Developing A Constrained Search Approach For Solving Of Nonlinear Equations

Pasukat Sembiring
Department of Mathematics, The University of Sumatera Utara
Email: pasukat@usu.ac.id

***Abstract*—This research Present a new constrained optimization approach for solving system of non-liner equations. Particular advantages are realized when all of the equations are convex. For Example, a global algorithm for finding the zero of a convex real-valued function of one variable is developed. If the algorithm terminates finitely. Then either the algorithm has computed a zero or determined that none exists: if an infinite sequence is generated, either that sequence converges to a zero or again no zero exists. For Solving n-dimensional convex equations, the constrained optimization algorithm has the capability of determining that the system of equations has on solution. Global convergence of the algorithm is established under weaker conditions than previously known. It is also shown once novelty has led to a new algorithm for solving the linear complementarily Problem.**

***Index Terms*—constrained optimization, algorithm, once novelty algorithm.**

I. INTRODUCTION

Due to its importance in solving problems, much effort has been devoted to developing algorithms for finding a zero of an (nxn) system of nonlinear equations (see Rheinboldt [1] and Allgower and Georg [2]). In this work, a new constrained optimization approach is proposed. The corresponding algorithm can sometimes solve problems that order methods cannot-for instance, several examples are provided where the proposed algorithm succeeds and Newton's method fails. However, the primary advantages of the proposed algorithm are realized when all of the equations are convex. Although less common, there are problems that give rise to the need to solve such a system. One of the most important applications is the linear complementarity problem (see Cottle, Pang and Stone [3]) which, given an (nxn) matrix M and an n-vector q, is the problem of trying to find two n-vectors w and x such that:

(1) $w = Mx + q$
(2) $w, x \geq 0$ (LCP)
(3) $w^T x = 0$

This problem can be stated equivalently as that of wanting to find a zero of the piecewise linear convex function $f = R^n \to R^n$, in which each coordinate function $f_i = R^n \to R^1$ is defined by $f_i(x) = \max[-(Mx+q)_{jk-}x_i]$. For this reason, other researchers have studied the problem of solving a system of convex equations. For example, Eaves [4] proposes a hemitropic approach for solving piecewise-linear convex equations and in Ortega and Rheingold [5] and more [6], variants of Newton's method are devised to solve differentiable convex equations.

Conditions placed on the system that ensure global convergence vary from one approach to another. In this work, an algorithm is developed to solve a constrained optimization problem associated with a system of differentiable convex equations. The advantages of the proposed approach include the ability: (1) to detect that the system has no solution; (2) to exploit the convexity of the equations and to establish global convergence under weaker conditions than previously know; and (3) to extend the algorithm from differentiable to piecewise-linear convex equations (and hence. To the Linear Complementarily Problem as a special ease).

In Section 2. A global algorithm is developed for finding a zero of a real-valued convex function of one variable or determining that no such point exists. The results in Section 2 are generalized in Section 3 to construct a constrained optimization algorithm for solving certain (n x n) systems of convex equations. Section 4 deals with computational implementation and conditions for convergence. Algorithmic performance and applications, including the Linear Complementarily Problem, are presented in Section 5. Which also includes a comparison of this approach with others for finding a zero of a system of nonlinear equations. Section 6 is a summary. The rest of this section is devoted to notation.

Let $\mathrm{P}^{\mathrm{n}} =$ n-dimensional Euclidean space. A vector $x \in \mathrm{R}^{\mathrm{n}}$ is always a column vector. The corresponding row vector is denoted by x^{T}; consequently, a row vector times a column vector produces the usual dot product. Subscripts refer to coordinates of vectors: superscripts are used for sequences. A sequences of vectors in R^{n} is written as $\{\mathrm{x}^{\mathrm{k}}\}$ for k= 0,1,..., or simply, $\{\mathrm{x}^{\mathrm{k}}\}$. A subsequence is written as $\{\mathrm{x}^{\mathrm{k}}\}$ for k $\in$ K, where K $\in$ {0,1,...}. A sequence of vectors $\{\mathrm{x}^{\mathrm{k}}\}$ in R^{n} converging to a vector $x \in \mathrm{R}^{\mathrm{n}}$ is written as $\{\mathrm{x}^{\mathrm{k}}\} \rightarrow \mathrm{x}$ as $\mathrm{k} \rightarrow \infty$, or, $\{\mathrm{x}^{\mathrm{k}}\} \rightarrow \mathrm{x}$. If a sequence $\{\mathrm{x}^{\mathrm{k}}\}$ converges to x along a subsequence K, it will be written as $\{\mathrm{x}^{\mathrm{k}}\} \rightarrow \mathrm{x}$ as $\mathrm{k} \in \mathrm{K}$. The vector whose coordinates are all 1 is denoted by c. If $x \in \mathrm{R}^{\mathrm{n}}$ then $x \geq 0$ means that for each $i = 1, \dots, n, x \geq 0$ and $0 \neq x$ means that there is an i such that $x_i \neq 0$. A sequence of point $\{x^k\} in\ R^1$ is said to be monotone if either (1) for all $k, x^k \leq x^{k+1}$ or (2) for all $k, x^k \geq x^{k+1}$. in the forme case the sequence is monotone inetreasing and, in the latter case, monotone decreasing.

A function f: Rn → R1 is said to be convex if for all x, y in R^n and $0 \leq \lambda \leq 1$ $f(\lambda x + (1-\lambda)y) + (1-\lambda)f(y)$. the reader is referred to Rockafellar [7] for an in-depth discussion of convex functions. A vector $a \in R^n$ is a subgradient of convex function f at a point $x \in \mathrm{R}^{\mathrm{n}}$ is a sub gradient inequality, and the set of all sub gradients of *f* at *x* is denoted by $\partial f(x)$. Also, if *f* is a convex function that is differentiable at a point $x \in \mathrm{R}^{\mathrm{n}}$ then $\nabla f(x)$ is the gradient of f at x. In Rockafellar [7] it is shown that if *f* is a convex function that is differentiable at $x \in \mathrm{R}$ then $\{\nabla f(x)\} = \partial f(x)$ and furthermore, for all $y \in \mathrm{R}^{\mathrm{n}}, f(y) \geq f(x) + \nabla f(x)^T(y-x)$. This is refereed to as the gradient inequality. A vector $d \in \mathrm{R}^{\mathrm{n}}$ is said to be a direction of recession of *f* if there is a $\beta \in \mathrm{R}^1$ and $x \in \mathrm{R}^{\mathrm{n}}$ such that for all $t \geq 0, f(x+td) \leq \beta$.

If *V* is a nonempty subset of R^{n}, then $(V)'$ denotes the set of all nonempty subsets of *V*. Let *S* and *T* be nonempty sets. A point-to-set map $M: S \rightarrow (T)'$ is said to be closed at $x \in S$ if whenever $\{x^k\} \rightarrow x, y^k \in M(x^k)$ for all k, and $\{y^k\} \rightarrow y$, it follows that $y \in M(x)$. M is closed if it is closed at each $x \in S$. With these notations, the algorithms and their proofs of convergence can be presented.

II. A GLOBAL ALGORITHM FOR FINDING A ZERO OF A CONVEX FUNCTION ON R^N

A modified Newton algorithm is proposed for finding a zero of a convex function $h: \mathrm{R}^1 \rightarrow \mathrm{R}^1$ or determining that no such point exists. If the algorithm terminates finitely, then either a zero of h is produced or no such exists. On the other hand, if the algorithm generates an infinite sequence of points, then either the sequence converges to a zero of h or one can conclude that zero of h exists. The algorithm proceeds like Newton's method, except that the derivative is replaced with an arbitrary sub gradient. The steps are as follows:

Step 1. Find $x^0 \in \mathrm{R}^1$ with $h(x^0) \geq 0$. If no such x^0 exists, stop. Otherwise set $i = 0$ and go to Step 2.

Step 2. If $h(x^i) = 0$, stop. Otherwise go to Step 3.

Step 3. Compute $a^i \in \partial h(x^0)$. If $a^i = 0$, stop. Otherwise, go to Step 4.

Step 4. Set $x^{\mathrm{i}+1} = x^i - h(x^i)/a^i$. If $\{x^0, x^1, \dots, x^{i+1}\}$ is not monotone, stop. Otherwise set $i = i + 1$ and go to Step 2.

The foregoing algorithm has the following properties, Theorem 1 *If the algorithm terminates finitely, then either a zero of h is obtained or no such point exists.* Proof. The algorithm may terminate finitely in any of the four steps. Each such termination is shown either to yield a zero of *h* or else establish that no zero exists.

Case 1. The algorithm steps in Step 1.

In this case, *h* clearly has no zero because $h(y) < 0$ for all $y \in \mathrm{R}^1$. By nothing that the only such convex function is $h(x) \equiv$ constant < 0, a technique for finding an x^0 with $h(x^0) \geq 0$ is presented at the end of this section.

Case 2. The algorithm terminates in Step 2 at iteration i.

In this case, a zero of *h* has been found, namely, x^{j}. If the algorithm does not stop in this step, then $h(x^i) > 0$ for each $i = 0,1, \dots$. To see this, note that the statement is true for $i = 0$. By construction, $x^{i+1} = x^i - h(x^i)/a^i$ where $a^i \in \partial h(x^i)$. From the subgradient inequality, $h(x^{i+1}) \geq h(x^i) + a^i(x^{i+1} - x^i) = h(x^i) + a^i(-h(x^i)/a^i) = 0$. Because $h(x^{i+1}) \neq 0$, it must be that $h(x^{j+1}) > 0$.

Case 3. The algorithm terminates in step 3 at iteration i.

In this case, *h* has no zero for, if $h(y) = 0$ for some $y \in \mathrm{R}^1$, the sub gradient inequality would yield the following contradiction:

$$0 = h(y) \geq h(x^i) + a^i(y - x^i) = h(x^i) > 0$$

Case 4. The algorithm terminates in Step 4 at iteration i.

In this case, *h* has no zero, for suppose $h(y) = 0$ for some $y \in \mathrm{R}^1$. Now $i \geq 1$ and assume, without less of generality, that $x^{i-1} < x^i$ and $x^{i+1} < x^i$. A contradiction is established by showing that $y \geq x^i$ and $y < x^i$.

To see that $y \geq x^i$, note that $a^{i-1} = -h(x^{i-1})/(x_i - x^{i-1}) < 0$ and so

$$0 = h(y) \geq h(x^{i-1}) + a^{i-1}(y - x^{i-1})$$
$$= h(x^{i-1}) + a^{i-1}(y - x^i) + a^{i-1}(x^i - x^{i-1})$$
$$= a^{i-1}(y - x^i)$$

Consequently, $y - x^i \geq 0$. It remains only to show that $y < x^i$. To this end, observe that $a^i = -h(x^i)/(x^{i+1} - x^i) > 0$ and so $0 = h(y) \geq h(x^i) + a^i(y - x^i) > a^i(y - x^i)$. Thus $y - x^i < 0$. This contradiction establishes the claim, completing the proof.

As result of Theorem 1, it may be assumed that the algorithm generates an infinite sequence of points $\{x^k\}$ for $k = 0,1, ...$ The next theorem establishes that either this sequence converges to a zero of *h* or no such point exists.

Theorem 2 Let $\{x^k\}$ for k = 0, 1, … be the sequence of points generated by the algorithm. Either $\{x^k\}$ converges to a zero of *h* or no such point exists.

Proof. Without loss of generality, assume that $\{x^k\}$ is monotone increasing. Now either this sequence is bounded above or not. In former case, $\{x^k\}$ converges to a zero of *h*; in the latter, case there is no zero of *h*.

Case 1. $\{x^k\}$ *is bounder above.*

In this case, the sequence $\{x^k\}$ converges to some $x \in \mathrm{R}^1$, it remains to show that $h(x) = 0$. To this end, for each $k = 0,1, ...,$ let $a^k \in \partial h(x^k)$ with $a^k(x^{k+1} - x^k) = -h(x^k)$. By Theorem 24.7 of Rockafellar [7], it is possible to choose a subsequence K and an $a \in \mathrm{R}^1$ such that $\{a^k\} \to a$ for $k \in K$. On taking the limit over $k \in K$ in the foregoing equality and using the continuity of *h* at *x*, it follows that

$h(x) = h(x^k) = -a^k(x^{k+1} - x^k) = -a(x - x) = 0$.

Case 2. The sequence $\{x^k\}$ *is unbounded above*.

In this case, no zero of *h* exists, for suppose $h(y) = 0$ for some $y \in \mathrm{R}^1$. By construction, for each $k = 0,1, ...$

$$a^k \equiv -h(x^k)/(x^{k+1} - x^k) < 0$$

Because $h(x^k) > 0$, one has that

$$0 = h(y) \geq h(x^k) + a^k(y - x^k) > a^k(y - x^k),$$

And consequently, $x^k < y$. This contradicts the fact that $\{x^k\}$ is unbounded above and establish the claim.

In order to implement this algorithm, it is necessary to find $x^0 \in \mathrm{R}^1$ with $h(x^0) \geq 0$. The next proposition indicates how to attempt to find such a point.

Proposition 1 *Let* $x \in \mathrm{R}^1$ *and suppose* $a \in \partial h(x)$ *with* $a \neq 0$. *Then* $x^0 = x - h(x)/a$ *satisfies* $h(x) \geq 0$

Proof. From the sub gradient inequality,

$h(x^0) \geq h(x) + a(x^0 - x) = h(x) + a(-h(x)/a) = 0$.

In this section, a global algorithm for finding a zero of a convex function $h: \mathrm{R}^1 \to \mathrm{R}^1$

or determining that none exists was presented. In Section 3, a constrained optimization approach is proposed for solving an $(n \times n)$ system of convex equations.

III. A CONSTRAINED OPTIMIZATION ALGORITHM FOR SOLVING CERTAIN SYSTEMS OF CONVEX EQUATION

Based on the observation that the value of the function at each point generated by the algorithm of Section 2 is nonnegative, the following constrained optimization approach to finding the zero of a function $f: \mathrm{R}^n \to \mathrm{R}^n$ is suggested:

$$\min e^T f(x) \qquad (COP)$$
$$s.t. f(x) \geq 0$$

While the foregoing approach can be applied to any system of equations, special benefits are realized when f is convex. Therefore, throughout this section, let $f: \mathrm{R}^n \to \mathrm{R}^1$ is convex and differentiable. For each $x \in \mathrm{R}^n$, let $\nabla f_i(x)$ be the gradient of f_i at *x* and let $f'(x)$ be the $(n \times n)$ derivative matrix in which row *i* is $\nabla f_i(x)^T$. Observe that the objective function of COP, being the sum of convex functions, is convex. However, the feasible region is not convex and, in fact, is the exterior of a convex set. Thus, COP is a nonconvex optimization problem that, in general, has local optima. In section 4, conditions are given to ensure that a global optimal can be obtained. In this section, the steps of an improvement algorithm are developed.

It is clear that if x^* is a feasible point for COP where the value of the objective function is 0, then $f(x^*) = 0$, and hence *′solves the system of equations. On the other hand, if *x* is a feasible point that is not a zero of *f*, then one might attempt to construct a feasible direction of descent at *x*. The next proposition shows that, under certain conditions, there is no point in doing so because it is possible to show that the system has no solution.

Proposition 2 *Let* $x \in R^n$ with $0 \neq f(x) \geq 0$, *If there is a* $\lambda \in R^n$ such that $\lambda \geq 0$, $\lambda^T f(x) = 0$ *and* $\lambda^T f(x) > 0$, *then there dose not exist a* $y \in R^n$ *such that* $f(v) = 0$.

Proof. Suppose that there is such a y. Then from the gradient inequality one has

$0 = f(v) \geq f(x) - f'(x)(y - x)$.

Multiplying both sides by the nonnegative vector λ^T yields the following contradiction:

$0 \geq \lambda^T f(x) - \lambda^T f'(x)(y - x) = \lambda^T f'(x) > 0$

The next proposition shows that if the hypotheses in proposition 2 are not satisfied, then is possible to find a potential direction of movement from *x*.

Proposition 3 Let $x \in R^n$ with $0 \neq f(x) \geq 0 \geq 0 \geq 0$. If there does not exist *a* $\lambda \in R^n$ with $\lambda \geq 0$, $\lambda^T f(x) > 0$ then there is a $d \in R^n$ such that $f'(x)d \leq -f(x)$

Proof. Consider the following linear program and its dual

min $(-f(x))^T w$ max $0^T u$

s.t. $f(x)^T w = 0$ and s.t. $f'(x)u \leq -f(x)$

$w \geq 0$ *u* unconstrained

Clearly any dual feasible vector $d \in R^n$ satisfied $f'(x)d \leq -f(x)$. To show that the dual is feasible, it is shown that the primal is feasible and bounded. Obliviously w = 0 is a primal feasible solution, so it remains only to show that the primal bounded. If it were unbounded, there would exist *a* $\lambda \in R^n$ such that $\lambda \geq 0$, $\lambda^T f(x) = 0$ and $\lambda^T f'(x) > 0$. The is contrary to the hypotheses and establishes the proposition.

The next proposition shows that the direction d from Proposition 3 is a direction of descent for the differentiable convex function $z(x) = e^T f(x)$ of COP.

Proposition 4 Let $x \in R^n$ with $0 \neq f(x) \geq 0$. If there is a $d \in R^n$ such that $f'(x)d \leq -f(x)$ then (1) $0 \neq f'(x)d \leq 0$ (in particular, $d \neq 0$) and (2) $VZ(x)^T d < 0$, so d is a direction of descent for Z at x.

Proof. The proof of (1) is oblivious because $f'(x)d \leq -f(x)$ and $0 \neq f(x) \geq 0$. To see (2), compute

$VZ(x)^T d < 0 = e^T f'(x)d \leq e^T(-f'(x)) = -e^T f(x) < 0$.

The next proposition provides additional conditions under which d is a feasible direction for COP at x.

Proposition 5 Let $x \in R^n$ with $0 \neq f(x) \geq 0$. Suppose there is a $d \in R^n$ such that (1)

$f'(x)d \leq -f(x)$ and (2) if $f_1(x) = 0$ then $(f'(x)d)_1 = 0$,

Under these conditions, there is

a $t_0 > 0$ such that for all $0 \leq t \leq t_0$, $f(x - td) \geq 0$.

Proof. Define $l(x) = \{i: l \leq i \leq n$ and $(f'(x)d)_1 < 0$; because d satisfied (1), by Proposition 4, $0 \neq f'(x)d \leq 0$, so l(x) is nonempty. Note that for each $i \in l(x)$, $f_1(x) \neq 0$ for, if there is an $i \in l(x)$ with $f_1(x) = 0$, then by (2). $(f'(x)d) = 0$ and $i \notin l(x)$. Therefore, one can define

$t_0 = \min\{-f_i(x)/f'(x)d\}_i : i \in I(x)$.

Clearly $t_0 > 0$. Now let $0 \leq t \leq t_0$. It is shown that $f_i(x + td) \geq 0$ for each $i \in \{1, \dots, n\}$.

Case 1. $i \notin I(x)$.

In this case, $(f'(x)d)_i = 0$ so, by the gradient inequality, $f(x + td) \geq f_i(x) + t(f'(x)d)_i = f_i(x) \geq 0$

Case 2. $i \in I(x)$.

In this case, $t \leq -f_i(x)/(f'(x)d)_i$. Hence agaim. $f_i(x + td) \geq f_i(x) + t(f'(x)d)_i \geq 0$.

Based on these four propositions, an algorithm is designed to search for a zero of *f* by attempting to solve COP. The algorithm consists of the following steps:

Step 1. Find $x^0 \in \mathrm{R}^n$ such that $f(x^0) \geq 0$. If none exists, stop. Otherwise set $k = 0$ and go to Step 2.

Step 2. If $f(x^k) = 0$, stop. Otherwise go to Step 3.

Step 3. If there is a $\lambda \in \mathrm{R}^n$ such that $\lambda \geq 0$, λ^T, $f'(x^k) = 0$ and $\lambda^T f(x^k) > 0$, stop. Otherwise go to Step 4.

Step 4. Compute $d^k \notin \mathrm{R}^n$ such that (1) $f'(x^k)d^k \leq -f(x^k)$ and (2) for each $1 \leq i \leq n$ with $f_i(x^k) = 0$, $(f'(x^k)d^k)_i = 0$. If no such dk exists, stop. Otherwise go to Step 5.

Step 5. Compute $I(x^k) = \{i: 1 \leq i \leq n\}$ and $(f'(x^k)d^k)_i < 0\}$,

$= \min\{-f_i(x^k)/(f'(x^k)d^k)_i : i \in I(x^*)\}$. Also compute t^k such that $Z(x^k + t^k d^k)$

$= \min\{Z(x^k + td^k): 0 \leq t \leq t_o^k\}$. Set $x^{k+1} = x^k + t^k d^k, k = k + 1$ and go to Step 2.

It should be noted that each time the algorithm generates a point $x^k \in \mathrm{R}^n, f(x^k) \geq 0$. This is a consequence of Proposition 5. In addition, the algorithm has the following properties.

Theorem 3 If finite termination occurs other than in Step 4, then either a zero of *f* is obtained or no such point exists.

Proof. The algorithm can terminate finitely in any of the first three steps. Each such termination is shown to yield either a zero of *f* or else establish that no zero exists.

Case 1. The algorithm stops in Step 1.

In this case, *f* clearly has no zero because there is no point at which *f* is nonnegative. A technique similar to that of Proposition 1 is presented at the end of this section to find an $x^0 \in \mathrm{R}^n$ where $f(x^0) \geq 0$.

Case 2. The algorithm stops in Step 2 at iteration k.

In this case, a zero of *f* has been computed, namely x^k. If this does not happen, it should be noted that $0 \neq f(x^k) \geq 0$ by Proposition 5.

Case 3. The algorithm stops in Step 3

In this case, Proposition 2 establishes that f has no zero and completes the proof.

Unfortunately, the algorithm can terminate finitely in Step 4 at a point *x* for which $f(x) \neq 0$ and yet there may be a zero of *f*. The following example illustrates this situation.

Example 1: Let $f: \mathrm{R}^2 \to \mathrm{R}^2$ be defined by

$f_1(x_1, x_2) = (x_1 - 2)^2 + x_2^2 - 4$ and

$f_2(x_1, x_2) = (x_1 - 1)^2 + x_2^2 - 1$

Clearly $(0,0)^T$ is a zero of *f*. To see that algorithm terminates finitely in Step 4 at a point that is not zero of *f*, let $x^0 = (4,0)^T$. Then $f(x^0) = (0,8)^T$ and $f'(x^0) =$

The algorithm does not stop in Step 1, 2, or 3; however, when it reaches Step 4, it attempts to find $d = (d_1, d_2)^T \in \mathrm{R}^2$ such that $4d_1 = 0$ and $6d_1 < -8$. This, of course, is not possible.

If, however, $f'(x)$ is a nonsingular matrix, the problem in Example 1 does not arise. Moreover, in this case all steps of the algorithm become computationally implementable. Section 4 deals with these issues as well as conditions for convergence.

Consider first problem of minimizing $xTQx + qTx$, where Q is an $(n \times n)$ positive definite symmetric matrix and q is an n-vector. Obtaining a solution to this problem is equivalent to finding a zero of the system of convex equations $f(x) = Qx + q$. By Proposition 7, starting with any $x \in \mathrm{R}^n$, one obtains $x^0 = x - f'(xk) - lf(xk) = -Q - lq$ and thus the starting point is the solution.

Another interesting observation is that if the function f is mapping from R^1 to R^1, then the algorithm of Section 4 virtually reduces to that Section 2. This is a good sign as the algorithm of Section 2 is globally.

To observe the performance of the algorithm, a computer program was written and several different types of small test problems were created. Because the objective here is to observe the algorithm's behavior on certain types of problems, no attempt is made to report specific details such as CPU time, number of function evaluations, and so on. Rather, the result on these test problems are used to interpret the qualitative behavior of the algorithm.

In the first group of regular convex equations, an $(n \times n)$ nonsingular matrix *C* is generated, with each C_{ij} being between -1 and 1. Then, a random n-vector a is obtained with each ai between 0 and 10. Using these data and row i of C, denoted by ci, coordinate

function i is defined by $fi(x) =$. It is easy to show that the zero of f is

$x^* = C - l \ln(a)$,

Where ln(a) is taken component wise. Problems ranging from size $n = 5$ to $n = 25$ are generated and, in each case, the zero was obtained, starting at the origin. However, it was observed that the algorithm never required the line-search subroutine. Instead, the algorithm performed exactly as Newton's method.

An observation at this point shows that the algorithm can be made quadratically convergent in the limit. Consider the sequence $\{xk\}$ generated by the procedure. If $\{xk\}$ contains a subsequence that converges to a zero of *f*, eventually $\{xk\}$ will enter the "radius of convergence" of Newton's method. When it does, one can eliminate the line search in Step 3, thus obtaining precisely Newton's method and realizing quadratic convergence.

In contrast to the previous group of the test problems, the line-search subroutine was required to solve the following continuously differentiable convex function $f: \mathrm{R}^2 \to \mathrm{R}^2$ defined by

$$f_1(x_1, x_2) = \begin{cases} x_2 + 3(x_1 - 1)^2 - \frac{1}{2}(x_1 - 1) - \frac{13}{16} & if\, \frac{3}{4} \leq x_1 \leq \frac{5}{4} \\ x_2 - 2x_1 + 1 & if\ x_1 \leq \frac{3}{4} \\ x_2 + x_1 - 2 & if\ x_1 \leq \frac{5}{4} \end{cases}$$

$$f_1(x_1, x_2) = \begin{cases} x_2 + 3(x_1 - 1)^2 - \frac{1}{2}(x_1 - 1) - \frac{13}{16} & if\, -\frac{5}{4} \leq x_1 \leq -\frac{3}{4} \\ x_2 - 2x_1 + 1 & if\ x_1 \leq -\frac{5}{4} \\ x_2 + x_1 - 2 & if\ x_1 \leq -\frac{3}{4} \end{cases}$$

The algorithm presented here, when started at (-3,5), moved first to the point (-, 5) and subsequently to the zero of f , namely (0,-1). However, Newton's method, when started at (-3,5), oscillates forever between (-3, 5) and (3, 5).

The next example shows that the Newton's method fails, but the algorithm in Section 3 can obtain the zero of *f*, in spite of the fact that the derivative matrix is singular at a point visited by the algorithm. To that end, suppose $f: \mathrm{R}^2 \to \mathrm{R}^2$with coordinate functions:

$f_1(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 2)^2 - 1$

$f_2(x_1, x_2) = + - 10.$

Obviously $f(1,3) = (0,0)$ and the derivatives of f is

$$f'(x) = \begin{bmatrix} 2(x_1 - 1) & 2(x_2 - 2) \\ 2x_1 & 2x_2 \end{bmatrix}$$

Starting at $x_0 = (2,4), f(x_0) = (4,10) \geq 0$ and $f'(x_0) =$. Because $f'(x_0)$ is singular, Newton's method cannot continue. But for the algorithm in Section 3, it can be verified that $d = (-1. -1)$ is feasible descent direction at x_0 and the algorithm continues and is able to find the zero of *f*.

The next type of convex system tested had each coordinates function in the form:

$f_i(x) = xTQ_ix + (q_i)Tx + b_i,$

where each Qi is a (5×5) randomly generated positive semi-definite symmetric matrix, q_i is a 5-vector, and bi is a known zero of the system. Specially, given Qi and q_i, b_i is computed as follows:

$b_i = -[eTQie + (qi)Te].$

Although each coordinates function is convex, no attempt is made to ensure non-singularity of the derivative matrix at each point.

The objective of this type of problem is to determine if the algorithm would obtain the known zero of the system, or whether it would find a point at which the derivative matrix is singular. Starting at $x = (2,2,2,2,2)$, the zero was indeed obtained. Moreover, no line search was required. Starting at the point $x = (2,0,2,0,2)$, the algorithm did obtain the zero of *f*, however, in so doing, the line search subroutine was used. When Newton's method was applied to the same problem starting at $x = (2,2,2,2,2)$, the algorithm did obtain the zero of *f*. When Newton's method was applied to the same problem starting at $x = (2,0,2,0,2)$, the solution was also obtained. However, the function $eTf(x)$ was observed to oscillated in value frequently, increasing to a maximum of 109 before finally reaching 0.

Finally, when started at $x = (0,2,0,2,0)$, the algorithm proposed here ended in three iterations at a point where the derivative matrix is singular. Newton's method fared no better. After 44 iterations, with the value of $eTf(x)$ oscillating, a different point of singularity was encountered. This example illustrates the sensitivity of the descent algorithm to the starting point when the system of equations is not regular.

Other Applications of the Constrained Optimization Approach

The approach of this paper can also used in an attempt to find the zero of a nonconvex system of equations $f: \mathrm{R}^n \to \mathrm{R}^n$. One would simply use any standard nonlinear programming package to solve the following associated COP:

$$\min e^T f(x) \qquad (COP)$$
$$s.t. f(x) \geq 0$$

The problem is that one might obtain a local minimum of COP that is not a zero of *f*.

Another application of this algorithm is to locate extrema of a real-valued function, because one is really searching for a zero of the gradient. If the gradient is convex, regular, and differentiable, then the algorithm of Section 4 can be used to search for such a point.

One of the more interesting and useful applications of the constrained optimization approach suggested in COP is that of solving piecewise-linier convex equations. One example of the need to solve such a system of equations arises in the linier complementarity problem which, given an (*n x n*) matrix *M* and an *n*-vector *q*. is the problem of wanting to find two n-vectors w and x such that:

(1) $w = Mx + q$

(2) $w, x \geq 0$ (LCP)

(3) $wTx = 0$

This problem can be stated equivalently as that of wanting to find a zero of the system of convex equations $f : Rn \to Rn$ in which each coordinate function $fi : Rn \to R1$ is defined by

$fi(x) = \max \{-(Mx + q)i - xi\}$.

It is easy to show that these coordinate functions are piecewise-linier and convex. It should be noted. However, that algorithm of Section 3 is not applicable to this problem because of the lack of differentiability in the piecewise -linier problem. In exchange for this deficiency. However, it is possible to develop a finite algorithm for solving the piecewise-linier problem. The general approach is to solve a finite sequence of (feasible) linier programing problems while strictly improving the objective function of COP. Specifically, since f is piecewise linier, there are a finite number of "pieces of linearity", say $P1, \ldots, Pk$ (polyhedral).

In each piece of linearity *Pi*, *f* is linier, that is, there exists an $(n \times n)$ matrix *Ai* and *n*- vector *ai* such that, for all x in *Pi*. $F(x) = Aix + ai$,. with this notation, solving COP is equivalent to solving the following finite collection of linier programming problem:

$$\begin{array}{ll}\min & e^T f(x) \\ s.t. & f(x) \geq 0\end{array} \equiv \min_{j\leq i\leq k}\left[\begin{array}{ll}\min & e^T f(x) \\ s.t & f(x) \geq 0 \\ & x \text{ in } P'\end{array}\right] \equiv \min_{j\leq i\leq k}\left[\begin{array}{ll}\min & e^T(A^i x + a^i) \\ s.t & A^i x + a^i \geq 0 \\ & x \text{ in } P'\end{array}\right]$$

$$\equiv \min_{j\leq i\leq k}\left[\begin{array}{ll}\min & (e^T A^i)x + e^T a^i \\ s.t & A^i x + a^i \geq 0 \\ & x \text{ in } P'\end{array}\right] \equiv \min_{j\leq i\leq k}[LP^{00}]$$

The algorithm starts by finding a feasible LP that is necessarily bounded. If the optimal solution x^* to that LP satisfies $f(x^*) = 0$, then a zero of *f* has been found. Otherwise, a new LP containing x^* is solved. under suitable conditions (regularity, for example), it can be shown that the objective functions of the new I.P is strictly better than that of the old I.P. in this way. A zero of f is found in finite number of steps (because there are only a finite number of LP's). the theoretical details for using this approach with COP to solve the LCP are given in Solow and Sengupta {10} (there, for example, it is shown that when M is a P-matrix, the convex function f corresponding the LCP is regular). The computational implementation of this algorithm is presented in Paparrizos and Solow [11].

IV. COMPUTATIONAL IMPLEMENTATION AND CONVERGENCE FOR REGULAR CONVEX FUNCTION

In section, an algorithm is developed for attempting to find a zero of a system of differentiable convex equations by attempting solve COP. In this section, it is shown that for regular, differentiable. Convex function (that is, a convex function whose $(n \times n)$ derivative matrix is nonsingular at every point in R^n), all steps of the algorithm for COP can be performed on a computer. Suppose throughout this section that $f : R^n \rightarrow R^n$ is a differentiable and regular convex function. The first step of the algorithm for solving COP requires finding a point $X^n < R^n$ where $f(x^n) \geq 0$. The next proposition show's that such a point can be obtained by starting anywhere in Rn and moving to the first Newton iterate.

Proposition 6 *if x is any point in* R^n. *then* $x^0 = x - f'(x)^{-1} f(x)$ *satisfies* $f(x^0) \geq 0$.

Proof. By the gradient inequality $f(x^0) \geq f(x) + f'(x)(x^0 - x) = f(x) - f(x) = 0$.

As a result of the regularity assumption. Step 3 of the algorithm need not be performed because there cannot be a $\lambda \geq 0$ with $\lambda^t f'(x) = 0$ and $\lambda^t f(x) > 0$. The reason is that if there were such a λ. Regularity would imply that $\lambda = 0$ and hence $\lambda^t f(x) = 0$, a contradiction. Thus, step 3 can be omitted.

The next proposition shows that the newton direction $d = - f'(x)^{-t} f(x)$ satisfies the two conditions in step 4.

Proposition 7 if $x \in R^n$ with $0 \neq f(x) \geq 0$. Then the direction $d = - f(x)^{-1} f(x)$ satisfies (1) $f'(x)\, d \leq -f(x)$ and (2) for each $1 \leq I \leq n$ with $f_i(x) = 0$. $(f'(x)\, d)_I = 0$

Proof. Now (1) can be seen by nothing that $f'(x)d = - f(x)$ and so $f'(X)\, d \leq - f(X)$. for (2), note that if $f_{,}(x)=0$, then $(f(x)\, d) = -f_i(x)=0$.

Finally, step 5 of the algorithm is a form of line search for the convex function $z = e^t f$ at a point x in direction of descent d over the interval [0,]. The next proposition shows that this line search can be done over the compact interval [0,1].

Proposition 8 *if* $x \in R^n$ with $f(x) \geq 0$ and $d = - f'(x)^{-1} f(x)$, then for all $t \in [0,1]$.

$F(x + td) \geq 0$

Proof. It follows from the gradient inequality that for all $t \in [0,1]$,$f(x + td) \geq f(x) + t$

$F'(x) = f(x) - \text{if}'(x) f'(x)^{-1} f(x) = (1-t)f(x) \geq 0$.

One method for performing the line search of h(t)= Z(x + td) over the interval [0,1] is discussed . the next proposition states that if $h'(1) \leq 0$, then the line search is over, Z(x + d) is the minimum . on the other hand, if $h'(1)>0$, then a bisection algorithm (see Zangwill [8]. For example) can be used because $h'(0) < 0$ by proposition 4.

Proposition 9 let x, d Є Rn be fixed and define: R1→R1 by h(t) = Z(x + td). IF $h'(1) \leq 0$, then $h(1) \leq h(t)$ for All $t \in [0,1]$.

Proof. H is convex and differentiable so for any $t \in [0,1]$. The gradient inequality yields $h(t) \geq h(1) + h'(1)(t-1) \geq h(1)$.

V. COMPARISON WITH OTHER METHODS

While there is a large and rich literature pertaining to the solution of nonlinear equations, few methods have exploited the property of convexity. In Ortega and Rheinboldt [5] and More [6], a monotone convergence, not only must the derivative matrix be nonsingular, but also, its inverse must be nonnegative everywhere. In comparison, the conditions given in Theorem 4 for the proposed algorithm are significantly weaker.

With regard to solving a system of piecewise linear convex equations, Eaves [4] proposed a homotopy approach in which the only condition for ensuring global convergence is regularity of all the sub gradient matrices. For the approach of COP proposed here, precisely the same conditions are required (see Solow and Sengupta [10]). More recently, Kojima and Shindo [12] extended Newton's method to handle a system of pricewise-continuous equations, and Kummer [13] has done so for a non-differentiable system of equations. Their results, however, provide only local convergence.

While the constrained optimization approach proposed here could, in theory, be used to solve a general nonlinear system of equations, it has a number of disadvantages with regard to other well-established methods. For example, one disadvantages of the proposed approach is the need to perform a line search that maintains nonnegativity of the system equations. While doing so for a system of convex equations is greatly simplified, this task could be difficult and time consuming to perform on a general system of nonlinear equations. In Dennis and Schabel [14], several Newton-type methods are given for solving systems of equations that do not require a line search, but rather, calculate a fixed step size each iteration. Global convergence of such methods, based on the work of Wolfe [15, 16], is established using the Euclidean norm of the system as an objective function, assuming a Lischitz continuity condition on the norm, (More recently, Pang [17] extended Newton's method to solve B-differentiable equations and gave local and global convergence conditions).

In contrast, global convergence of the method proposed here requires a line search but uses the sum of the coordinates functions as an overall objective function to measure improvement.

VI. CONCLUSION

A constrained optimization approach has been presented for solving a system of nonlinear equations by minimizing the sum of the coordinate functions while keeping each coordinate function nonnegative. While any existing nonlinear programming package can be used to solve the resulting problem, specialized algorithms have been developed when the system of equations is convex. Advantages of this approach include the ability to detect that the system of equations has no solution, and to extend the method to solve systems of piecewise-linier convex equations (of which the Linear Complementarity Problems a special case). Global convergences of this algorithm under the reasonable conditions of regularity, continuity of the derivative matrix, and the existence of a coordinate function that has no non-zero direction of recession has been established.

REFERENCES

[1] Rheinboldt, W.C. Methods for Solving Systems of Nonlinear Equations (2nd Edition). Society for Industrial and Applied Mathematics, 1998.

[2] Allgower, E.L and Georg, K. Computational Solution of Nonlinear Systems of Equations. American Society, 1990.

[3] Cottle, R.W, Pang, J.S and Stone, R.E. The Linear Complementarity Problem. Academic Press, Inc., 1992.

[4] Eaves, B. C. "Solving Piecewise Linear Convex Equations". Mathematical Programming, 1:96-119, 1974.

[5] Ortega, J. M. and Reinboldt, W. C. Iterative solutions of Nonlinear equations in several variables. Academic Press, Princeston, NJ, 1968.

[6] More, J.J. "Global Convergence of Newton-Gauss-Seidel Methods". SIAM Journal on Numerical Analysis, 8(2):325-336, June 1971.

[7] Rockafellar, R. T. Convex Analysis. Princeton University Press. Princeton, NJ, 1968.

[8] Zangwil, W. I. Nonlinear Programming: A Unified Approach, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[9] Stoer, J. and Witzgall, C. Convexity and Optimization in Finite Dimensions 1. Springer-Verlag, NY, 1970.

[10] Sollow, D. and Sengupta, S. "A finite Descent Theory for Linear Programming, Piecewise Linear Convex Minimization, and the Linear Complementarity Problem". Naval Research Logistics Qurterly, 32:417-431, 1985.

[11] Paparrizos, K and Solow, D. "A Finite Improvement Algorithm for the Linear Complementarity Problem". European Journal of Operations Research, 3:305-324. 1989.

[12] Kojima, M. and Shindo, S. "Extensions of Newton and Quasi-Newton Methods to Systems of PCI Equations". Journal of The Operation Research Society of Japan, 29:352-374, 1986.

[13] Kummer, B. "Newton's Method for Non-differentiable Functions". In J. Guddat, B. Bank, H, Hollatz, P, Kall, D, Klatte, B, Kummer, K. Lommatzsch, K. Tammer, M, Vlach, and K. Zimmermann, editor, Advances in Mathematical Optimization, pages 114-125, Academic-Verlag, Berlin. 1988.

[14] Dennis, J. E., Jr. and Schnabel, R. B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice Hall, Englewood Cliffs, NJ, 1983.

[15] Wolfe, P. "Convergence Conditions for Ascent Methods". SIAM Review, 11:226-235, 1969.

[16] Wolfe, P. "Convergence Conditions for Ascent Methods II Some Corrections". SIAM Review, 13:185-188, 1971.

[17] Pang, J. S. "Newton's Method for B-Differentiable Equations". Mathematics of Operations Research, 15:311-341, 1990.

Pasukat Sembiring
Drs(USU)
M.Si. (USU)
Fakultas Matematika dan Ilmu Pengetahuan Alam / Faculty of Mathematics ND Natural Sciences