22, is affine and maps P into X.*

Proof Consider a single i and set $\Xi_i := [\xi_i^L, \xi_i^U]$ as in Definition 3.22. It should be noted that the cases where $x_i^A(\mathbf{p}) \le x_i^L$ and/or $x_i^a(\mathbf{p}) \ge x_i^U$ for any $\mathbf{p} \in P$ cannot occur since, by definition $x_i^a(\mathbf{p}) \le x_i(\mathbf{p}) \le \mathbf{x}_i^A(\mathbf{p})$, $\forall \mathbf{p} \in P$, implying $x_i(\mathbf{p}) \le x_i^L$ and/or $x_i(\mathbf{p}) \ge x_i^U$, violating Assumption 3.14(1). First, consider the case that $x_i^L \le \xi_i^L$ and $\xi_i^U \le x_i^U$. Trivially, $z_i(\cdot) := \lambda x_i^a(\cdot) + (1-\lambda)x_i^A(\cdot)$ satisfies $x_i^L \le z_i(\cdot) \le x_i^U$, $\forall \mathbf{p} \in P$, and thus z_i maps P into X_i and since it is a convex combination of affine functions, it is affine. Next, consider the case that $\xi_i^L < x_i^L$ and $x_i^U < \xi_i^U$. Then $z_i(\cdot) := \lambda x_i^L + (1-\lambda)x_i^U$ maps P into X_i, trivially, and since it is a convex combination of two affine (constant) functions, it is affine. Consider the case that only one bound is violated, say $\xi_i^L < x_i^L$ and $\xi_i^U \le x_i^U$. Then $z_i(\cdot) := \lambda x_i^L + (1-\lambda)x_i^A(\cdot)$ and since x_i^L is affine (constant) and x_i^A is affine, z_i is affine and $x_i^L \le \lambda x_i^L + (1-\lambda)x_i^A(\cdot)$. A similar argument can be made for the case in which the upper bound is violated: $\xi_i^U > x_i^U$ and $x_i^L \le \xi_i^L$. Therefore $\mathbf{z}$ is affine and maps P into X. ■

The `if` statements in Definition 3.22 check, for a particular choice of λ, whether or not the hyperplanes defined by $\lambda\mathbf{x}^a(\cdot) + (1-\lambda)\mathbf{x}^A(\cdot)$, will violate the bounds on X for some ith component. If that is the case, the hyperplane is calculated so as to not violate the bounds on X. A convenient choice for the ith hyperplane is simply the plane that lies in the middle corresponding to $\lambda = 0.5$. Other choices for $\mathbf{z}$ exist. For instance, in one dimension, the function z can be taken to be the secant connecting the endpoints $x(p^L)$ and $x(p^U)$. The above result together with the

definition of the composite subgradient (Definition 2.19), offers an automatic way to calculate $\mathbf{z}$ that is valid for *all* systems in general. In order to simplify the notation for later results, the following procedure will be defined.

Subroutine 3.24 (`Aff`*)*

$$
\begin{aligned}
&\texttt{Aff}(\mathbf{c}, \mathbf{C}, \boldsymbol{\sigma}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{C}}, \lambda, X, P, \bar{\mathbf{p}})\{\\
&\texttt{for} \quad i = 1, \ldots, n_x \ \texttt{do}\\
&\qquad X_i^a := c_i + \sum_{j=1}^{n_p} (\boldsymbol{\sigma}_{\mathbf{c}}{}^{\mathrm{T}})_{ij} (P_j - \bar{p}_j)\\
&\qquad X_i^A := C_i + \sum_{j=1}^{n_p} (\boldsymbol{\sigma}_{\mathbf{C}}{}^{\mathrm{T}})_{ij} (P_j - \bar{p}_j)\\
&\qquad \Xi_i := \lambda X_i^a + (1-\lambda) X_i^A\\
&\qquad \text{if } \ \xi_i^L < x_i^L \ \texttt{then}\\
&\qquad\qquad (\boldsymbol{\sigma}_{\mathbf{c}})_{ji} := 0, \forall j = 1, \ldots, n_p\\
&\qquad\qquad c_i := x_i^L\\
&\qquad \text{if } \ \xi_i^U > x_i^U \ \texttt{then}\\
&\qquad\qquad (\boldsymbol{\sigma}_{\mathbf{C}})_{ji} := 0, \forall j = 1, \ldots, n_p\\
&\qquad\qquad C_i := x_i^U\\
&\texttt{end}\\
&\texttt{return} \ \{\mathbf{c}, \mathbf{C}, \boldsymbol{\sigma}_c, \boldsymbol{\sigma}_C\}\\
&\}
\end{aligned}
$$

Remark 8 Note that the first three computations in Subroutine 3.24 are interval computations performed using interval analysis.

Remark 9 The reader is reminded that, by Definition 3.3 and the previous definitions of subgradients, $\bar{\mathbf{p}}$ is the point at which the subgradient information is calculated; referred to as the reference point. The choice of this point is arbitrary from a theoretical point-of-view provided that it satisfies certain properties. From an application point-of-view, the choice may result in tighter or weaker relaxations and is left to the user to modify as a *tuning* parameter.

THEOREM 3.25 *Let* $\mathbf{x}^{0,c}, \mathbf{x}^{0,C} : P \to \mathbb{R}^{n_x}$ *be defined as* $\mathbf{x}^{0,c}(\mathbf{p}) = \mathbf{x}^L$ *and* $\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{x}^U$ *for every* $\mathbf{p} \in P$. *Let* $\boldsymbol{\sigma}_{\mathbf{x}}^{0,c}, \boldsymbol{\sigma}_{\mathbf{x}}^{0,C} : P \to \mathbb{R}^{n_p \times n_x}$ *be defined as* $\boldsymbol{\sigma}_{\mathbf{x}}^{0,c}(\mathbf{p}), \boldsymbol{\sigma}_{\mathbf{x}}^{0,C}(\mathbf{p}) = \mathbf{0}$ *for every* $\mathbf{p} \in P$. *Let* $\mathbf{u}_{\mathbf{B}}, \mathbf{o}_{\mathbf{B}}$ *be composite relaxations of* $\mathbf{B}$ *on* $X \times \cdots \times X \times P$ *and* $\bar{\mathbf{u}}_{\psi}, \bar{\mathbf{o}}_{\psi}$ *be composite relaxations of* $\boldsymbol{\psi}$ *on* $X \times M \times X \times P$. *Let* $\mathcal{S}_{\mathbf{u}_B}, \mathcal{S}_{\mathbf{o}_B}$ *be composite subgradients of* $\mathbf{u}_{\mathbf{B}}, \mathbf{o}_{\mathbf{B}}$, *respectively. Then, for any choice of* $\{\bar{\mathbf{p}}^k\}$, *and* $\{\lambda^k\}$ *with* $\bar{\mathbf{p}}^k \in P$ *and* $\lambda^k \in [0,1]$ *for* $k \in \mathbb{N}$, *the elements of the sequences* $\{\mathbf{x}^{k,c}\}$ *and* $\{\mathbf{x}^{k,C}\}$ *defined by the iteration*:

$$
\begin{aligned}
(\mathbf{c}, \mathbf{C}, \boldsymbol{\sigma}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{C}}) &:= \texttt{Aff}(\mathbf{x}^{k,c}(\bar{\mathbf{p}}^k), \mathbf{x}^{k,C}(\bar{\mathbf{p}}^k), \boldsymbol{\sigma}_{\mathbf{x}}^{k,c}(\bar{\mathbf{p}}^k), \boldsymbol{\sigma}_{\mathbf{x}}^{k,C}(\bar{\mathbf{p}}^k), \lambda^k, X, P, \bar{\mathbf{p}}^k)\\
\mathbf{x}^{k,a}(\mathbf{p}) &:= \mathbf{c} + (\boldsymbol{\sigma}_{\mathbf{c}})^{\mathrm{T}}(\mathbf{p} - \bar{\mathbf{p}}^k), \quad \forall\, \mathbf{p} \in P
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x}^{k,A}(\mathbf{p}) &:= \mathbf{C} + (\boldsymbol{\sigma}_\mathbf{C})^\mathrm{T}(\mathbf{p} - \bar{\mathbf{p}}^k), \quad \forall\, \mathbf{p} \in P \\
\mathbf{z}^k(\cdot) &:= \lambda^k \mathbf{x}^{k,a}(\cdot) + (1-\lambda^k)\mathbf{x}^{k,A}(\cdot) \\
\boldsymbol{\sigma}_\mathbf{z}^k &:= \lambda^k \boldsymbol{\sigma}_\mathbf{c} + (1-\lambda^k)\boldsymbol{\sigma}_\mathbf{C} \\
\mathbf{M}^{k,c}(\cdot) &:= \mathbf{u}_\mathbf{B}(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \cdot) \\
\mathbf{M}^{k,C}(\cdot) &:= \mathbf{o}_\mathbf{B}(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \cdot) \\
\hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,c}(\cdot) &:= \mathcal{S}_{u_B}(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_\mathbf{c}, \boldsymbol{\sigma}_\mathbf{C}, \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_\mathbf{c}, \boldsymbol{\sigma}_\mathbf{C}, \cdot) \\
\hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,C}(\cdot) &:= \mathcal{S}_{o_B}(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_\mathbf{c}, \boldsymbol{\sigma}_\mathbf{C}, \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_\mathbf{c}, \boldsymbol{\sigma}_\mathbf{C}, \cdot) \\
\mathbf{x}^{k+1,c}(\cdot) &:= \bar{\mathbf{u}}_\psi(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot) \\
\mathbf{x}^{k+1,C}(\cdot) &:= \bar{\mathbf{o}}_\psi(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot) \\
\boldsymbol{\sigma}_\mathbf{x}^{k+1,c}(\cdot) &:= \mathcal{S}_{\bar{\mathbf{u}}_\psi}(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \boldsymbol{\sigma}_\mathbf{z}^k, \boldsymbol{\sigma}_\mathbf{z}^k, \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,c}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \\
&\qquad \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot) \\
\boldsymbol{\sigma}_\mathbf{x}^{k+1,C}(\cdot) &:= \mathcal{S}_{\bar{\mathbf{o}}_\psi}(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \boldsymbol{\sigma}_\mathbf{z}^k, \boldsymbol{\sigma}_\mathbf{z}^k, \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,c}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \\
&\qquad \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)
\end{aligned}
$$

are convex and concave relaxations of $\mathbf{x}$ *on* P*, respectively. Furthermore, the elements of the sequences* $\{\boldsymbol{\sigma}_\mathbf{x}^{k,c}\}$ *and* $\{\boldsymbol{\sigma}_\mathbf{x}^{k,C}\}$ *are subgradients of the elements of the sequences* $\{\mathbf{x}^{k,c}\}$ *and* $\{\mathbf{x}^{k,C}\}$*, respectively, at the reference points* $\{\bar{\mathbf{p}}^k\}$ *for* $k \in \mathbb{N}$.

Proof By definition, $\mathbf{x}^{0,c}$ and $\mathbf{x}^{0,C}$ are, respectively, convex and concave relaxations of $\mathbf{x}$ on P. Similarly, $\boldsymbol{\sigma}_\mathbf{x}^{0,c}$ and $\boldsymbol{\sigma}_\mathbf{x}^{0,C}$ are subgradients of $\mathbf{x}^{0,c}$ and $\mathbf{x}^{0,C}$ on P, respectively. Suppose this holds for arbitrary $k \in \mathbb{N}$. Then $\mathbf{x}^{k,c}$ and $\mathbf{x}^{k,C}$ are, respectively, convex and concave relaxations of $\mathbf{x}$ on P and $\boldsymbol{\sigma}_\mathbf{x}^{k,c}$ and $\boldsymbol{\sigma}_\mathbf{x}^{k,C}$ are subgradients on P. Then it follows from the definition of Subroutine 3.24 that $\mathbf{x}^{k,a}$ and $\mathbf{x}^{k,A}$ are affine relaxations of $\mathbf{x}$ on P. Furthermore, $\mathbf{z}^k$ is affine and maps into X by Lemma 3.23. From the definition of $\mathbf{z}^k$, it is clear that $\mathbf{x}^{k,a}(\mathbf{p}) \leq \mathbf{z}^k(\mathbf{p}) \leq \mathbf{x}^{k,A}(\mathbf{p}), \forall \mathbf{p} \in P$, which implies that $\mathbf{M}^{k,c}$ and $\mathbf{M}^{k,C}$ are relaxations of $\mathbf{M}$ on P by Lemma 3.16. Moreover, $\boldsymbol{\sigma}_\mathbf{c}$ and $\boldsymbol{\sigma}_\mathbf{C}$ are subgradients of $\mathbf{x}^{k,a}$ and $\mathbf{x}^{k,A}$, respectively, so that $\hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,c}$ and $\hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,C}$ are subgradients of $\mathbf{M}^{k,c}$ and $\mathbf{M}^{k,C}$ on P, respectively, by Definition 2.19. By Theorem 3.18,

$$
\begin{aligned}
\mathbf{x}^{k+1,c}(\cdot) &:= \bar{\mathbf{u}}_\psi(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot) \\
\mathbf{x}^{k+1,C}(\cdot) &:= \bar{\mathbf{o}}_\psi(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot)
\end{aligned}
$$

are relaxations of $\mathbf{x}$ on P and by Definition 3.3 and Theorem 3.19

$$
\begin{aligned}
\boldsymbol{\sigma}_\mathbf{x}^{k+1,c}(\cdot) &:= \mathcal{S}_{\bar{\mathbf{u}}_\psi}(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \boldsymbol{\sigma}_\mathbf{z}^k, \boldsymbol{\sigma}_\mathbf{z}^k, \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,c}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \\
&\qquad \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot) \\
\boldsymbol{\sigma}_\mathbf{x}^{k+1,C}(\cdot) &:= \mathcal{S}_{\bar{\mathbf{o}}_\psi}(\mathbf{z}^k(\cdot), \mathbf{z}^k(\cdot), \boldsymbol{\sigma}_\mathbf{z}^k, \boldsymbol{\sigma}_\mathbf{z}^k, \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,c}(\cdot), \hat{\boldsymbol{\sigma}}_\mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \\
&\qquad \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\cdot), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\cdot), \cdot)
\end{aligned}
$$

are subgradients of $\mathbf{x}^{k+1,c}$ and $\mathbf{x}^{k+1,C}$, respectively. Induction completes the proof. ■

Therefore, the iterations outlined in the above theorem can be regarded as methods for potentially refining the calculated bounds on $\mathbf{x}$ or any other initial convex and concave bounds on $\mathbf{x}$.

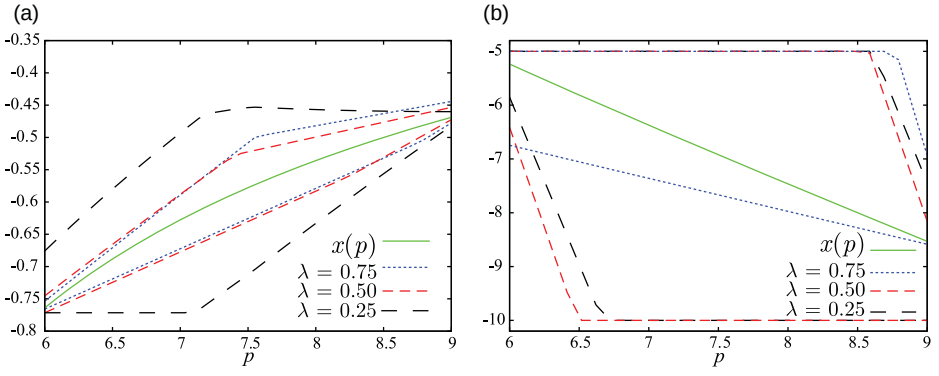

Figure 1. Relaxations of the solution in (a) X^1 and (b) X^2 for the simple example.

However, the above theorem does not guarantee that the calculated convex and concave relaxations will in fact always be improvements on the initial bounds. Nevertheless, the theorem is important because it does offer a way to calculate relaxations that are no worse than the original bounds and potentially tighter, unlike the situation discussed in Theorem 3.8. To illustrate relaxations constructed using this result, the following simple example is given.

Example 3.26 Consider the system $h(z,p) = z^2 + pz + 4$ with $p \in P = [6,9]$. The two real roots are given by the quadratic formula. Using the parametric interval-Newton method [10,24,34], two conservative intervals, $X^1 = [-0.78, -0.4]$ and $X^2 = [-10.0, -5.0]$, were calculated that are guaranteed to each contain a unique solution $x(p)$ such that $h(x(p),p) = 0, \forall p \in P$. Three different z functions were used, each corresponding to a different $\lambda^k = \lambda$ value, and convex and concave relaxations of $x(p)$ were constructed. For each λ, $\bar{p}^k = \bar{p}$ was chosen to be the midpoint of P. Figure 1 shows the relaxations for the two solutions corresponding to each λ value after applying two iterations of the procedure, after which, no significant refinements could be made.

Another method for refining the bounds of an implicit function through McCormick relaxations can be derived from Theorem 3.21. The following method is the analogue to the Krawczyk interval method for bounding solutions of nonlinear systems.

THEOREM 3.27 *Let* $\mathbf{x}^{0,c}, \mathbf{x}^{0,C} : P \to \mathbb{R}^{n_x}$ *be defined as* $\mathbf{x}^{0,c}(\mathbf{p}) = \mathbf{x}^L$ *and* $\mathbf{x}^{0,C}(\mathbf{p}) = \mathbf{x}^U$ *for every* $\mathbf{p} \in P$. *Let* $\boldsymbol{\sigma}_{\mathbf{x}}^{0,c}, \boldsymbol{\sigma}_{\mathbf{x}}^{0,C} : P \to \mathbb{R}^{n_p \times n_x}$ *be defined as* $\boldsymbol{\sigma}_{\mathbf{x}}^{0,c}(\mathbf{p}) = \boldsymbol{\sigma}_{\mathbf{x}}^{0,C}(\mathbf{p}) = \mathbf{0}$ *for every* $\mathbf{p} \in P$. *Let* $\mathbf{u_B}, \mathbf{o_B}$ *be composite relaxations of* $\mathbf{B}$ *on* $X \times \cdots \times X \times P$ *and* $\bar{\mathbf{u}}_\chi, \bar{\mathbf{o}}_\chi$ *be composite relaxations of* $\boldsymbol{\chi}$ *on* $X \times M \times X \times P$. *Let* $\mathcal{S}_{\mathbf{u}_B}, \mathcal{S}_{\mathbf{o}_B}$ *be composite subgradients of* $\mathbf{u_B}, \mathbf{o_B}$, *respectively. Then, for any choice of* $\{\bar{\mathbf{p}}^k\}$, *and* $\{\lambda^k\}$ *with* $\bar{\mathbf{p}}^k \in P$ *and* $\lambda^k \in [0,1]$ *for* $k \in \mathbb{N}$, *the elements of the sequences* $\{\mathbf{x}^{k,c}\}$ *and* $\{\mathbf{x}^{k,C}\}$ *defined by the iteration*:

$$
\begin{aligned}
(\mathbf{c}, \mathbf{C}, \boldsymbol{\sigma}_\mathbf{c}, \boldsymbol{\sigma}_\mathbf{C}) &:= \mathtt{Aff}(\mathbf{x}^{k,c}(\bar{\mathbf{p}}^k), \mathbf{x}^{k,C}(\bar{\mathbf{p}}^k), \boldsymbol{\sigma}_\mathbf{x}^{k,c}(\bar{\mathbf{p}}^k), \boldsymbol{\sigma}_\mathbf{x}^{k,C}(\bar{\mathbf{p}}^k), \lambda^k, X, P, \bar{\mathbf{p}}^k) \\
\mathbf{x}^{k,a}(\mathbf{p}) &:= \mathbf{c} + (\boldsymbol{\sigma}_\mathbf{c})^\mathrm{T}(\mathbf{p} - \bar{\mathbf{p}}^k), \quad \forall\, \mathbf{p} \in P \\
\mathbf{x}^{k,A}(\mathbf{p}) &:= \mathbf{C} + (\boldsymbol{\sigma}_\mathbf{C})^\mathrm{T}(\mathbf{p} - \bar{\mathbf{p}}^k), \quad \forall\, \mathbf{p} \in P \\
\mathbf{z}^k(\cdot) &:= \lambda^k \mathbf{x}^{k,a}(\cdot) + (1 - \lambda^k)\mathbf{x}^{k,A}(\cdot) \\
\boldsymbol{\sigma}_\mathbf{z}^k &:= \lambda^k \boldsymbol{\sigma}_\mathbf{c} + (1 - \lambda^k)\boldsymbol{\sigma}_\mathbf{C} \\
\mathbf{M}^{k,c}(\cdot) &:= \mathbf{u_B}(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \cdot) \\
\mathbf{M}^{k,C}(\cdot) &:= \mathbf{o_B}(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \cdot)
\end{aligned}
$$

$$\hat{\boldsymbol{\sigma}}^{k,c}_{\mathbf{M}}(\cdot) := \mathcal{S}_{u_B}\left(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{C}}, \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{C}}, \cdot\right)$$

$$\hat{\boldsymbol{\sigma}}^{k,C}_{\mathbf{M}}(\cdot) := \mathcal{S}_{o_B}\left(\mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{C}}, \ldots, \mathbf{x}^{k,a}(\cdot), \mathbf{x}^{k,A}(\cdot), \boldsymbol{\sigma}_{\mathbf{c}}, \boldsymbol{\sigma}_{\mathbf{C}}, \cdot\right)$$

$$\mathbf{x}^{k+1,c}(\cdot) := \bar{\mathbf{u}}_{\chi}\left(\mathbf{z}^{k}(\cdot), \mathbf{z}^{k}(\cdot), \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot\right)$$

$$\mathbf{x}^{k+1,C}(\cdot) := \bar{\mathbf{o}}_{\chi}\left(\mathbf{z}^{k}(\cdot), \mathbf{z}^{k}(\cdot), \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \cdot\right)$$

$$\boldsymbol{\sigma}^{k+1,c}_{\mathbf{x}}(\cdot) := \mathcal{S}_{\bar{\mathbf{u}}_{\chi}}\Big(\mathbf{z}^{k}(\cdot), \mathbf{z}^{k}(\cdot), \boldsymbol{\sigma}^{k}_{\mathbf{z}}, \boldsymbol{\sigma}^{k}_{\mathbf{z}}, \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \hat{\boldsymbol{\sigma}}^{k,c}_{\mathbf{M}}(\cdot), \hat{\boldsymbol{\sigma}}^{k,C}_{\mathbf{M}}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}^{k,c}_{\mathbf{x}}(\cdot), \boldsymbol{\sigma}^{k,C}_{\mathbf{x}}(\cdot), \cdot\Big)$$

$$\boldsymbol{\sigma}^{k+1,C}_{\mathbf{x}}(\cdot) := \mathcal{S}_{\bar{\mathbf{o}}_{\chi}}\Big(\mathbf{z}^{k}(\cdot), \mathbf{z}^{k}(\cdot), \boldsymbol{\sigma}^{k}_{\mathbf{z}}, \boldsymbol{\sigma}^{k}_{\mathbf{z}}, \mathbf{M}^{k,c}(\cdot), \mathbf{M}^{k,C}(\cdot), \hat{\boldsymbol{\sigma}}^{k,c}_{\mathbf{M}}(\cdot), \hat{\boldsymbol{\sigma}}^{k,C}_{\mathbf{M}}(\cdot), \mathbf{x}^{k,c}(\cdot), \mathbf{x}^{k,C}(\cdot), \boldsymbol{\sigma}^{k,c}_{\mathbf{x}}(\cdot), \boldsymbol{\sigma}^{k,C}_{\mathbf{x}}(\cdot), \cdot\Big)$$

are convex and concave relaxations of $\mathbf{x}$ *on* P*, respectively, for* $k \in \mathbb{N}$*. Furthermore, the elements of the sequences* $\{\boldsymbol{\sigma}^{k,c}_{\mathbf{x}}\}$ *and* $\{\boldsymbol{\sigma}^{k,C}_{\mathbf{x}}\}$ *are subgradients of the elements of the sequences* $\{\mathbf{x}^{k,c}\}$ *and* $\{\mathbf{x}^{k,C}\}$*, respectively, at the reference points* $\{\bar{\mathbf{p}}^{k}\}$ *for* $k \in \mathbb{N}$*.*

Proof The proof is analogous to the proof of Theorem 3.25. ■

Remark 10 There are many alternative implementations of the iterations in Theorems 3.25 and 3.27. Computationally, evaluating the relaxations constructed using the iterations in Theorems 3.25 and 3.27 can only be done at a single $\mathbf{p}$. In order to accomplish this, relaxations at $\bar{\mathbf{p}}^{k}$ must first be computed. Therefore, one such alternative implementation is to choose a single $\bar{\mathbf{p}}^{k} = \bar{\mathbf{p}} \in P$ and apply one of the iterations to get affine relaxation information, and subsequently, use this information to define the $\mathbf{z}^{k}$ function. With this information calculated up front, the first nine instructions are no longer dependent on the iteration k.

4. Global optimization of implicit functions

The continuous *branch-and-bound* (B&B) framework is a popular algorithm for solving globally nonconvex NLPs as in (1). It is discussed in [11,15] thoroughly. The B&B algorithm relies on refining bounds on the global optima while rigorously ruling out potentially large regions of the search space where global optima are guaranteed not to lie, termed fathoming. The algorithm is guaranteed to terminate in finitely many iterations when ϵ-tolerance has been reached. B&B will be employed here to solve programs with embedded implicit functions, as in (4), in a similar fashion. In fact, the B&B algorithm will be applied to (4) without modifying any of its underlying features or procedures. Therefore, the only difference between the B&B algorithm presented here and the B&B algorithm for standard form global optimization problems, is simply how the functions involved are evaluated and how their relaxations are calculated. Furthermore, due to the required properties of the bounding information on implicit functions, namely that X encloses a unique implicit function, the B&B algorithm presented here will only handle one solution branch at a time. For systems with multiple solution branches, as in Example 3.26, the user has the freedom to decide how the full problem is solved. For instance, for each m solution branches, the B&B algorithm can be called to solve each m problem in a parallel fashion on a multi-core computer. Alternatively, the B&B algorithm could be called to solve each m problems sequentially making use of the best upper bound found and potentially fathoming subsequent problems whose lower bound determined at the root node is greater than the best upper bound (fathoming on value dominance). Before presenting the full B&B algorithm, the NLP subproblems, on which it relies, will be discussed.

4.1 *Upper-bounding problem*

Given a subinterval, P^l, of the decision space P, define the upper-bounding problem:

$$\begin{aligned} \min_{\mathbf{z}\in X, \mathbf{p}\in P^l} \quad & f(\mathbf{z},\mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{z},\mathbf{p}) \le \mathbf{0}, \\ & \mathbf{h}(\mathbf{z},\mathbf{p}) = \mathbf{0}. \end{aligned} \tag{15}$$

This problem is solved locally to obtain a local solution $(\hat{\mathbf{z}}^l, \hat{\mathbf{p}}^l)$, if one exists. Lastly, a valid upper bound on the optimal solution value will be defined as $f_l^{\mathrm{UBD}} \equiv f(\hat{\mathbf{z}}^l, \hat{\mathbf{p}}^l)$.

4.2 *Lower-bounding problem*

Given a subinterval, P^l, of the decision space P, define the lower-bounding problem:

$$\begin{aligned} f_l^{\mathrm{LBD}} = \min_{\mathbf{p}\in P^l} \quad & f^c(\mathbf{p}) = \mathrm{u}_f(\mathbf{x}^c(\mathbf{p}), \mathbf{x}^C(\mathbf{p}), \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{g}^c(\mathbf{p}) = \mathbf{u}_{\mathbf{g}}(\mathbf{x}^c(\mathbf{p}), \mathbf{x}^C(\mathbf{p}), \mathbf{p}) \le \mathbf{0}, \end{aligned} \tag{16}$$

where the composite relaxations u_f and $\mathbf{u}_{\mathbf{g}}$ will be constructed by first using the procedures outlined in Section 3 for constructing convex and concave relaxations of the implicit function $\mathbf{x}$ on P^l and then applying the rules of generalized McCormick relaxations for composition. The lower-bounding problem (16) is convex by construction and is solved to global optimality. Denote the solution found by $\check{\mathbf{p}}$, if it exists, and let $f_l^{\mathrm{LBD}} \equiv \mathrm{u}_f(\mathbf{x}^c(\check{\mathbf{p}}), \mathbf{x}^C(\check{\mathbf{p}}), \check{\mathbf{p}})$.

4.3 *Global optimization algorithm*

The B&B algorithm for global optimization of implicit functions is given.

ALGORITHM 1

(1) *Initialization*
  (a) *Set* $\Sigma = \{P\}$.
  (b) *Set* $k := 0$, $\epsilon_{\mathrm{tol}} > 0$, $\alpha_0 = +\infty$, $\beta_0 = -\infty$.
(2) *Termination*
  (a) *Check if* $\Sigma = \emptyset$. *If true, terminate, the instance is infeasible*
  (b) *Check if* $\alpha_k - \beta_k \le \epsilon_{tol}$. *If true, terminate,* $f^* := \alpha_k$ *is an* ϵ_{tol}*-optimal estimate for the optimal objective function value and* $\mathbf{p}^*$ *is a feasible point at which* f^* *is attained.*
  (c) *Delete from* Σ *all nodes* P^l *with* $f_l^{\mathrm{LBD}} \ge \alpha_k$ *and set* $\beta_k := \min_{P^l\in\Sigma} f_l^{\mathrm{LBD}}$.
(3) *Node Selection*
  (a) *Pop and delete a node* P^l *from stack* Σ *such that* $\beta_k = f_l^{\mathrm{LBD}}$.
(4) *Lower-Bounding Procedure*
  (a) *Solve convex lower-bounding problem* (16) *globally on* P^l.
  (b) *If no feasible solution exists, set* $f_l^{\mathrm{LBD}} := +\infty$, *otherwise set* $f_l^{\mathrm{LBD}} := \mathrm{u}_f(\mathbf{x}^c(\check{\mathbf{p}}), \mathbf{x}^C(\check{\mathbf{p}}), \check{\mathbf{p}})$. *If a feasible solution is found that is feasible in* (4) *and* $f(\mathbf{x}(\check{\mathbf{p}}), \check{\mathbf{p}}) < \alpha_k$, *set* $\alpha_k := f(\mathbf{x}(\check{\mathbf{p}}), \check{\mathbf{p}})$, *and* $\mathbf{p}^* := \check{\mathbf{p}}$.
(5) *Upper-Bounding Procedure* (*optional*)
  (a) *Solve the NLP subproblem* (15) *locally on* P^l.
  (b) *If a feasible solution is found and* $f_l^{\mathrm{UBD}} < \alpha_k$, *set* $\alpha_k := f_l^{\mathrm{UBD}}$, $\mathbf{p}^* := \hat{\mathbf{p}}$.

(6) *Fathoming*
  (a) *Check if* $f_l^{\mathrm{LBD}} = +\infty$ *or* $f_l^{\mathrm{LBD}} \geq \alpha_k$. *If true, go to 2.*
(7) *Branching*
  (a) *Find* $j \in \arg\max_{i=1,\ldots,n_p} w(P_i^l)$ *and create two new nodes* $P^{l'}$ *and* $P^{l''}$ *by bisecting* P_j^l.
  (b) *Set* $f_{l'}^{\mathrm{LBD}}, f_{l''}^{\mathrm{LBD}} := f_l^{\mathrm{LBD}}$ *and push the new nodes onto top of stack* Σ.
  (c) *Set* $k := k + 1$, *go to 2.*

4.4 *Finite convergence*

Guaranteed finite ϵ_{tol}-optimal convergence of Algorithm 1 is established in this section.

DEFINITION 4.1 (X) *Let* $X : \mathbb{I}P \to \mathbb{IR}^{n_x}$ *be a continuous, interval-valued function which is both an interval extension and inclusion function of* $\mathbf{x}$ *on* P *such that for each* $\mathbf{p} \in P$, $\mathbf{x}(\mathbf{p})$ *is the unique solution of* $\mathbf{h}(\mathbf{x}(\mathbf{p}), \mathbf{p}) = \mathbf{0}$ *in* $X(P)$.

It is assumed that such a function X is readily available by some procedure, such as the parametric extension of interval-Newton methods [10,24,34]. In (15), the set X is the initial interval bounds on the implicit function that satisfies the previous assumptions (i.e. Assumption 3.1, 3.9, and 3.14). Under Definition 4.1, X has much more specific properties which are required to guarantee the convergence properties of the relaxations. This stricter definition is not required in order to solve the upper-bounding problem (15). However, the methods for calculating X satisfying the previous assumptions also ensure the stricter interval properties in Definition 4.1. Therefore, X in (15) will be equal to $X(P)$ in Definition 4.1.

Assumption 4.2 For $Z \equiv X(P)$, there exist continuous functions $F : \mathbb{I}Z \times \mathbb{I}P \to \mathbb{IR}$ and $G : \mathbb{I}Z \times \mathbb{I}P \to \mathbb{IR}^{n_g}$ such that F is both an interval extension and an inclusion function of f on $Z \times P$ and G is both an interval extension and an inclusion function of $\mathbf{g}$ on $Z \times P$.

For f and $\mathbf{g}$ factorable and continuous on open sets containing $Z \times P$, F and G are calculable by taking natural interval extensions [20,24].

LEMMA 4.3 *Consider a nested sequence of intervals* $\{P^q\}$ (*i.e.* $P^m \subset P^q$, $\forall m > q$), $P^q \subset P$, $q \in \mathbb{N}$, *such that* $\{P^q\} \to [\bar{\mathbf{p}}, \bar{\mathbf{p}}]$ *for some* $\bar{\mathbf{p}} \in P$. *Let* $\mathbf{x}_q^c, \mathbf{x}_q^C$ *be relaxations of* $\mathbf{x}$ *on* P^q. *Let* $f_q^c(\cdot) = \mathrm{u}_f^q(\mathbf{x}_q^c(\cdot), \mathbf{x}_q^C(\cdot), \cdot)$ *be a convex relaxation of the objective function* f *on* P^q. *Let* $\hat{f}_q^c = \min_{\mathbf{p} \in P^q} f_q^c(\mathbf{p})$. *Then* $\lim_{q\to\infty} \hat{f}_q^c = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$.

Proof From continuity of X on $\mathbb{I}P$, it is clear that $\lim_{q\to\infty} X(P^q) = X([\bar{\mathbf{p}}, \bar{\mathbf{p}}])$ and since X is an interval extension of $\mathbf{x}$, $X([\bar{\mathbf{p}}, \bar{\mathbf{p}}]) = [\mathbf{x}(\bar{\mathbf{p}}), \mathbf{x}(\bar{\mathbf{p}})]$. Let F^q be an interval function satisfying Assumption 4.2 on $\mathbb{I}X(P^q) \times \mathbb{I}P^q$. Then, by continuity of F^q, we have $\lim_{q\to\infty} F^q(X(P^q), P^q) = F([\mathbf{x}(\bar{\mathbf{p}}), \mathbf{x}(\bar{\mathbf{p}})], [\bar{\mathbf{p}}, \bar{\mathbf{p}}]) = [f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}}), f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})] = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. By construction, $\hat{f}_q^c(\mathbf{p}) \in F^q(X(P^q), P^q), \forall \mathbf{p} \in P^q$ for every q, and therefore it follows $\lim_{q\to\infty} \hat{f}_q^c = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. ■

LEMMA 4.4 *Suppose Algorithm 1 generates an infinite sequence of nested nodes* $\{P^q\}$, *then* $\lim_{q\to\infty} P^q = [\bar{\boldsymbol{p}}, \bar{\boldsymbol{p}}]$.

Proof Each node P^q is a subinterval partition of P that is an n_p-dimensional rectangle. The branching rule is a bisection along one of the longest edges of the currently selected node P^q. This result follows analogously from Proposition IV.2 in [15]. ■

LEMMA 4.5 *Suppose Algorithm 1 generates an infinite sequence of nested nodes* $\{P^q\}$, *then* $\{P^q\} \to [\bar{\mathbf{p}}, \bar{\mathbf{p}}]$ *and* $\bar{\mathbf{p}}$ *is feasible in (4).*

Proof By Lemma 4.4, if Algorithm 1 generates an infinite sequence of nested nodes $\{P^q\}$, then $\{P^q\} \to [\bar{\mathbf{p}}, \bar{\mathbf{p}}]$. Suppose $\bar{\mathbf{p}}$ is infeasible in the original problem, i.e. $g_i(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}}) > \mathbf{0}$ for some $i = 1, 2, \ldots, n_g$. Let $\mathbf{g}^c(\cdot) = \mathbf{u}_{\mathbf{g}}(\mathbf{x}^c(\cdot), \mathbf{x}^C(\cdot), \cdot)$. By continuity of $\mathbf{g}$, there exists an open ball, of radius $\delta > 0$, around $\bar{\mathbf{p}}$, labelled $B_\delta(\bar{\mathbf{p}})$, such that $\hat{\mathbf{p}} \in B_\delta(\bar{\mathbf{p}}) \Rightarrow g_i(\mathbf{x}(\hat{\mathbf{p}}), \hat{\mathbf{p}}) > 0$ for some $i = 1, 2, \ldots, n_g$. This implies that for some finite q', $P^{q'} \subset B_\delta(\bar{\mathbf{p}})$. Therefore, there exists a $q'' > q'$ such that for some $i = 1, 2, \ldots, n_g$, we have $g_i^c(\mathbf{p}) > 0, \forall \mathbf{p} \in P^{q''}$, where continuity of g_i^c (and $\mathbf{x}^c, \mathbf{x}^C$) on P follows from the definition of composite relaxations (Definition 2.16) and the properties of generalized McCormick [31] relaxations. Thus, the convex lower-bounding problem (16) is infeasible for all $q > q''$. Finally, the node containing $\bar{\mathbf{p}}$ would be fathomed no later than at node $q'' + 1$. Therefore, Algorithm 1 cannot generate an infinite sequence of nested nodes that converge to an infeasible point. ■

LEMMA 4.6 *Suppose an infinite sequence of nested nodes,* $\{P^q\}$, *is generated by Algorithm 1. Let* $f_q^c(\cdot) = \mathrm{u}_f^q(\mathbf{x}_q^c(\cdot), \mathbf{x}_q^C(\cdot), \cdot)$ *and* $\mathbf{g}_q^c(\cdot) = \mathbf{u}_{\mathbf{g}}^q(\mathbf{x}_q^c(\cdot), \mathbf{x}_q^C(\cdot), \cdot)$ *be convex relaxations of* f *and* $\mathbf{g}$ *on* P^q, *respectively. Let* $f_q^{*,c} = \min_{\mathbf{p} \in P^q} f_q^c(\mathbf{p}) : \mathbf{g}_q^c(\mathbf{p}) \leq \mathbf{0}$. *Then* $\{P^q\} \to [\bar{\mathbf{p}}, \bar{\mathbf{p}}]$ *and* $\lim_{q \to \infty} f_q^{*,c} = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$.

Proof By Lemma 4.5, $\{P^q\} \to [\bar{\mathbf{p}}, \bar{\mathbf{p}}]$, with $\bar{\mathbf{p}} \in P$ feasible. Let $\hat{f}_q^c = \min_{\mathbf{p} \in P^q} f_q^c(\mathbf{p})$. Since $\hat{f}_q^c$ is the solution of the convex unconstrained problem, it is clear that $\hat{f}_q^c \leq f_q^{*,c}$. Since $f_q^{*,c}$ is a rigorous lower bound of $f(\mathbf{x}(\cdot), \cdot)$ on P^q, we have $\hat{f}_q^c \leq f_q^{*,c} \leq f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. Since $\lim_{q \to \infty} \hat{f}_q^c = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$ from Lemma 4.3, it is clear that $\lim_{q \to \infty} f_q^{*,c} = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. ■

LEMMA 4.7 *Let* f^* *denote the globally optimal objective function value for (4). The sequence of lower bounds generated by Algorithm 1 is either finite or satisfies* $\lim_{k \to \infty} \beta_k = f^*$.

Proof This result follows from Theorem 2.1 in [12] where the hypotheses are guaranteed by Lemmas 4.4–4.6 above. ■

LEMMA 4.8 *Suppose that an infinite sequence of nested nodes,* $\{P^q\}$, *is generated by Algorithm 1. Also, suppose that the upper-bounding problem* (15) *can locate a feasible point for every* $q \geq q'$ *for some finite* q', *and thus a valid upper bound can be located in every subsequent node. Then, the upper-bounding operation converges to the global solution of* (4), *i.e.* $\lim_{k \to \infty} \alpha_k = f^*$.

Proof From Lemma 4.5, if Algorithm 1 generates an infinite sequence of nested nodes, $\{P^q\}$, then $\{P^q\} \to [\bar{\mathbf{p}}, \bar{\mathbf{p}}]$ and $\bar{\mathbf{p}}$ is feasible. From Lemma 4.6, we know that $\lim_{q \to \infty} f_q^{*,c}(\mathbf{p}) = f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. Suppose that $\bar{\mathbf{p}}$ is not a global minimizer. Then $f^* < f(\mathbf{x}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$ implying that for some q'' we have $f_{q''}^{*,c} > f^*$. However, using the bound-improving node selection property of Algorithm 1, this node would have never been selected again for branching. Therefore $\bar{\mathbf{p}}$ must be a global minimizer $\mathbf{p}^* = \bar{\mathbf{p}}$.

From continuity of f, for some $\epsilon > 0$, there exists an open ball of radius $\delta > 0$ around $\mathbf{p}^*$, $B_\delta(\mathbf{p}^*)$, such that $\mathbf{p} \in B_\delta(\mathbf{p}^*) \Rightarrow |f(\mathbf{x}(\mathbf{p}), \mathbf{p}) - f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*)| < \epsilon$, where continuity of $\mathbf{x}$ on P follows from continuous differentiability of $\mathbf{h}$ and the implicit function theorem.

By hypothesis, after some finite q', a feasible point $\hat{\mathbf{p}} \in P^q$ can be found that provides a valid upper bound f_q^{UBD}. By the bound-improving property, if f_q^{UBD} is lower than the current upper bound α_k, then $\alpha_k := f_q^{\mathrm{UBD}}$. For q large enough, a feasible point $\mathbf{p}$ will be located such that $\mathbf{p} \in B_\delta(\mathbf{p}^*)$. By continuity of f, we have $|f(\mathbf{x(p)},\mathbf{p}) - f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*)| < \epsilon \Rightarrow |f_q^{\mathrm{UBD}} - f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*)| < \epsilon \Rightarrow f_q^{\mathrm{UBD}} <$

$f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*) + \epsilon$. Since $f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*) \leq \alpha_k \leq f_q^{\text{UBD}}$, we have $f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*) \leq \alpha_k < f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*) + \epsilon$. Thus $\lim_{k\to\infty} \alpha_k = f(\mathbf{x}(\mathbf{p}^*), \mathbf{p}^*) = f^*$. ■

THEOREM 4.9 (Finite Convergence) *Let X be as defined in Definition 4.1 and suppose Assumption 4.2 holds. Also, suppose the hypotheses of Lemma 4.8 are satisfied. Then, after finitely many iterations, Algorithm 1 terminates with either ϵ-optimal global solutions, such that $\alpha_k - \beta_k \leq \epsilon_{\text{tol}}$, or a guarantee that the problem is infeasible.*

Proof Follows immediately from Lemmas 4.7 and 4.8 and the deletion by infeasibility rule. ■

5. Illustrative examples

Example 5.1 For the purposes of illustrating a problem having multiple implicit function branches, consider the problem outlined in Example 3.26 with the objective function $f : Z \times P \to \mathbb{R}$ defined as

$$f(z, p) = z.$$

The absolute and relative convergence tolerances were set to 10^{-3} and Algorithm 1 was called in a sequential fashion with the domain X^1 first, followed by X^2. The solution $p^* = 9$ was found with a value of $f^* = -8.53113$. The problem was solved by Algorithm 1 taking five iterations on X^1 and nine iterations on X^2 with a total time of 8.7×10^{-3}s. The lower-bounding problems were solved using PBUN, a nonsmooth optimization algorithm developed in [17].

If a lower-bounding problem returned a feasible point $\mathbf{p}$, the model equations were solved at this point using Newton's method with Gauss–Seidel and the objective function was evaluated for an upper bound on the solution, instead of solving (15) locally.

For comparison, this problem was modelled in GAMS version 23.9 [28] using the BARON solver [35]. For a fairer comparison, the local search procedure for obtaining an upper bound was turned off. Similarly, since no preprocessing steps were being employed with Algorithm 1, the GAMS preprocessor was turned off. BARON solved the problem after two iterations with guaranteed optimality after 0.04 s.

Example 5.2 Let $Z \in \mathbb{IR}^3$ and $P \in \mathbb{IR}^3$. Consider the objective function $f : Z \times P \to \mathbb{R}$ defined as

$$f(\mathbf{z}, \mathbf{p}) = \sum_{j=1}^{3} \left([a_j(p_j - c_j)]^2 + \sum_{i \neq j} a_i(p_i - c_i) - 5\left((j-1)(j-2)(z_2 - z_1) + \sum_{i=1}^{3} (-1)^{i+1} z_i\right)\right)^2 \tag{17}$$

with a_i, c_i being constants for $i = 1, 2, 3$, given in Table 1.

Consider the equality constraints

$$\mathbf{h}(\mathbf{z}, \mathbf{p}) = \begin{pmatrix} 1.00 \times 10^{-9}(\exp[38z_1] - 1) + p_1 z_1 - 1.6722 z_2 + 0.6689 z_3 - 8.0267 \\ 1.98 \times 10^{-9}(\exp[38z_2] - 1) + 0.6622 z_1 + p_2 z_2 + 0.6622 z_3 + 4.0535 \\ 1.00 \times 10^{-9}(\exp[38z_3] - 1) + z_1 - z_2 + p_3 z_3 - 6.0 \end{pmatrix} = \mathbf{0}. \tag{18}$$

Table 1. Constants for the objective function of Example 5.2.

Example 1 Constants	
a_1	37.3692
c_1	0.602
a_2	18.5805
c_2	1.211
a_3	6.25
c_3	3.60

The full-space optimization formulation is

$$\begin{aligned} \min_{(\mathbf{z},\mathbf{p})\in Z\times P} \quad & f(\mathbf{z},\mathbf{p}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{z},\mathbf{p}) = \mathbf{0} \\ & Z = [-5,5]^3 \\ & P = [0.6020, 0.7358] \times [1.2110, 1.4801] \times [3.6, 4.4]. \end{aligned} \tag{19}$$

The reduced-space, box-constrained, formulation becomes

$$\begin{aligned} \min_{\mathbf{p}\in P} \quad & f(\mathbf{x}(\mathbf{p}),\mathbf{p}) \\ & P = [0.6020, 0.7358] \times [1.2110, 1.4801] \times [3.6, 4.4] \end{aligned} \tag{20}$$

Using the parametric interval-Newton method with interval Gauss–Seidel, an interval, X, that conservatively bounds the implicit function $\mathbf{x}$ on all of P, can be calculated:

$$X = [0.5180, 0.5847] \times [-3.9748, -3.0464] \times [0.3296, 0.5827].$$

It is apparent that X is significantly tighter than Z. This problem has a suboptimal local minimum at $\mathbf{p} = (0.602, 1.46851, 3.6563)$ with a value of 731.197 and a global minimum at $\mathbf{p}^* = (0.703918, 1.43648, 3.61133)$ with a value of 626.565. This problem was solved in 0.4 s with Algorithm 1 taking 43 iterations with tolerances for convergence as 10^{-3} for relative error and absolute error. The convex lower-bounding problems were again solved using PBUN.

The upper bound was obtained as in the previous example. Plots of the implicit objective function $f(\mathbf{x}(\mathbf{p}),\mathbf{p})$ are shown below in Figure 2 for four different values of p_3. Similarly, the implicit objective function and corresponding relaxations are shown in Figure 3 for the same four values of p_3.

For comparison, this problem was modelled in GAMS version 23.9 [28] using the BARON solver [35]. Starting with the variable interval Z, BARON failed to solve the problem noting 'No feasible solution was found'. Using the interval X calculated above, BARON solved the problem and returned the global solution in 1 s after 810 iterations. For completeness, the preprocessor was turned on. Without solving NLPs, BARON performs no differently than without the preprocessor. Allowing the preprocessor to solve NLPs, the solution is found after solving two NLPs and BARON terminates with guaranteed optimality in 0.5 s.

Example 5.3 Consider the parameter estimation example presented in [19] which was adapted from [32,33]. This problem attempts to determine whether or not a proposed kinetic mechanism

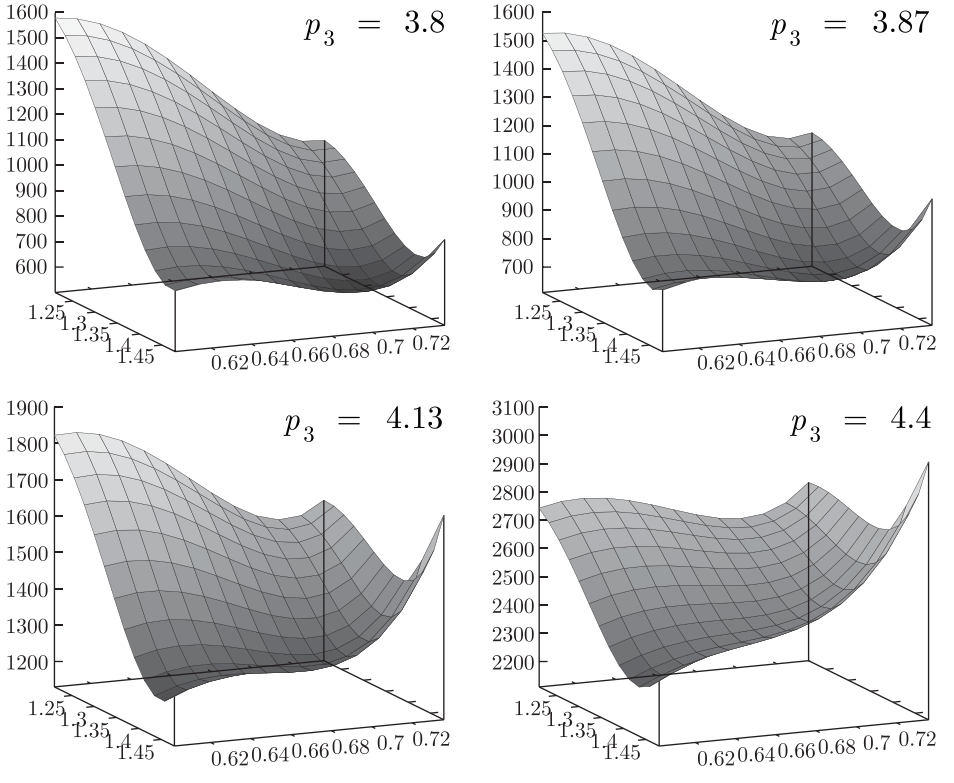

Figure 2. The objective function of Example 5.2 on $P_1 \times P_2$ at three different p_3 values.

sufficiently predicts the behaviour of a reacting system for which experimental data are available. The following kinetic mechanism is proposed:

$$Z + Y \xrightarrow{k_1} A, \quad A + O_2 \underset{k_{2f/K_2}}{\overset{k_{2f}}{\longleftrightarrow}} D,$$

$$A + O_2 \underset{k_{3f/K_2}}{\overset{k_{3f}}{\longleftrightarrow}} B, \quad B \xrightarrow{k_4} M + N,$$

$$2A \xrightarrow{k_5} P,$$

which is modelled as a system of nonlinear ordinary differential equations (ODEs):

$$\frac{\mathrm{d}c_A}{\mathrm{d}t} = k_1 c_Z c_Y - c_{O_2}(k_{2f} + k_{3f})c_A + \frac{k_{2f}}{K_2}c_D + \frac{k_{3f}}{K_3}c_B - k_5 c_A^2,$$

$$\frac{\mathrm{d}c_B}{\mathrm{d}t} = k_{3f} c_{O_2} c_A - \left(\frac{k_{3f}}{K_3} + k_4\right) c_B, \quad \frac{\mathrm{d}c_D}{\mathrm{d}t} = k_{2f} c_A c_{O_2} - \frac{k_{2f}}{K_2} c_D,$$

$$\frac{\mathrm{d}c_Y}{\mathrm{d}t} = -k_{1s} c_Y c_Z, \quad \frac{\mathrm{d}c_Z}{\mathrm{d}t} = -k_1 c_Y c_Z,$$

$$c_A(t=0) = 0, \quad c_B(t=0) = 0, \quad c_D(t=0) = 0, \quad c_Y(t=0) = 0.4, \quad c_Z(t=0) = 140, \tag{21}$$

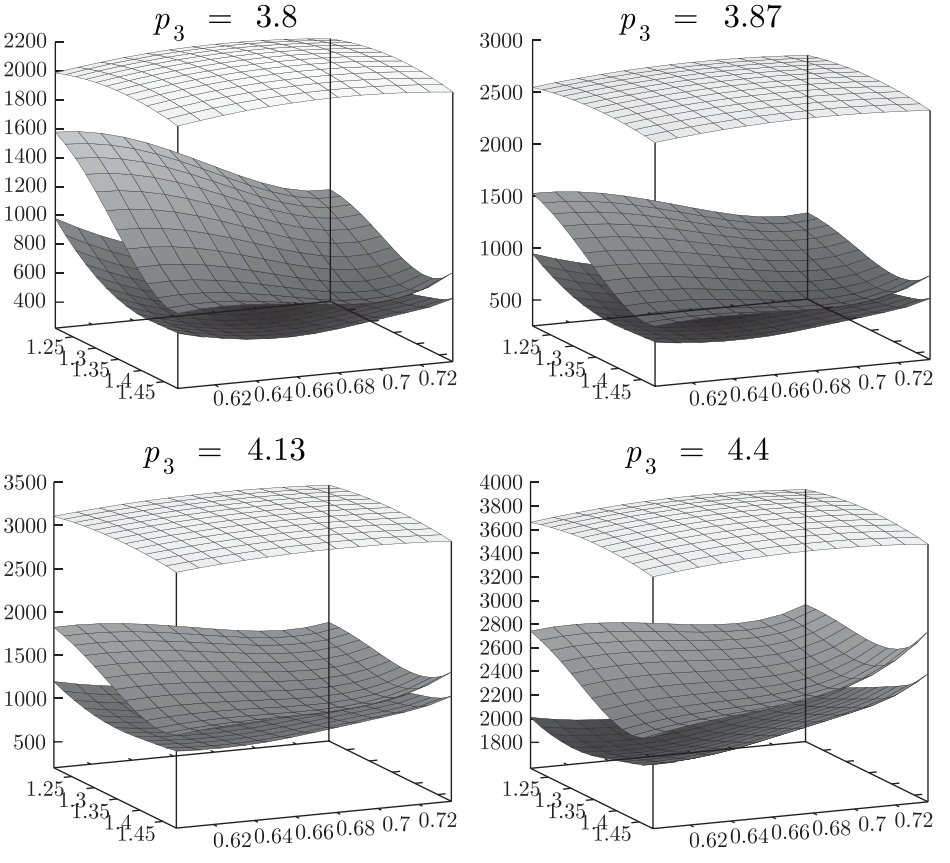

Figure 3. The objective function of Example 5.2 on $P_1 \times P_2$ at three different p_3 values and corresponding convex and concave relaxations.

where c_j is the concentration (in appropriate units) of species j, $T = 273$, $K_2 = 46 \exp[6500/T - 18]$, $K_3 = 2K_2$, $k_1 = 53$, $k_{1s} = k_1 \times 10^{-6}$, $k_5 = 1.2 \times 10^{-3}$, and $c_{O_2} = 2 \times 10^{-3}$. The uncertain model parameters are $\mathbf{p} = (k_{2f}, k_{3f}, k_4)$ with $k_{2f} \in [10, 1200]$, $k_{3f} \in [10, 1200]$, and $k_4 \in [0.001, 40]$. Each experimental measurement is given in the form of $I_d = c_A + \frac{2}{21}c_B + \frac{2}{21}c_D$ which comes from the Beer–Lambert law for relating measured absorbance to concentration with a correction for multiple species [32]. The same data used in [19] is used here and can be downloaded from http://yoric.mit.edu/libMC/libmckinexdata.txt.

Using the implicit-Euler discretization scheme, the time domain is discretized into $n = 200$ evenly spaced nodes and the solution of the ODE system (21) can be approximated, with reasonable accuracy, as the solution of a corresponding nonlinear algebraic system with $5n$ state variables and 3 parameters. The method of Mitsos et al. [19] was not applicable to this implicit scheme and so in [19] the method was demonstrated using the explicit-Euler discretization scheme. As an aside, approximating the solution of an ODE system using the explicit-Euler numerical integration method may suffer from numerical instabilities when the problem is *stiff* (i.e. when the solution exhibits fast transient behaviour) whereas the implicit technique, albeit more computationally expensive per time step, is unconditionally stable and can therefore handle much larger time steps than the explicit approach. However, it should be noted that unconditional stability does not imply that the solution is reasonably accurate for large time steps. The ODE (21) is considered to be moderately stiff and so either approach may work well. For $i = 1, \ldots, n$, the resulting nonlinear

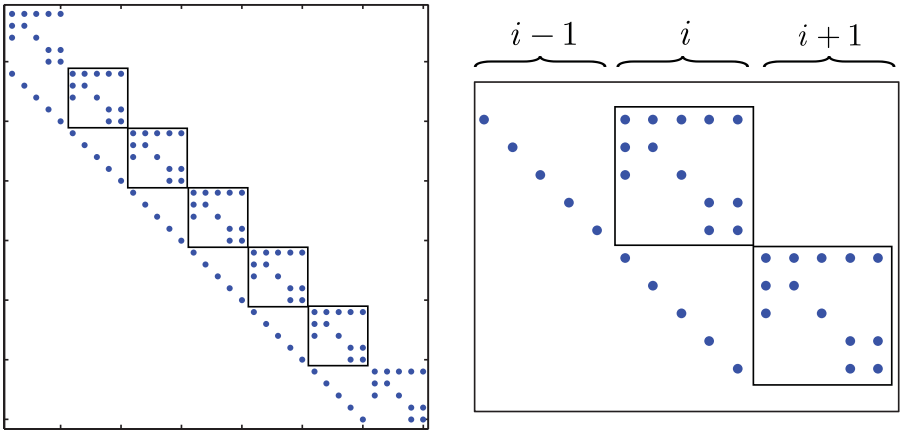

Figure 4. (Left) The sparsity pattern of the system with $n = 7$ discretization. Each 5×5 block is highlighted to show how the system can be solved in a sequential block-by-block fashion. (Right) An expanded view of three time steps showing how information from the previous node is used to solve the 5×5 system associated with the current node.

algebraic system is

$$\begin{aligned}
0 &= c_A^{i-1} - c_A^i + \Delta t \left(k_1 c_Y^i c_Z^i - c_{O_2}(k_{2f} + k_{3f}) c_A^i + \frac{k_{2f}}{K_2} c_D^i + \frac{k_{3f}}{K_3} c_B^i - k_5 {c_A^i}^2 \right), \\
0 &= c_B^{i-1} - c_B^i + \Delta t \left(k_{3f} c_{O_2} c_A^i - \left(\frac{k_{3f}}{K_3} + k_4 \right) c_B^i \right), \\
0 &= c_D^{i-1} - c_D^i + \Delta t \left(k_{2f} c_A^i c_{O_2} - \frac{k_{2f}}{K_2} c_D^i \right), \\
0 &= c_Y^{i-1} - c_Y^i + \Delta t(-k_{1s} c_Y^i c_Z^i), \\
0 &= c_Z^{i-1} - c_Z^i + \Delta t(-k_1 c_Y^i c_Z^i),
\end{aligned} \tag{22}$$

where for $n = 200$, $\Delta t = 0.01$. The resulting explicit NLP formulation therefore has $5n + 3$ variables with

$$\mathbf{z} = (c_A^1, c_B^1, c_D^1, c_Y^1, c_Z^1, \ldots, \ldots, \ldots, \ldots, \ldots, c_A^{200}, c_B^{200}, c_D^{200}, c_Y^{200}, c_Z^{200}).$$

By solving the system for the state variables as implicit functions of the parameters, the resulting implicit NLP formulation has just 3 independent variables. This can be done using two different techniques. The first, which is not recommended, is to treat the nonlinear system of equations as fully coupled and essentially solve for the state variables simultaneously. Thus, in order to construct relaxations of implicit functions, using this technique would require relaxing 1000 implicit functions simultaneously. The second technique, which is how numerical integration is typically performed, exploits the block structure of the problem.

Taking a look at the sparsity pattern of the system, a portion of which is shown in Figure 4, it is easy to notice that each equation at node i is only dependent on the variables at node i and the variables at node $i - 1$. Therefore, if the variables at node $i - 1$ are known, node i can be solved as a system of five nonlinear equations. Since node 0 is specified by the initial conditions, this technique can be applied sequentially from node 1 to node 200. Again, this is how the implicit-Euler numerical integration method is applied. Constructing relaxations is then done in an analogous fashion. Relaxations are constructed for each system of 5 equations using the method of Section 3.4 and subsequently used in the construction of relaxations of each system associated

Table 2. Suboptimal local minima of Example 5.3 (using the sequential block solve technique) found using the multi-start SQP approach.

k_{2f}	k_{3f}	k_4	$f^* \times 10^{-4}$
235.04	1048.8	0.33151	1.7066726
350.72	931.25	0.38279	1.7056881
678.53	596.96	0.82748	1.7024373
765.26	450.21	12.414	1.6807190
355.02	926.55	11.766	1.7056560
740.18	523.81	13.717	1.6993238
735.88	528.60	13.993	1.6995289
627.16	552.87	12.187	1.7051711
775.44	437.23	17.576	1.6802801

with the next node with the relaxations of node 0 taken to be exactly the initial conditions for all $\mathbf{p} \in P$ (since they are constant on P). The initial intervals are taken as $c_j^i \in [0, 140], j \neq Y, \forall i$ and $c_Y^i \in [0, 0.4], \forall i$. This approach is recommended over the simultaneous approach as it is not only significantly less computationally expensive, but it also produces much tighter relaxations.

The objective function for this problem is stated as

$$f(\mathbf{z}, \mathbf{p}) = \sum_{i=1}^{n} (I^i - I_d^i)^2$$

where $I^i = c_A^i + \frac{2}{21} c_B^i + \frac{2}{21} c_D^i, i = 1, \ldots, n$, with $c_A^i, c_B^i, c_D^i, i = 1, \ldots, n$, given by the solution of the nonlinear system (22) and $I_d^i, i = 1, \ldots, n$, are the experimental data mentioned previously.

In an effort to survey the topological features of the objective function for this problem, multi-start optimization techniques were employed. The full-space NLP formulation (i.e. with 1003 variables and 1000 equality constraints) was solved by multi-starting the MINOS solver [22] in GAMS version 23.9 [28]. Only one optimum was found and it happened to correspond with the global solution. The SNOPT solver [7] was also used and a single suboptimal feasible solution was identified along with the solution found by the MINOS solver. Alternatively, the implicit NLP formulation, where the implicit functions are evaluated using the second technique described above (i.e. sequential block solution), was then solved by multi-starting the MATLAB SQP solver. In this case, eight suboptimal local minima were found along with the global minimum. The suboptimal local minima that were found are reported in Table 2. This is a rather interesting result because it means that, for this problem, the reduced-space formulation has many suboptimal local minima, whereas the full-space formulation may only have a few. This is consistent with what was found in the methanol-to-hydrocarbons example in [5] but is not a result that holds in general as is demonstrated by [5] with the Lotka–Volterra example.

The reduced-space NLP was solved using Algorithm 1, without a local-search upper-bounding procedure, taking 4700 s (1.3 h) and 1133 iterations with convergence tolerances of 10^{-2} and 10^{-5} for absolute and relative error, respectively. The optimal parameter values were found, $\mathbf{p}^* = (798.019, 423.845, 12.9685)$ with $f^* = 16,796.04$, and the 'best fit' corresponding to the optimal solution $\mathbf{p}^*$ was plotted against the experimental data in Figure 5. It should be noticed that the width of the parameter interval for k_4 is quite a lot smaller than the width of the other two parameter intervals. For such cases, it is recommended that branching on the parameter space occurs according to relative width as opposed to absolute width. Therefore, the relative-width branching heuristic was employed here.

As was concluded in [19], the model with the best fit parameters does not agree with experimental data at early times. Since a certificate of global optimality was obtained, one can conclude

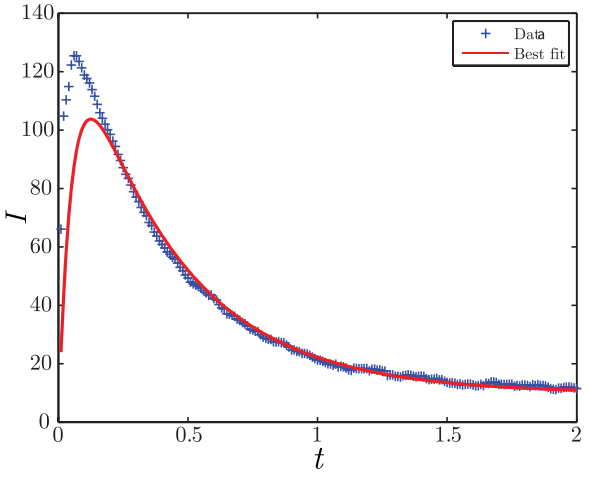

Figure 5. The optimal 'best fit' of the model plotted against the experimental data.

that the model cannot represent the physical system at early times. The high cost-per-node is due to the extremely expensive function evaluations for this problem. Even though there are only three independent variables, evaluating the objective function requires evaluating 1000 implicit functions and constructing relaxations requires interval bounds and relaxations on 1000 implicit functions.

For comparison, the full-space NLP was modelled in GAMS 23.9 [28] and solved using BARON [35]. Both a selective-branching strategy and the standard strategy of branching on all variables were studied. Using MINOS as the local-search algorithm for solving the upper-bounding problem, BARON converged to the solution found using the multi-start approach discussed above within just a few seconds, for each branching strategy. However, using SNOPT [7] as the local-search algorithm for solving the upper-bounding problem, BARON converged to a suboptimal solution in just a few seconds for each branching strategy. It should be noted that in each case, BARON terminates normally claiming that it found a solution with a guarantee of global optimality. The behaviour of BARON here is not fully understood and so it is considered to be ineffective at solving this problem. Alternatively, each strategy was tried without using local-search algorithms for the upper-bounding problems and without using preprocessing. When considering the strategy of branching on all of the variables, BARON fails to solve the problem. For this case, the algorithm terminates after about 460 s with the result that no feasible solution could be found. Again, this is a very strange result since the problem is indeed feasible. Figure 6 is a plot showing the performance, in terms of the ratio of the lower and upper bounds versus CPU time in seconds, of Algorithm 1 versus BARON with selective branching and without a local-solve upper-bounding procedure. After about 30 s, Algorithm 1 improves on the bounds quite effectively, even without a local-search upper-bounding procedure. It takes BARON about 50,000 s to achieve the same level of convergence as Algorithm 1 achieves after the 30 s mark. After about 100 s, Algorithm 1 begins to exhibit slower but consistent improvement on the bounds until it converges. When Algorithm 1 converges to the global solution (after 4700 s), BARON is about 75% converged. The BARON selective branching strategy fails to converge even after more than 70 h when the maximum number of iterations of 100,000 is reached. At this time BARON is only 97% converged. It is clear that for this problem, Algorithm 1 performs far more favourably than BARON.

Again, for completeness, the experiments were run with preprocessing switched on. There was no change in performance with preprocessing switched on without solving NLPs. Allowing NLPs to be solved in the preprocessing step yields results similar to those discussed previously regarding

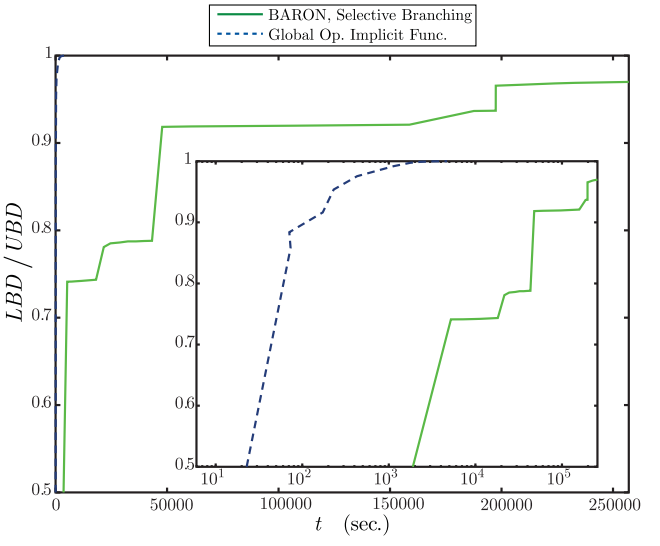

Figure 6. The performance of two methods on the kinetic mechanism example in terms of convergence.

the multi-start approach. Again, with MINOS as the NLP solver, the problem is solved in just a few seconds. Using SNOPT, the problem is solved in about 3 min.

6. Conclusion

A reformulation of the standard NLP with equality constraints has been proposed that is an equivalent formulation offering a potentially large reduction in dimensionality. By solving the equality constraints for dependent variables as implicit functions of independent variables, they are eliminated from the program and the implicit functions are embedded within the objective function and inequality constraint(s). If the original problem had only equality constraints, the reduced-space reformulation is simply an inequality-constrained problem. In order to solve the reduced-space problem, new results for relaxing implicit functions were developed.

One new result was presented that guarantees that relaxations of a successive-substitution iteration are also valid relaxations of the implicit function. Another key result pertaining to solutions of parametric linear systems was presented. This result states that relaxations of the solution of a parametric linear system can be calculated iteratively in a fashion analogous to the Gauss–Seidel method. It was demonstrated that relaxations of the generic Newton-type iteration cannot be refinements of the original bounds on the implicit function. This proves that direct relaxations of Newton-type iterations are not useful, but relaxations of convergent successive-substitution iterations may be useful. Because of this, new methods, analogous to interval Newton-type methods, were developed that essentially relax the implicit functions by relaxing the mean-value theorem. These novel developments offer ways to calculate relaxations of an implicit function that is a parametric solution of a general nonlinear system of equations that cannot be approximated via a successive-substitution iteration. Furthermore, subgradients of such relaxations can be calculated, which are useful in the solution of the resulting nonsmooth convex program.

Utilizing these new results, a reduced-space global optimization algorithm has been proposed for solving nonconvex NLPs with embedded implicit functions. The algorithm was shown to converge in finitely many iterations to an ϵ-optimal solution. The algorithm was applied to three numerical examples which demonstrate a proof-of-concept.

Acknowledgements

The authors would like to acknowledge the Chevron University Partnership Program for supporting this work through the MIT Energy Initiative (MITEI/UPP). The authors would also like to give a special acknowledgement to Harry Watson for his contribution to the kinetic model example (Example 5.3).

References

[1] C.S. Adjiman and C.A. Floudas, *Rigorous convex underestimators for general twice-differentiable problems*, J. Global Optim. 9 (1996), pp. 23–40.
[2] C. Bendtsen and O. Stauning, *FADBAD, a flexible C++ package for automatic differentiation*, Tech. Rep. 1996-x5-94, Technical University of Denmark, Lyngby, Denmark, 1996.
[3] B. Chachuat, *MC++: A versatile library for McCormick relaxations and Taylor models*. Available at http://www3.imperial.ac.uk/people/b.chachuat/research.
[4] T.G. Epperly and E.N. Pistikopoulos, *A reduced space branch and bound algorithm for global optimization*, J. Global Optim. 11 (1997), pp. 287–311.
[5] W.R. Esposito and C.A. Floudas, *Global optimization for the parameter estimation of differential-algebraic systems*, Ind. Eng. Chem. Res. 39 (2000), pp. 1291–1310.
[6] J.E. Falk and R.M. Soland, *An algorithm for separable nonconvex programming problems*, Manag. Sci. 15 (1969), pp. 550–569.
[7] P.E. Gill, W. Murray, and M.A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev. 47 (2005), pp. 99–131.
[8] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Automatic Differentiation*, 2nd ed., SIAM, Philadelphia, PA, 2008.
[9] E.R. Hansen and R.I. Greenberg, *Interval Newton methods*, Appl. Math. Comput. 12 (1983), pp. 89–98.
[10] E. Hansen and G.W. Walster, *Global Optimization Using Interval Analysis*, 2nd ed., Marcel Dekker, New York, 2004.
[11] R. Horst, *A general class of branch-and-bound methods in global optimization with some new approaches for concave minimization*, J. Optim. Theory Appl. 51 (1986), pp. 271–291.
[12] R. Horst, *Deterministic global optimization with partition sets whose feasibility is not known: Application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization*, J. Optim. Theory Appl. 58 (1988), pp. 11–37.
[13] R. Horst and N.V. Thoai, *Conical algorithm for the global minimization of linearly constrained decomposable concave minimization problems*, J. Optim. Theory Appl. 74 (1992), pp. 469–486.
[14] R. Horst and N.V. Thoai, *Constraint decomposition algorithms in global optimization*, J. Global Optim. 5 (1994), pp. 333–348.
[15] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed., Springer, Berlin, 1996.
[16] R.B. Kearfott, *Preconditioners for the interval Gauss-Seidel method*, SIAM J. Numer. Anal. 27 (1990), pp. 804–822.
[17] L. Luksan and J. Vlcek, *Algorithms for non-differentiable optimization*, ACM Trans. Math. Soft. 27 (2001), pp. 193–213.
[18] G.P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems*, Math. Program 10 (1976), pp. 147–175.
[19] A. Mitsos, B. Chachuat, and P.I. Barton, *McCormick-based relaxations of algorithms*, SIAM J. Optim. 20 (2009), pp. 573–601.
[20] R.E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, PA, 1979.
[21] J.R. Munkres, *Analysis on Manifolds*, Westview Press, Boulder, CO, 1991.
[22] B.A. Murtagh and M.A. Saunders, *MINOS 5.5 user's guide*, Tech. Rep., Stanford University, Stanford, CA, 1983.
[23] L.D. Muu and W. Oettli, *Combined branch-and-bound and cutting plane methods for solving a class of nonlinear programming problems*, J. Global Optim. 3 (1993), pp. 377–391.
[24] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.
[25] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, Inc., Boston, MA, 1970.
[26] A.T. Phillips and J.B. Rosen, *A parallel algorithm for constrained concave quadratic global optimization*, Math. Program. 42 (1988), pp. 421–448.
[27] E.D. Popova, *On the solution of parametrised linear systems*, in *Scientific Computing, Validated Numerics, Interval Methods*, K. Walter von Gudenberg and J. Wolff, eds., Kluwer Academic Publishers, Boston, MA, 2001, pp. 127–138.
[28] R.E. Rosenthal, *GAMS—A User's Manual*, Washington, DC, 2012.
[29] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed., McGraw-Hill, New York, 1976.
[30] H.S. Ryoo and N.V. Sahinidis, *A branch-and-reduce approach to global optimization*, J. Global Optim. 8 (1996), pp. 107–138.
[31] J.K. Scott, M.D. Stuber, and P.I. Barton, *Generalized McCormick relaxations*, J. Global Optim. 51 (2011), pp. 569–606.
[32] A.B. Singer, *Global dynamic optimization*, Ph.D. thesis, Massachusetts Institute of Technology, 2004.
[33] A.B. Singer, J.W. Taylor, P.I. Barton, and W.H. Green, *Global dynamic optimization for parameter estimation in chemical kinetics*, J. Phys. Chem. A 110 (2006), pp. 971–976.

[34] M.D. Stuber, *Evaluation of process systems operating envelopes*, Ph.D. thesis, Massachusetts Institute of Technology, 2013.
[35] M. Tawarmalani and N.V. Sahinidis, *A polyhedral branch-and-cut approach to global optimization*, Math. Program. 103(2) (2005), pp. 225–249.

Appendix 1. Parameterized generalized bisection

Calculating an interval X such that existence and uniqueness of an enclosed implicit function $\mathbf{x(p)}, \forall \mathbf{p} \in P$ is guaranteed, a prerequisite for applying Algorithm 1, is still an open research topic. One such method, discussed in length in [34] is the *parameterized generalized bisection* procedure. This implementation of the generalized bisection procedure accounts for the parameter dimension by eliminating the situation referred to as a *partial enclosure*, whereby $\mathbf{x(p)} \in X$ only for some values of $\mathbf{p} \in P$. Such a partial enclosure situation is likely if the X dimension were to be blindly bisected as in the standard generalized bisection procedure. Furthermore, there may not exist a cut in the X dimension such that the partial enclosure situation can be avoided without also considering partitioning P. Such is the case when multiple solution branches are located near one another. Therefore, both the X and P dimensions must be systematically partitioned. Figure A1 illustrates this idea.

The overall objective then is to produce intervals $X^i \times P^j, i = 1, \ldots, n_j, j = 1, \ldots m$ such that the P^j are as large as possible. Furthermore, $\cup_j P^j = P$ and $\cup_j \cup_{i=1,\ldots,n_j} X^i \times P^j$ covers the entire solution set on P with a unique continuous solution branch in each $X^i \times P^j$. The algorithmic framework is given below.

ALGORITHM 2

(1) *Initialization*
  (a) *Pick initial box* (X^0, P^0), *initialize solution set* Ξ *and stack* $\mathscr{S} = \{(X^0, P^0)\}$. *Set iteration count* $l := 0$.
(2) *Termination*
  (a) *Stack empty?* ($\mathscr{S} = \emptyset$?)
    (i) *Yes. Algorithm terminates.*
    (ii) *No. Pop and delete a box* (Z, P) *from stack* $\mathscr{S}$, *set* $l := l + 1$.
  (b) *If* $\mathbf{0} \in H(Z, P)$, *go to 4. Else, go to 2* ((Z, P) *has been fathomed*).
(3) *Interval Refinement*
  (a) *Apply parametric interval-Newton-type iteration with extended division starting at* $Z^0 := Z$.
    (i) *If any iteration k yields two disjoint intervals, labelled* Z^L *and* Z^R, *by extended division, place* (Z^L, P) *and* (Z^R, P) *on* $\mathscr{S}$. *Go to 2.*
    (ii) *At every iteration k, apply the standard interval inclusion test.*
      (a). *If the interval-Newton-type operator yields an empty interval, go to 2,* (Z, P) *has been fathomed.*
    (iii) *If the iteration converges and the inclusion test has never been passed, go to 4. Else, place* (Z^k, P) *in solution set* Ξ, *go to 2.*
(4) *Improved Existence and Uniqueness Test*
  (a) *Apply a sharper existence and uniqueness test.*
    (i) *If test is passed, place* (Z^k, P) *on solution set* Ξ, *go to 2. Else, continue.*
    (ii) *If test is failed but partial enclosure possibility is excluded, go to 5. Else, go to 6.*
(5) *Partition X Direction*
  (a) *Partition Z using some strategy to avoid creating partial enclosures and add the resulting boxes to the stack* $\mathscr{S}$.

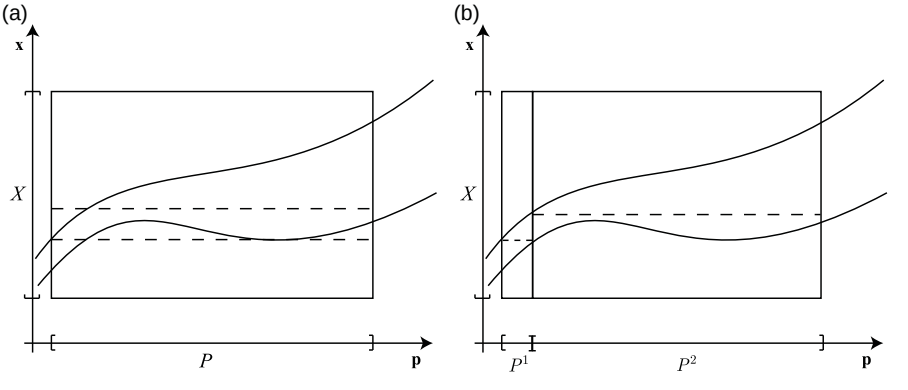

Figure A1. (a) A box $X \times P$ in which there does not exist a position to cut X (dashed lines) such that no partial enclosures are produced. (b) After partitioning P, there exist positions to cut X avoiding partial enclosures.

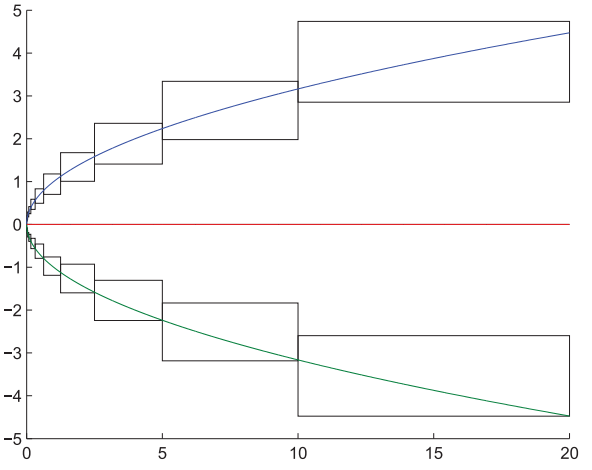

Figure A2. The three solution branches of Example A.1 and the interval boxes computed by Algorithm 2. Note that the middle solution branch has an interval box enclosing it but the box is exact to within machine precision.

(6) *Partition P Direction*
(a) *Partition P using some strategy. Add resulting boxes to the stack* $\mathscr{S}$.

The specific methods for partitioning the X direction and the P direction are discussed in [34] but they are largely heuristic and open for modification and tuning by the user. To demonstrate the application of Algorithm 2, consider the following numerical example.

Example A.1 Consider

$$h(z,p) = -z^3 + pz = 0,$$

with $P^0 = [0, 20]$ and $X^0 = [-10, 10]$. This system has three continuous solution branches that can be defined analytically: $x(p) = 0, \pm\sqrt{p}$. The result of applying Algorithm 2 using the parametric interval-Newton method is shown in Figure A2. The solution branch $x(p) = 0$ has an interval box enclosing it but it is exact within machine precision and therefore does not appear on the plot. Furthermore, the bifurcation point at $p = 0$ is enclosed by a box having a width equal to a user-selected minimum. The box enclosing the bifurcation point is flagged 'unknown' by the algorithm after the stack $\mathscr{S}$ is emptied. This means that further analysis is required by the user as the algorithm cannot process it further. As can be seen in Figure A2, Algorithm 2 effectively partitions and refines the initial interval $X^0 \times P^0$ into subintervals that each enclose locally unique continuous solution branches.