

A Fixed Point Method for Convex Systems

Morteza Kimiaei¹, Farzad Rahpeymai²

¹Department of Mathematics, Razi University, Kermanshah, Iran
²Department of Mathematics, Payame Noor University, Tehran, Iran
 Email: morteza.kimiaei@gmail.com, rahpeyma_83@yahoo.com

Received June 26, 2012; revised September 27, 2012; accepted October 6, 2012

ABSTRACT

We present a new fixed point technique to solve a system of convex equations in several variables. Our approach is based on two powerful algorithmic ideas: operator-splitting and steepest descent direction. The quadratic convergence of the proposed approach is established under some reasonable conditions. Preliminary numerical results are also reported.

Keywords: Convex Equations; Least Squares; ℓ_1 -Regularization Problems; Fixed Point; Quadratically Convergence

1. Introduction

System of convex equations is a class of problems that is conceptually close to both constrained and unconstrained optimization and often arise in the applied areas of mathematics, physics, biology, engineering, geophysics, chemistry, and industry. Consider the following system of convex equations

$$F(x) = 0, x \in R^n \quad (1)$$

in which $F: R^n \rightarrow R^m$ is a convex continuously differentiable function. It is noticed that if $F(x) = Ax - b$ the system (1) is a linear system of equations and there are a lot of approaches to solve this problem. One of the most interesting methods for solving linear system is fixed point methods that have been comprehensively studied by many authors. For example, shrinkage, subspace optimization and continuation [1], fixed-Point continuation method [2], nonlinear wavelet image processing [3], EM method [4], iterative thresholding method [5] and fast iterative thresholding [6]. The system (1) is called an overdetermined system whenever $m > n$ and under-determined for $n > m$. If $m = n$, we obtain a square system of convex equations. Most of the time, we wish to find a proper $x \in R^n$ such that (1) holds as closely as possible. This means that our objective is to reduce $\|F(x)\|_2$ as much as and, if possible, reduce it to zero. Hence the system of convex Equations (1) can be written as an unconstrained optimization problem

$$\min f(x) \text{ s.t } x \in R^n \quad (2)$$

in which

$$f(x) = \frac{1}{2} \|F(x)\|_2^2. \quad (3)$$

It is obvious that

$$g(x) = \nabla f(x) = J^T(x)F(x),$$

where $J(x)$ is the Jacobin matrix of $F(x)$. In this work, we consider a ℓ_1 -regularized least squares problem for system (1):

$$\min \varphi(x) = \|x\|_1 + \mu f(x) \quad (4)$$

in which $\mu > 0$ is a parameter. We note that if $F = (F_1, F_2, \dots, F_m)^T$ and any F_i is convex, then F is convex. On the other hand, convexity of $\|\cdot\|_2^2$ implies that f is convex. Therefore, φ is a convex function.

As an example, Hale, Yin, and Zhang in [2] presented a fixed-point continuation method for ℓ_1 -regularized minimization that based on operator-splitting and continuation:

$$x^{k+1} = (\text{shrink}_\nu \circ h)(x^k), \quad (5)$$

in where $\nu = \frac{\tau}{\mu}$ and mappings $\text{shrink}_\nu, h: R \rightarrow R$ are defined as :

$$h(\cdot) = I(\cdot) - \tau g(\cdot), \quad (6)$$

$$\text{shrink}_\nu(\cdot) = \text{sgn}(\cdot) \otimes \max\{|\cdot| - \nu, 0\}. \quad (7)$$

The operator \otimes in the right hand side of relation (7) denote the component-wise product of $\text{sgn}(\cdot)$ and $\max\{|\cdot| - \nu, 0\}$. Because of the parameter τ is constant, the number of iterations and computational costs increase and so it is not suitable. To overcome the mentioned disadvantage, we create innovation in the parameter τ based on the steepest descent direction:

$$\tau^k = \frac{g^{kT} g^k}{(J^k g^k)^T (J^k g^k)} \tag{8}$$

The analysis of the new approach shows that it inherits both stability of fixed point methods and low computational cost of steepest descent methods. We also investigate the global convergence to first-order stationary points of the proposed method and provide the quadratic convergence rate. To show the efficiency of the proposed method in practice, some numerical results are also reported.

The rest of this paper is organized as follows: In Section 2, we describe the motivation behind the proposed algorithm in the paper together with the algorithm's structure. In Section 3, we prove that the proposed algorithm is globally convergent. Preliminarily numerical results are reported in Section 4. Finally, some conclusions are expressed in Section 5.

2. The New Algorithm: Motivation and Structure

In this section, we first introduce a fixed point algorithm for small-scale convex systems of equations. Then, given some properties of the algorithm and investigate its global convergence as well as the quadratic convergence rate. The objective function in (4) is a sum of two convex functions. By convex analysis, minimizing a convex function $\varphi(x)$ is equivalent to finding a zero of the subdifferential $\partial\varphi(x)$. Let X^* be the set of optimal solutions of (4). It is well-known that an optimality condition for (4) is

$$x^* \in X^* \Leftrightarrow 0 \in \text{sgn}(x^*) + \mu g(x^*),$$

or equivalently,

$$x^* \in X^* \Leftrightarrow \mu g(x^*) \begin{cases} = -1 & x_i^* > 0 \\ \in [-1, 1] & x_i^* = 0 \\ = 1 & x_i^* < 0 \end{cases} \tag{9}$$

where 0 denotes the zero vector in R^n and x_i^* is i -th component of x^* . It follows readily from (9) that 0 is an optimal solution of (4) if and only if $\mu \|g(0)\|_\infty \leq 1$, or in other words,

$$0 \in X^* \Leftrightarrow \mu \leq \frac{1}{\|g(0)\|_\infty}$$

Therefore, it is easy to check whether 0 is a solution of (4) (see [2]).

One of the simplest methods for solving (4) generates a sequence $\{x^k - \tau^k g^k\}_{k \geq 0}$ that based on steepest descent direction. Here,

$$\tau^k = \frac{g^{kT} g^k}{(J^k g^k)^T (J^k g^k)}$$

and $g^k = J^k F^k$. Note that if the system (1) be a linear system of equations, then $\nabla g = 2A^T A$ and

$\tau \in \left(0, \frac{2}{\lambda_{\max}(A^T A)}\right)$. Here, the system (1) is a convex

system of equations, then $\nabla g^k = J^{kT} J^k$ and for the purpose of our analysis, we will always choose

$0 < \tau^k < \frac{1}{\lambda_{\max}(J^{kT} J^k)}$. Using these information, we

present a proximal regularization of the linearized function f at x^k for problem (4) (see [7]), and written it equivalently as

$$\begin{aligned} \hat{x}^{k+1} = \mu^k \arg \min_x \left\{ f(x^k) + g^{kT} (x - x^k) \right. \\ \left. + \frac{1}{2\tau^k} \|x - x^k\|_2^2 + \frac{1}{\mu^k} \|x\|_1 \right\}. \end{aligned} \tag{10}$$

After ignoring constant terms, (10) can be rewritten as

$$\begin{aligned} x^{k+1} = \mu^k \arg \min \left\{ \frac{1}{2\tau^k} \|x - (x^k - \tau^k g^k)\|_2^2 + \frac{1}{\mu^k} \|x\|_1 \right\} \\ = \arg \min_x \sum_{i=1}^n \left\{ \frac{\mu^k}{2\tau^k} (x_i - (x^k - \tau^k g^k)_i)^2 + |x_i| \right\}. \end{aligned} \tag{11}$$

Notice that the function in the problem (11) is minimized if and only if each functions

$$q_i(x) = \frac{\mu^k}{2\tau^k} (x_i - (x^k - \tau^k g^k)_i)^2 + |x_i|, \quad i = 1, 2, \dots, n$$

is minimized. If we take $\nu^k = \frac{\tau^k}{\mu^k}$, then we can simply

obtain the minimizer of $q_i(x)$ as follows:

$$x_i^* = \begin{cases} (x^k - \tau^k g^k)_i - \nu^k & (x^k - \tau^k g^k)_i > \nu^k \\ 0 & |(x^k - \tau^k g^k)_i| \leq \nu^k \\ (x^k - \tau^k g^k)_i + \nu^k & (x^k - \tau^k g^k)_i < -\nu^k \end{cases}$$

Therefore, $x_i^{k+1} = x_i^*$ and the solution of (4) is obtained. Now, based on above arguments, a new fixed point algorithm can be outlined as follows:

Algorithm 1: Fixed point algorithm (FP)

Input: Choose an initial point $x^0 \in R^n$ and constants $\varepsilon > 0$, $0 < \mu^0 < 1$, $\eta \in [1, 10]$.

Begin

$\tau^0 = 1$; $k \leftarrow 0$

While ($\|F_k\|_2 \geq \varepsilon$ and $k \leq k_{\max}$) **{Start loop}**

Step 1: {Parameter shrinkage calculation}

$$\nu^k = \frac{\tau^k}{\mu^k};$$

Step 2: {Operation shrinkage calculation}

$$x^{k+1} = (\text{shrink}_{\nu} \circ h)(x^k);$$

Step 3: {Parameters update}

Calculate τ^k as (8);

Generate $\mu^{k+1} = \eta\mu^k$;

Increment k by one and go to Step 1;

End While {End loop}

End

3. Convergence Analysis

In this section, we will give the convergence analysis of the proposed algorithm given in Section 2. In the convergence analysis, we need the following assumption:

(H1) Problem (4) has an optimal solution set $X^* \neq \emptyset$, and there exists a set

$$\Omega = \{x : \|x - x^*\|_2 \leq \delta\} \supset X^*,$$

for some $x^* \in X^*$ and $\delta > 0$, such that f is twice continuously differentiable on Ω and

$$\hat{\lambda}(J^T(x)J(x)) = \max_{x \in \Omega} \lambda_{\max}(J^T(x)J(x)),$$

for all $x \in \Omega$. Using the mean-value theorem, we have

$$\begin{aligned} g(x) - g(y) &= \left(\int_0^1 J(y + t(x-y))^T J(y + t(x-y)) dt \right)^T (x-y) \\ &= \bar{J}(x,y)^T \bar{J}(x,y)(x-y), \end{aligned}$$

for any $x, y \in \Omega$.

(H2) There exists a constant $M > 0$ such that

$$\|J(x^k)\| \geq M, \forall k \in N \cup \{0\}.$$

Lemma 3.1. By the definition of τ^i and τ^j satisfying (8), we have

$$x^i = x^j \Leftrightarrow \tau^i = \tau^j$$

for any $x^i, x^j \in \Omega$.

Proof. Suppose that $x^i = x^j$ then $g^i = g^j$ and $J^{iT}J^i = J^{jT}J^j$. So by (8), we conclude that $\tau^i g^i = \tau^j g^j$. Now suppose that $\tau^i = \tau^j$. Then, we show that $x^i = x^j$. By contradiction, we assume that $x^i \neq x^j$. Then $g^i \neq g^j$ and $J^{iT}J^i \neq J^{jT}J^j$.

Therefore we have $\tau^i g^i \neq \tau^j g^j$ that is a contradiction.

In the following lemma, we show that the new choice of τ^k satisfying in the lemma 4.1 and corollary 4.1 of

[2] when $\tau^k \in \left(0, \frac{1}{\lambda_{\max}(J^{kT}J^k)} \right)$.

Lemma 3.2. Under assumption (H1), the choice of τ^i and τ^j result in $h(\cdot) = I(\cdot) - \tau g(\cdot)$ is nonexpansive over Ω , i.e. for any $x^i, x^j \in \Omega$

$$\|h(x^i) - h(x^j)\|_2 \leq \|x^i - x^j\|_2. \tag{12}$$

Moreover, $g^i = g^j$ whenever equality holds in (12).

Proof. Let $\bar{J} = \bar{J}(x^i, x^j)$ and $d = x^i - x^j$. Now, we have two cases:

1) If $\tau^j \leq \tau^i$, then

$$\begin{aligned} h(x^i) - h(x^j) &= x^i - x^j - \tau^i g^i + \tau^j g^j \tag{13} \\ &\leq d - \tau^i (g^i - g^j) = d - \tau^i (\bar{J}^T \bar{J} d) \\ &= (I - \tau^i \bar{J}^T \bar{J})d. \end{aligned}$$

Hence,

$$\begin{aligned} \|h(x^i) - h(x^j)\|_2 &\leq \|(I - \tau^i \bar{J}^T \bar{J})d\|_2 \\ &\leq \|I - \tau^i \bar{J}^T \bar{J}\|_2 \|x^i - x^j\|_2 \\ &\leq \max\left\{1 - \tau^i \hat{\lambda}(\bar{J}^T \bar{J}), 1\right\} \|x^i - x^j\|_2 \\ &\leq \max\left\{\left| \frac{\hat{\lambda}(J^{iT}J^i) - \hat{\lambda}(\bar{J}^T \bar{J})}{\hat{\lambda}(J^{iT}J^i)} \right|, 1\right\} \|x^i - x^j\|_2. \end{aligned}$$

Let $p = \bar{J}d$. By the lemma 3.1, we have $d = x^i - x^j \neq 0$ if and only if $\tau^i g^i - \tau^j g^j \neq 0$.

Then

$$\begin{aligned} \|h(x^i) - h(x^j)\|_2 = \|d\|_2 &\Leftrightarrow \|d - (\tau^i g^i - \tau^j g^j)\|_2 = \|d\|_2 \\ \Leftrightarrow -2d^T (\tau^i g^i - \tau^j g^j) + (\tau^i g^i - \tau^j g^j)^T (\tau^i g^i - \tau^j g^j) &= 0 \\ \Leftrightarrow \frac{(\tau^j g^j - \tau^i g^i)^T (\tau^j g^j - \tau^i g^i)}{d^T (\tau^i g^i - \tau^j g^j)} &= 2 \end{aligned}$$

$$\text{if } \tau^i g^i - \tau^j g^j \neq 0,$$

by the equation (13), we obtain

$$1 < 2 \leq \frac{(-\tau^i \bar{J}^T \bar{J})^T (-\tau^i \bar{J}^T \bar{J})}{\tau^i d^T \bar{J}^T \bar{J} d} \Rightarrow 1 < 2 \leq \frac{\tau^i p^T \bar{J}^T \bar{J} p}{p^T p},$$

which contradicts to $\frac{p^T \bar{J}^T \bar{J} p}{p^T p} \leq \hat{\lambda}(\bar{J}^T \bar{J})$.

Hence $p = 0$ so that

$$g(x^i) - g(x^j) = \bar{J}^T p = 0.$$

2) If $\tau^i \leq \tau^j$, then

$$h(x^j) - h(x^i) = x^j - x^i - \tau^j g^j + \tau^i g^i \tag{14}$$

$$\begin{aligned} \leq d - \tau^j (g^j - g^i) &= d - \tau^j (\bar{J}^T \bar{J}) \\ &= (I - \tau^j \bar{J}^T \bar{J})d. \end{aligned}$$

Hence,

$$\begin{aligned} \|h(x^j) - h(x^i)\|_2 &\leq \|(I - \tau^j \bar{J}^T \bar{J})d\|_2 \\ &\leq \|I - \tau^j \bar{J}^T \bar{J}\|_2 \|x^j - x^i\|_2 \\ &\leq \max\left\{ |1 - \tau^j \hat{\lambda}(\bar{J}^T \bar{J})|, 1 \right\} \|x^j - x^i\|_2 \\ &\leq \max\left\{ \left| \frac{\hat{\lambda}(J^{jT} J^j) - \hat{\lambda}(\bar{J}^T \bar{J})}{\hat{\lambda}(J^{jT} J^j)} \right|, 1 \right\} \|x^j - x^i\|_2. \end{aligned}$$

Let $p = \bar{J}d$. By the lemma 3.1, we have $d = x^i - x^j \neq 0$ if and only if $\tau^j g^j - \tau^i g^i \neq 0$. Then

$$\begin{aligned} \|h(x^j) - h(x^i)\|_2 = \|d\|_2 &\Leftrightarrow \|d - (\tau^j g^j - \tau^i g^i)\|_2 = \|d\|_2 \\ \Leftrightarrow -2d^T (\tau^j g^j - \tau^i g^i) + (\tau^j g^j - \tau^i g^i)^T (\tau^j g^j - \tau^i g^i) &= 0 \\ \Leftrightarrow \frac{(\tau^i g^i - \tau^j g^j)^T (\tau^i g^i - \tau^j g^j)}{d^T (\tau^j g^j - \tau^i g^i)} &= 2 \\ \text{if } \tau^j g^j - \tau^i g^i \neq 0, & \end{aligned}$$

by the Equation (14), we obtain

$$1 < 2 \leq \frac{(-\tau^j \bar{J}^T \bar{J})^T (-\tau^j \bar{J}^T \bar{J})}{\tau^j d^T \bar{J}^T \bar{J} d} \Rightarrow 1 < 2 \leq \frac{\tau^j p^T \bar{J}^T \bar{J} p}{p^T p},$$

which contradicts to $\frac{p^T \bar{J}^T \bar{J} p}{p^T p} \leq \hat{\lambda}(\bar{J}^T \bar{J})$. Hence $p = 0$

so that

$$g(x^j) - g(x^i) = \bar{J}^T p = 0,$$

which completes the proof.

Corollary 3.3. (Constant optimal gradient). From (H1) assumption, for any $x^* \in X^* \cap \Omega$, there is a vector $\gamma^* \in R^n$ such that $g(x^*) = \gamma^*$.

Let X^* be the solution set of (4), $x^* \in X^*$, and γ^* be the vector specified in corollary 3.2. Then, we define

$$\begin{aligned} L &= \{i : \mu |\gamma_i^*| < 1\}, E = \{i : \mu |\gamma_i^*| = 1\}, \\ \omega &= \{\nu(1 - \mu |\gamma_i^*|) : i\}, \end{aligned}$$

where $\nu = \frac{\tau}{\mu}$. We will show that the sequence generated by (5) is finite convergence for components in L and is quadratic convergence for components in E .

It is obvious from the optimality condition (9) that

$L \cup E = \{1, 2, \dots, n\}$, and for any $x^* \in X^*$, we have

$$\text{supp}(x^*) = \{i : x_i^* \neq 0\} \subseteq E, x_i^* = 0, \forall i \in L.$$

Hale, Yin, Zhang in [2] establishes the finite convergence properties of $\{x^k\}$ stated in the following of theorem. The proof of the theorem 3.4 and 3.5 is similar to the theorem 4.1 and 4.2 in [2].

Theorem 3.4. Under assumption (H1), the sequence $\{x^k\}$ is generated by the fixed point iteration (5) applied to problem (4) from any starting point $x^0 \in \Omega$ converges to some $x^* \in X^* \cap \Omega$. In addition, for all but finitely many iterations, we have

$$x_i^k = x_i^* = 0 \quad \forall i \in L, \tag{15}$$

$$\text{sgn}(h_i(x^k)) = \text{sgn}(h_i(x^*)) = -\mu g_i^*, \quad \forall i \in L \tag{16}$$

where the numbers of iterations not satisfying (15) and (16) do not exceed $\|x^0 - x^*\|_2^2 / \omega^2$ and $\|x^0 - x^*\|_2^2 / \nu^2$, respectively.

Theorem 3.5. (The quadratic case). Let f be a convex quadratic function that is bounded below, $J^{kT} J^k$ be its Hessian, and τ^k satisfy

$$\tau^k \in \left(0, \frac{1}{\lambda_{\max}(J^{kT} J^k)} \right),$$

then the sequence $\{x^k\}$ is generated by the fixed point iteration (5) applied to problem (4) from any starting point $x^0 \in \Omega$ converges to some $x^* \in X^* \cap \Omega$. In addition, for all but finitely many iterations, we have (15) - (16) hold for all but finitely many iterations.

Lemma 3.6. Suppose assumptions (H1) and (H2) holds. Then, we have

$$\lim_{k \rightarrow \infty} \|F_k\|_2 = 0.$$

Proof. From (9), we can obtain

$$\mu_k \|g_k\|_\infty \leq 1,$$

then, by (H2) and the above inequality, we have

$$\mu_k M \|F_k\|_\infty \leq 1 \Rightarrow \|F_k\|_2 \leq \|F_k\|_\infty \leq \frac{1}{\mu_k M} = \varepsilon.$$

For sufficiently large k , we conclude that

$$\lim_{k \rightarrow \infty} \|F_k\|_2 = 0.$$

Now, consider the sequence $\{x^k\}$ generated by the FP algorithm. According to the fixed point iterations (5), it converge to some point $x^* \in X^* \cap \Omega$. We will show that the convergence is quadratic. In order to do, the following additional assumption is required:

(H3) The following condition

$$\|\bar{G}^k - G^k\| = O(\|x^{k+1} - x^k\|), \tag{17}$$

holds, in where $\bar{G}^k = J(x^k, x^{k+1})^T J(x^k, x^{k+1})$ and $G^k = J(x^k)^T J(x^k)$. Suppose that k is enough large so that $x_i^k = x_i^* = 0$, for all $i \in L$. Also, suppose that

$$C_{EE} = [C_{i,j}]_{i,j \in E}, \quad \|F(x^k)\| \leq 10^{-5}.$$

denote the square sub-matrix of the matrix C corresponding to the index set E . Firstly we suppose that $\tau^* \leq \tau$, then the mean-value theorem yields

$$\begin{aligned} h_E(x^k) - h_E(x^*) &= x_E^k - x_E^* - \tau_E^k g_E^i + \tau_E^* g_E^* \quad (18) \\ &\leq x_E^k - x_E^* - \tau_E^k (g_E^k - g_E^*) = x_E^k - x_E^* - \tau_E^k (\bar{G}_{EE}^k (x_E^k - x_E^*)) \\ &= (I - \tau_E^k \bar{G}_{EE}^k)(x_E^k - x_E^*) \end{aligned}$$

Since $x^{k+1} = (\text{shrink}_v \circ h)(x^k)$ and $\text{shrink}_v(\cdot)$ is non-expansive [2], using H2, (17), and (18), we have that

$$\begin{aligned} \|x^{k+1} - x^*\|_2 &= \|x_E^{k+1} - x_E^*\|_2 \leq \|h_E(x^k) - h_E(x^*)\|_2 \\ &\leq \|I - \tau_E^k \bar{G}_{EE}^k\| \|x_E^k - x_E^*\| \leq \left\| I - \frac{g_E^{kT} g_E^k}{g_E^{kT} G_{EE}^k g_E^k} \bar{G}_{EE}^k \right\| \|x_E^k - x_E^*\| \\ &\leq \frac{\|g_E^{kT} G_{EE}^k g_E^k - g_E^{kT} \bar{G}_{EE}^k g_E^k\|}{g_E^{kT} G_{EE}^k g_E^k} \|x_E^k - x_E^*\| \\ &\leq \frac{\|g_E^{kT} (G_{EE}^k - \bar{G}_{EE}^k) g_E^k\|}{g_E^{kT} G_{EE}^k g_E^k} \|x_E^k - x_E^*\| \\ &\leq \|g_E^k\|_2^2 \frac{\|G_{EE}^k - \bar{G}_{EE}^k\|}{M^2 \|g_E^k\|_2^2} \|x_E^k - x_E^*\| = \frac{\|G_{EE}^k - \bar{G}_{EE}^k\|}{M^2} \|x^k - x^*\| \\ &= \frac{O(\|x^k - x^*\|)}{M^2} \|x^k - x^*\| \\ &= \frac{O(\|x^k - x^*\|^2)}{M^2}. \end{aligned}$$

Secondly, we suppose that $\tau^k \leq \tau^*$. Similar first case, we conclude that

$$\|x^{k+1} - x^*\| \leq O(\|x^k - x^*\|^2).$$

Theorem 3.7. Suppose that (H1)-(H3) holds and let $\{x^k\}$ is the sequence generated by the FP algorithm starting $x^0 \in \Omega$. For sufficiently large k , the sequence $\{x^k\}$ is converges to some point in $X^* \cap \Omega$ quadratically.

4. Preliminary Numerical Experiments:

This section reports some numerical results and comparisons regarding the implementations of the new proposed idea of the present study with some other algorithms for small-scale problems. All codes are written in

MATLAB 9 programming environment with double precision format by a same subroutine. In the experiments, the presented algorithms are stopped whenever

Test problems are as follows:

1) Tridiagonal system linear is

$$F(x) = Ax,$$

in where A is a $n \times n$ tridiagonal matrix given by

$$A = \text{tridiag}(-1, 8, -1),$$

and

$$A(:, i) = \frac{A(:, i)}{\|A(:, i)\|}, \quad \forall i : i = 1, \dots, n$$

2) Five diagonal system linear is

$$F(x) = Ax,$$

in where A is a $n \times n$ five matrix given by

$$A = \text{tridiag}(1, -1, 8, -1, 1)$$

and

$$A(:, i) = \frac{A(:, i)}{\|A(:, i)\|}, \quad \forall i : i = 1, \dots, n$$

3) Logarithmic function [8]

$$F_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \quad i = 1, \dots, n,$$

initial point: $x^0 = (1, \dots, 1)$.

4) Strictly convex function 1 [8] $f(x)$ that is the gra-

dient of $h(x) = \sum_{i=1}^n e^{x_i} - x_i$

$$F_i(x) = e^{x_i} - 1, \quad i = 1, \dots, n,$$

initial point: $x^0 = \left(\frac{1}{n}, \frac{2}{n}, \dots, \frac{n}{n}\right)$.

5) Strictly convex function 2 [8] $f(x)$ that is the gra-

dient of $h(x) = \sum_{i=1}^n \frac{i}{10} (e^{x_i} - x_i)$

$$F_i(x) = \frac{i}{10} (e^{x_i} - 1), \quad i = 1, \dots, n$$

initial point: $x^0 = (1, \dots, 1)$.

6) Strictly convex function 3 [8] $f(x)$ that is the gra-

dient of $h(x) = \sum_{i=1}^n \frac{i}{10} \left(e^{x_i} - \frac{x_i}{i} \right)$

$$F_i(x) = \frac{i}{10} \left(e^{x_i} - \frac{1}{i} \right), \quad i = 1, \dots, n,$$

initial point: $x^0 = (1, \dots, 1)$.

7) Linear function-full rank [8]

$$F_i(x) = x_i - \frac{2}{n} \sum_{j=1}^n x_j, \quad i = 1, \dots, n$$

initial point: $x^0 = (100, \dots, 100)$.

8) Penalty function [8]

$$F_i(x) = \sqrt{10^{-5}} (x_i - 1), \quad i = 1, \dots, n-1,$$

$$F_n(x) = \frac{1}{4n} \sum_{j=1}^n x_j^2 - \frac{1}{4},$$

initial point: $x^0 = \left(\frac{1}{3}, \dots, \frac{1}{3}\right)$.

9) Sum square function [9]

$$F_i(x) = \sqrt{i} x_i, \quad i = 1, \dots, n,$$

initial point: $x^0 = (1, \dots, 1)$.

10) Trigonometric exponential function [10]

$$F_1(x) = x_1 - \exp(\cos(h(x_1 + x_2))),$$

$$F_i(x) = x_i - \exp(\cos(h(x_{i-1} + x_i + x_{i+1}))),$$

$$i = 2, \dots, n-1, \quad F_n(x) = x_n - \exp(\cos(h(x_{n-1} + x_n))),$$

where $h = h/n + 1$.

initial point: $x^0 = (1.5, \dots, 1.5)$.

In this section, we compare the numerical results obtained by running Algorithms following:

1) FP1 ($\tau^k = 1$)

$$2) \text{ FP2 } \left(\tau^k = \frac{g^{kT} g^k}{(J^k g^k)^k (J^k g^k)} \right)$$

3) DS (steepest descent direction,

$$d_k = - \frac{g^{kT} g^k}{(J^k g^k)^T (J^k g^k)} g^k.$$

in where the Jacobian matrix J^k is exact. In running the algorithm FP1 and FP2 takes advantages of the parameters $\tau^0 = 1, \mu^0 = 1/\|g^0\|_\infty, \nu^0 = 0.005\|g^0\|_\infty, \eta = 10$ and $k_{\max} = 1000$. The dimensions of problems are selected from 2 to 100. The results for small-scale problems are summarized in **Table 1**.

In **Table 1**, n_i and n_f respectively indicate the total number of iterates and the total number of function evaluations. **Table 1** indicates the total number of iterations and function evaluations for some small scale problems with dimensions 2 to 100. Evidently, one can see that FP2 performs better than the other presented algorithms in the sense of both the total number of iterations and the total number of function evaluations. From **Table 1**, we ob-

serve that the proposed algorithm is the best one on the all of test problems. We can deduce that our new algorithm is more efficient and robust than the other considered algorithms for solving small scale system of convex equations problems. In more details, the results of **Table 1** in **Figure 1** are interpreted thanks to the Dolan and More's performance profile in [11].

In the procedure of Dolan and More, the profile of each code is measured considering the ratio of its computational outcome versus the best numerical outcome of all codes. This profile offers a tool for comparing the performance of iterative processes in statistical structure. In particular, let S is set of all algorithms and P is a set of

Table 1. Numerical results for small scale problems.

Problem	Dim	FP1	SD	FP2
		n_i/n_f	n_i/n_f	n_i/n_f
1	5	15/16	16/17	2/3
	10	17/18	16/17	2/3
	50	19/20	17/18	2/3
2	10	2/3	14/15	2/3
	15	2/3	15/16	2/3
	50	2/3	23/24	2/3
3	5	17/18	5/6	5/6
	10	12/13	5/6	4/5
	50	8/9	5/6	4/5
4	5	32/33	4/5	8/9
	10	32/33	4/5	8/9
	50	32/33	4/5	8/9
5	5	803/804	55/56	3/4
	7	797/798	132/138	3/4
	10	771/772	228/229	3/4
6	2	844/845	5/6	7/8
	3	865/866	5/6	7/8
	2	844/845	5/6	7/8
7	5	9/10	1/2	9/10
	10	9/10	1/2	9/10
	50	9/10	1/2	9/10
8	5	240/241	5/6	5/6
	10	484/485	5/6	5/6
	20	972/973	5/6	5/6
9	5	Over 1000	29/30	16/18
	10	Over 1000	59/60	31/32
	50	Over 1000	288/289	146/147
10	30	7/8	7/8	8/9
	50	7/8	5/6	7/8
	100	8/9	4/5	8/9

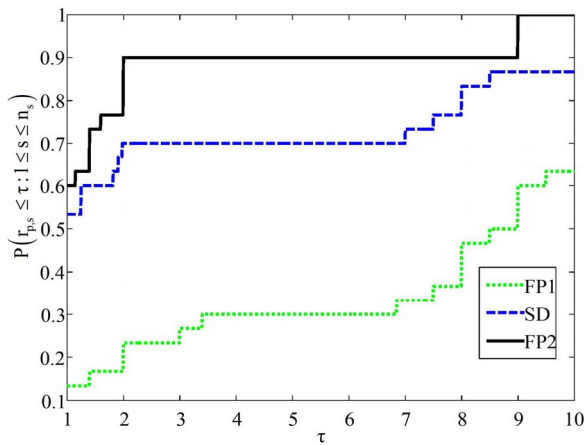


Figure 1. Performance profile for the number of iterates.

test problems, with n_s solvers and n_p problems. For each problem p and solver s and $t_{p,s}$ is the computation result regarding to the performance index. Then, the following performance ratio is defined

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}} \quad (19)$$

If algorithm S is not convergent for a problem p , the procedure sets $r_{p,s} = r_{\text{fail}}$, where r_{fail} should be strictly larger than any performance ratio (19). For any factor τ , the overall performance of algorithm S is given by

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\}.$$

In fact $\rho_s(\tau)$ is the probability of algorithm $s \in S$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in R$ of the best possible ratio. The function $\rho_s(\tau)$ is the distribution function for the performance ratio. Especially, $\rho_s(1)$ gives the probability that algorithm s wins over all other algorithms, and $\lim_{\tau \rightarrow r_{p,s}} \rho_s(\tau)$ gives the probability of that algorithm s solve a problem. Therefore, this performance profile can be considered as a measure of the efficiency and the robustness among the algorithms. In **Figure 1**, the x-axis shows the number τ while the y-axis inhibits

$$P(r_{p,s} \leq \tau : 1 \leq s \leq n_s).$$

From **Figure 1**, it is clear that FP2 had the most wins compared with the other algorithm while it solved about 60% of the test problems with the greatest efficiency. If one concentrates on the ability of completing a run successfully, it can be seen that FP2 is the best algorithm among the considered algorithms because it reaches faster than the other.

5. Conclusion

In this paper, we have presented a new algorithm for

small-scale systems of convex equations that blending steepest descent direction and fixed point ideas. Preliminary numerical effort on the set of small-scale convex systems of equations indicates that significant profits in both the total number of iterations and the total number of function evaluations can be achieved.

REFERENCES

- [1] Z. Wen, W. Yin, D. Goldfarb and Y. Zhang, "A Fast Algorithm for Sparse Reconstruction Based on Shrinkage," Optimization, and Continuation, CAAM Technical Report TR09-01, 2009.
- [2] E. T. Hale, W. Yin and Y. Zhang, "A Fixed-Point Continuation Method for ℓ_1 -Regularized Minimization with Applications to Compressed Sensing," CAAM Technical Report TR07-07, 7 July 2007.
- [3] A. Chambolle, R. A. De Vore, N. Y. Lee and B. J. Lucier, "Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal through Wavelet Shrinkage," *IEEE Transactions on Image Processing*, Vol. 7, No. 3, 1998, pp. 319-335. doi:10.1109/83.661182
- [4] M. A. T. Figueiredo and R. D. Nowa, "An EM Algorithm for Wavelet-Based Image Restoration," *IEEE Transactions on Image Processing*, Vol. 12, No. 8, 2003, pp. 906-916. doi:10.1109/TIP.2003.814255
- [5] I. Daubechies, M. Defrise and C. D. Mol, "An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint," *Communications on Pure and Applied Mathematics*, Vol. 57, No. 11, 2004, pp. 1413-1457. doi:10.1002/cpa.20042
- [6] C. Vonesch and M. Unser, "Fast Iterative Thresholding N. Algorithm for Wavelet-Regularized Deconvolution," *Proceedings of the SPIE Optics and Photonics 2007 Conference on Mathematical Methods: Wavelet XII*, Vol. 6701, San Diego, 26-29 August 2007, p. 15.
- [7] B. Martinet, "Régularisation d'Inéquations Variation Nelles par Approximations Successives," *Recherche Operationnelle*, Vol. 4, No. 3, 1970, pp. 154-158
- [8] G. L. Yuan, Z. X. Wei and S. Lu, "Limited Memory BFGS Method with Backtracking for Symmetric Non-linear Equations," *Mathematical and Computer Modelling*, Vol. 54, No. 1-2, 2011, pp. 367-377. doi:10.1016/j.mcm.2011.02.021
- [9] M. Moga and C. Smutnicki, "Test Functions for Optimization Need," 2005. www.zsd.itc.pwr.pc/file/docs/function.pdf.
- [10] L. Lukšan and J. Vlček, "Sparse and Partially Separable Test Problems for Unconstrained and Equality Constrained Optimization," *Technical Report*, Vol. 767, 1999.
- [11] E. D. Dolan and J. J. Moré, "Benchmarking Optimization Software with Performance Profiles," *Mathematical Programming*, Vol. 91, No. 2, 2002, pp. 201-203.