This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier’s archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright
A projected semismooth Newton method for problems of calibrating least squares covariance matrix

Qingna Li a, Donghui Li b,∗

a Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing 100080, China
b School of Mathematical Sciences, South China Normal University, Guangzhou, 510631, China

1. Introduction

In this paper, we consider the following calibrating least squares covariance matrix problem

$$\min_{X \in \mathbb{R}^{n \times p}} \frac{1}{2} \|X - G\|^2$$

s.t. $$\langle A_i, X \rangle = b_i, \quad i \in \mathcal{P} = \{1, \ldots, p\},$$

$$\langle A_i, X \rangle \geq b_i, \quad i \in \mathcal{Q} = \{p + 1, \ldots, p + q\},$$

$$X \succeq 0$$

where $$G, A_i \in \mathbb{R}^n, i = 1, \ldots, p + q$$, are given matrices, $$b = (b_1, \ldots, b_{p+q})^T \in \mathbb{R}^{p+q}$$ is a given vector, $$\mathbb{R}^n$$ denotes the space of $$n \times n$$ symmetric matrices and $$\langle \cdot, \cdot \rangle$$ denotes the Frobenius norm of matrices induced by the standard inner product $$(\cdot, \cdot)$$. Here and throughout, we use $$X \succeq 0$$ to mean that $$X$$ is positive semidefinite. We will also use $$X > 0$$ to mean that $$X$$ is positive definite.

Problem (1) has many applications in finance, insurance and reinsurance. From the mathematical point of view, a natural way to deal with it is to solve its dual problem. Denote $$A : \mathbb{R}^n \rightarrow \mathbb{R}^{p+q}$$ by $$A(X) = \{A_1, X\}, \ldots, (A_{p+q}, X)\}^T \in \mathbb{R}^{p+q}$$ with $$X \in \mathbb{R}^n$$. The dual problem of (1) can be reformulated as

$$\min_{Y \in \mathbb{R}^{p+q}} \frac{1}{2} \|P_{S_+}(A^*y + G)\|^2 - b^TY$$

s.t. $$y_i \geq 0, \quad i \in \mathcal{Q},$$

where $$P_{S_+}(\cdot)$$ is the metric projection operator onto the positive semidefinite matrix cone $$S_+$$, $$A^*$$ is the adjoint operator of $$A$$ defined by $$A^*y = \sum_{i=1}^{p+q} YA_i$$ with $$y \in \mathbb{R}^{p+q}$$. We refer to [12] for details about the dual relation between (1) and (2).

The dual problem (2) has some interesting properties such as continuous differentiability of the function $$\theta(y)$$ with

$$\nabla \theta(y) = A(P_{S_+}(A^*y + G)) - b,$$

and the strong semismoothness of the projection operator $$P_{S_+}(\cdot)$$ and $$A$$ [see [18]].

If $$\mathcal{Q} = \emptyset$$, the dual problem (2) reduces to an unconstrained optimization problem. A typical case is the nearest correlation matrix problem

$$\min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2} \|X - G\|^2$$

s.t. $$X_{ii} = 1, \quad i = 1, \ldots, n,$$

$$X \succeq 0$$

with the corresponding dual problem

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \|P_{S_+}(\text{Diag}(y) + G)\|^2 - e^Ty$$

where $$e = (1, \ldots, 1)^T \in \mathbb{R}^n$$, Diag(y) is the diagonal matrix formed by the vector $$y \in \mathbb{R}^n$$. Qi and Sun [14] proposed a quadratically convergent semismooth Newton method to solve (4), where the generalized Hessian, $$\partial^2 \theta$$, of $$\theta$$ was not computed directly but replaced by some larger set

$$\tilde{A}^2 \theta(y) \triangleq \partial \partial \Pi_{S_+}(A^*y + G)A^*$$

where$$\Pi_{S_+}(\cdot)$$ is the metric projection operator onto the positive semidefinite matrix cone $$S_+$$, $$\tilde{A}^*$$ is the adjoint operator of $$\tilde{A}$$ defined by $$\tilde{A}^*y = \sum_{i=1}^{p+q} Y \tilde{A}_i$$ with $$y \in \mathbb{R}^{p+q}$$. We refer to [12] for details about the dual relation between (1) and (2).
with $A(X) \colon= \text{diag}(X)$, the vector formed by all diagonal elements of $X$, and $A^\top y = \text{Diag}(y)$. The numerical results given in [14] showed that the semismooth Newton method outperforms the other existing methods [5,10,12]. This method in [14] can be, at least conceptually, extended to solve (1) with a general operator $A$ in the absence of inequality constraints.

In the case of $Q \neq \emptyset$, problem (2) is a constrained optimization problem, which is not that easy to tackle. The difficulty lies in the presence of the inequality constraints $y_i \geq 0, i = p + 1, \ldots, p + q$. In order to handle the inequality constraints, Boyd and Xiao [5] proposed a projected gradient method and proved its linear convergence. In [9], Gao and Sun developed a smoothing Newton method that is quadratically convergent and is numerically very efficient. Quite recently, Chen, Gao and Liu [7] proposed an inexact SQP method for convex $SC^1$ minimization problems including (2) as a special case.

Keeping the fast convergence rate of the semismooth Newton method [14] in mind, a natural question arises: for $Q \neq \emptyset$, can we still apply the semismooth Newton method to solve (1) and expect a fast convergence rate? This paper will give us an affirmative answer to it by introducing a projected semismooth Newton method.

The first key point to achieve our goal is to find a proper way to deal with the inequality constraints in (2). Actually, (2) is an optimization problem with simple constraints, which has already been well studied in nonlinear optimization. Most of those methods fall into two categories: the active set methods and trust region methods, see [2,8] for example. Due to the nondifferentiability of $\nabla \theta$, gradient methods, which only use the first order information, can be applied directly to solve (2). However, the convergence rate is at best linear, simply because no second order information is included. For second order methods, they cannot be applied to (2) without modifications because of the nondifferentiability of $\nabla \theta$. Fortunately, these modifications are not impossible because the semismooth Newton method [14] provides a way to deal with the nondifferentiability of $\nabla \theta$. One of the second order methods is Bertsekas’ projected Newton method [2], which is not only simple to implement but also one of the efficient active set methods. We choose to follow the frame of his method because of two reasons. First, we can follow his way to deal with the inequality constraints. Second, the use of the Newton step for those free variables in the method ensures the superlinear convergence. With the strong semismoothness of $\nabla \theta$ and the constraint nondegeneracy at the primal optimal solution, which guarantees the nonsingularity of $\partial^2 \theta$. The connection between constraint nondegeneracy and the nonsingularity of $\partial^2 \theta$ has been investigated in [6,17,19], which is also valid for our problem. With such connection as well as the strict complementarity condition, we can keep the quadratic convergence of the proposed method.

The main contribution of our paper is that we modify Bertsekas’ projected Newton method to develop a projected semismooth Newton method. At each step, we still estimate the active set and make projection to guarantee the feasibility of iterative points. The main difference between the proposed method and Bertsekas’ method lies in the Newton step. Specifically, we use the semismooth Newton method for those free variables. Furthermore, we will borrow the stopping criterion from the QP-free constrained Newton-type method by Kanzow and Q [11] to terminate our method efficiently.

The paper is organized as follows. In Section 2, we introduce the main idea of the projected semismooth Newton method and give some properties. In Section 3, we analyze the convergence property of the proposed method. We establish the global convergence with general assumptions, and prove the quadratic convergence under constraint nondegeneracy and strict complementarity condition. In Section 4, we implement the algorithm and compare it with the smoothing Newton method by Gao and Sun [9]. The numerical results confirm the efficiency of our method. It is comparable with the smoothing Newton method.

Some words about notation: capital letters stand for matrices, small letters for vectors and Greek letters for scales. We use ‘$\otimes$’ to denote the Hardmard product, and ‘$\otimes$’ the Kronecker product. For subsets $\alpha$, $\beta$ of $\{1, \ldots, n\}$, denote $B_{\alpha, \beta}$ as the submatrix of $B$ indexed by $\alpha$ and $\beta$. Similarly, $y_\alpha$ is the subvector of vector $y$ indexed by $\alpha$. $|\alpha|$ denotes the cardinality of $\alpha$, $E$ is the matrix of all ones. $I$ is the identity matrix.

The idea of the projected Newton method in [2] is that, at each iteration, $y^{k+1}$ is updated by

$$y^{k+1} = [y^k + \alpha_k d^k]^+$$

In this section, we will present the algorithm of the projected semismooth Newton method and investigate its useful properties. The main differences between our method and the existing projected Newton method is the treatment for free variables. In addition, an efficient stopping criterion from [11] will be used in our method.

We do some preliminaries first.

Let $g(y) = \nabla \theta(y)$. We say $y$ is a stationary point of (2) if it satisfies

$$\begin{cases} g_\alpha(y) &= 0, \\ y_i &\geq 0, \quad g_i \geq 0, \quad y_i g_i(y) = 0, \quad i \in Q. \end{cases}$$

Suppose $y^*$ is a stationary point of (2). Then $X^* := \Pi_{S^*} (A^* y^* + G)$ is an optimal solution of (1). The system (7) implies that $X^*$ and $y^*$ satisfy the following mixed complementarity condition:

$$\begin{cases} (A_i, X^*) - b_i = 0, \quad i \in S, \\ (A_i, X^*) - b_i \geq 0, \quad y_i (A_i, X^*) - b_i = 0, \quad i \in Q. \end{cases}$$

Define the active set at $y^*$ as

$$I(y^*) = \{ i \in Q \mid y_i^* = 0 \}$$

and the active set at $X^*$ as

$$I(X^*) = \{ i \in Q \mid (A_i, X^*) = b_i \}.$$
where $\alpha_k$ is the step length, $d_k$ is determined by $d_k = -D_k g(y_k)$ ($y_k$), $D_k$ is a positive definite matrix and partly diagonal. It is mentioned in [2] that $D_k$ can be calculated on the basis of the second derivatives of $\theta$ (provided that $\theta$ is twice continuously differentiable) so that the resulting Newton-like method has a typically superlinear rate of convergence. Inspired by this idea and the properties of $\theta$, we can choose $D_k$ in a way so that those variables which are estimated to be free, can be updated by the semismooth Newton method. Specifically, the algorithm is stated as follows.

**Algorithm 2.1** (Projected semismooth Newton Method).

Step 0. Initial point: $y_0 \in \mathbb{R}^{p+q}$ with $y_0^T > 0$. Tolerance: $\text{Tol} > 0$. Parameters: $0 < \rho < 1, 0 < \sigma < 1/2, \epsilon > 0, 0 < \eta < 1$. Iteration: $k := 0$.

Step 1. Calculate the estimated active set at $y_k$ as
\[
l_k = \{i \mid 0 \leq y_k^i \leq \epsilon_k, g_i(y_k) > 0, i \in Q\}
\]
where $\epsilon_k = \min\{\epsilon, \alpha_k\}$, $\alpha_k = \|y_k^i - [y_k - M(y_k)]\|$.

$M$ is a fixed diagonal positive definite matrix in $\mathbb{R}^n$. Let $\tilde{l}_k := (\partial \cup Q) \setminus l_k$.

Step 2. Calculate the residual $h \in \mathbb{R}^{p+q}$ by
\[
h_k := \begin{bmatrix} h_{k}^T \\ h_{\tilde{l}_k}^T \end{bmatrix}
\]
with $h_{k}^T = \min \{y_k^i, g_i(y_k)\}$ and $h_{\tilde{l}_k} = g_{\tilde{l}_k}(y_k)$. If $\|h_k\| \leq \text{Tol}$, stop. Otherwise, go to Step 3.

Step 3. Calculate $d_k$ by
\[
d_k = \begin{bmatrix} d_{k}^T \\ d_{\tilde{l}_k}^T \end{bmatrix}
\]
where
\[
d_{k}^T = -(2^{\beta_k})^{-1} y_{\tilde{l}_k}
\]
with $2^{\beta_k} \in \mathbb{R}^{\|k\|}$ diagonal, positive definite, and $d_{\tilde{l}_k}^T$ is determined by
\[
(V_{l_k}^k + \delta_k d_k)^T d_{\tilde{l}_k} = -g_{l_k}(y_k)
\]
such that
\[
\|g_{l_k}(y_k) + V_{l_k}^k d_{\tilde{l}_k}\| \leq \alpha_k \|g_{l_k}(y_k)\|,
\]
where $V_{l_k}^k \in \bar{\Sigma}^2(y_k)$, $\delta_k = \min\{\|\tilde{g}_{l_k}(y_k)\|^2\}$, $\tilde{g}_{l_k}(y_k)$ is the identity matrix in $\mathbb{R}^{\|k\|}$. If (14) is not satisfied or if the condition
\[
g_{l_k}(y_k)^T d_{\tilde{l}_k} \leq -\alpha_k \|d_{\tilde{l}_k}\|^2
\]
fails, let $d_{\tilde{l}_k} = -(2^{\beta_k})^{-1} \tilde{g}_{l_k}(y_k)$ where $\tilde{g}_{l_k}$ is any symmetric positive definite matrix in $\mathbb{R}^{\|k\|}$.

Step 4. Choose $\alpha_k = \beta_m$ where $m_k$ is the first nonnegative integer $m$ such that
\[
\theta((y_k^i + \beta_m d_k^i)^{+}) - \theta(y_k^i) \leq \sigma (\beta_m \sum_{i \in l_k} g_i(y_k^i)) d_k^i + \sum_{i \in l_k} g_i(y_k^i)(y_k^i + \beta_m d_k^i)^{+} - y_k^i,
\]
where $\theta((y_k^i + \beta_m d_k^i)^{+}) = \max(0, (y_k^i + \beta_m d_k^i))$

Step 5. Update $y_k+1$ by (10). $k := k + 1$, go to Step 1.

**Remark 1.** The differences between Algorithm 2.1 and the projected Newton method [2] are in three aspects.

(i) $d_{k}^T$ is updated by (12) instead of $d_{k}^T = -(2^{\beta_k})^{-1} g_{l_k}(y_k)$ because the choice (12) leads to a better numerical performance.

(ii) $d_{k}^T$ is determined by the semismooth Newton method (13). This is what we mentioned in Introduction. We will see in Section 3 that (13) plays an important role in convergence rate analysis.

(iii) The stopping criterion in Step 2 comes from Algorithm 3.1 in [11]. It is also efficient in Algorithm 2.1. With (7) and the definition of $h_k$, it is easy to get that $h_k = 0$ if and only if $y_k$ is a stationary point of (2).

**Remark 2.** Despite the differences we listed in Remark 1, we can still show, in a similar way to that in [2], that the line search scheme (15) is well defined. The right hand side of (15) is nonpositive, and is negative if and only if $y_k$ is not a stationary point of (2). To save space, we omit the proof here.

We end this section by showing how to calculate the submatrix $V_{l_k}^k$. To that end, we need some notations for the spectral decomposition. Let $X \in \mathbb{R}^n$ have the following spectral decomposition:
\[
X = P \text{Diag}(\lambda_1, \ldots, \lambda_n) P^T,
\]
where $P^T P = I_1 \leq \lambda_i \leq \cdots \leq \lambda_n$ are eigenvalues of $X$. Define $\alpha, \beta, \gamma$ as follows
\[
\alpha = \{i \mid \lambda_i > 0\}, \quad \beta = \{i \mid \lambda_i = 0\} \quad \text{and} \quad \gamma = \{i \mid \lambda_i < 0\}.
\]
Accordingly, $P$ can be written as $P = [P_\alpha P_\beta P_\gamma]$. In general, it is difficult to compute the generalized Hessian $\tilde{\beta}^2 \theta$ exactly. We define the following alternative for $\tilde{\beta}^2 \theta$
\[
\hat{\beta}^2 \theta(y) = A \delta P_{\beta} (A^* y + G) A^*.
\]

**Lemma 2.2.** Given $y \in \mathbb{R}^{p+q}$, suppose $A^* y + G$ has the spectral decomposition of (16) with $X := A^* y + G$. Similar to Lemma 3.5 in [14], let $V \in \bar{\Sigma}^2(y)$ be defined by
\[
V_{h} = A(\lambda (\Omega (P \ominus P \text{Diag}(\Omega(Y))) P^T), \quad h \in \mathbb{R}^{p+q}.
\]

Then for any $I \subset \{1, \ldots, p + q\}$, there is $V_{h_{l_k}I} = A_{l_k} (\lambda (P \ominus P \text{Diag}(\Omega(Y))) P^T) I_{l_k}$ where $A_{l_k} (X) = (X)_{l_k} \in \mathbb{R}^{\|I\|}$. $\lambda (\Omega (P \ominus P \text{Diag}(\Omega(Y))) P^T) I_{l_k}$ is the matrix $V_{h_{l_k}I}$ can be calculated as
\[
V_{h_{l_k}I} = (P \ominus P \text{Diag}(\Omega(Y))) P^T I_{l_k} = A(l_{k} :) (P \ominus P \text{Diag}(\Omega(Y))) P^T I_{l_k}
\]
where $A(l_{k} :)$ denotes the rows of $A$ indexed by subset $I$ and $A(l_{k} :)$ denotes the columns of $A^T$ indexed by $I$. Consequently, for $h \in \mathbb{R}^{p+q}$, there is
\[
V_{h_{l_k}I} = A(l_{k} :) (P \ominus P \text{Diag}(\Omega(Y))) P^T I_{l_k} = A(l_{k} :) (P \ominus P \text{Diag}(\Omega(Y))) P^T I_{l_k}.
\]

We finish the proof. $\square$

3. Convergence analysis

In this section, we will analyze the convergence properties of Algorithm 2.1. We will show that the proposed method is globally convergent with general assumptions. Furthermore,
under constraint nondegeneracy and strict complementarity condition, the proposed method converges quadratically. Without loss of generality, denote \( d_k = -D_k \begin{bmatrix} \frac{1}{g_k} & 0 \\ 0 & \frac{1}{k_0} \end{bmatrix} \) with \( D_k = \begin{bmatrix} (z_k)^2 & 0 \\ 0 & k_0^{-1} \end{bmatrix} \). In the latter part of the paper, without specification, we always suppose that the following assumption holds.

**Assumption 1.** There exist positive scalars \( \xi_1, \xi_2 \) such that \( \xi_1 \|z\|^2 \leq z^T D^2 z \leq \xi_2 \|z\|^2 \), \( \forall z \in \mathbb{R}^{p+n}, k = 0, 1, \ldots \).

The following theorem can be proved in a similar way (with slight modifications) to the proof of Proposition 2 in [2].

**Theorem 3.1.** Suppose Assumption 1 holds, then every limit point of sequence \( \{y^k\} \) generated by Algorithm 2.1 is a stationary point of (2).

To study the convergence rate of Algorithm 2.1, we need the concept of constraint nondegeneracy. Constraint nondegeneracy was introduced by Bonnans and Shapiro [3] for general optimization problems. It is the generalization of the well-known linear independence constraint qualification (LICQ). For more details of constraint nondegeneracy, see [1, 3]. The following lemma is an extension of Proposition 3.2 in [19], which establishes the equivalence between constraint nondegeneracy at the primal optimal solution and the positive definiteness of the corresponding submatrix of \( \tilde{A}^T \tilde{\theta} \). It can also be derived from [6, 17].

**Lemma 3.2.** Suppose \( y^* \) is the optimal solution of (2). Then constraint nondegeneracy holds at \( X^* := \Pi_S (A^{-1} y^* + G) \) if and only if for any \( V \in \tilde{A}^T \tilde{\theta} (y^*) \), \( V \) is positive definite in the subspace \( H = \{ z \in \mathbb{R}^{p+n} | z_i = 0, i \in \mathbb{O} \setminus \{I(X^*)\} \} \). That is,

\[
\text{Constraint nondegeneracy holds at } X^* \iff V_{i | I(X^*)} > 0
\]

where \( J^* = \mathbb{P} \cup I(X^*) \). \( I(X^*) \) is defined as in (9).

With Lemma 3.2, we are ready to state the local convergence result.

**Theorem 3.3.** Let \( y^* \) be a limit point of \( \{y^k\} \) generated by Algorithm 2.1. Suppose constraint nondegeneracy holds at \( X^* := \Pi_S (A^{-1} y^* + G) \) and the strict complementarity condition holds at \( y^* \). Then \( \{y^k\} \) converges to \( y^* \) quadratically.

**Proof.** With the strict complementarity condition, Theorem 3.1 implies that there exists a positive integer \( k_0 \) such that for any \( k \geq k_0 \), \( l_k = l(y^*) \). It means that after finite iterations, Algorithm 2.1 is equivalent to an unconstrained optimization method restricted on the subspace \( H \), where \( H \) is defined as in Lemma 3.2. With Theorem 2.1 in [14] and Lemma 3.2, we get that \( \{y^k\} \) converges to \( y^* \) quadratically.

**Remark 3.** The strict complementarity condition is assumed in the original projected Newton method [2] to guarantee the quadratic convergence rate. In our method, we still need it in order to achieve a quadratic convergence rate.

4. Numerical results

In this section, we will compare our algorithm (denoted as PSpminiNewton) with the inexact smoothing Newton method in [9] (denoted as CaliMat), which is available from [http://www.math.nus.edu.sg/~matundf]. We take the following special case as our test problem

\[
\begin{align*}
\min & \quad \frac{1}{2} \|X - G\|^2 \\
\text{s.t.} & \quad X_{ij} = 1, \quad (i,j) \in E_1, \\
& \quad X_{ij} \geq l_{ij}, \quad (i,j) \in E_2, \\
& \quad X_{ij} \leq u_{ij}, \quad (i,j) \in E_3, \\
& \quad X \succeq 0,
\end{align*}
\]

where \( E_1, E_2, E_3 \) are three index subsets of \( \{(i,j) | 1 \leq i \leq j \leq n\} \) satisfying: \( E_1 \cap E_2 = \emptyset, E_1 \cap E_3 = \emptyset, l_{ij} < u_{ij}, \forall (i,j) \in E_1 \cap E_3 \). Denote the cardinalities of \( E_1, E_2, E_3 \) by \( p, q_1, q_2 \) respectively. Let \( m := p + q_1 + q_2 \). Let \( e_{1i} \) be the \( l \)-th column of the identity matrix \( I \in \mathbb{R}^n \). Denote \( A^T = \frac{1}{2}(e_{1i}^T + e_{1j}^T) \), then (20) can be written as a special case of (1) with

\[
\begin{align*}
A(X) := & \begin{pmatrix} \{ [A^T, X]_{(i,j) \in E_1} \} & \{ [A^T, X]_{(i,j) \in E_2} \} \end{pmatrix}, \quad X \in \mathbb{R}^{m} \\
& \begin{pmatrix} \{ [A^T, X]_{(i,j) \in E_3} \} \end{pmatrix}
\end{align*}
\]

To see when constraint nondegeneracy holds with respect to the constraints in (20), we refer the reader to [14–16].

(a) Implementations of Algorithm 2.1

(i) We use the diagonally preconditioned MINRES [13] to solve (13). The reason is, in the implemented version of the semismooth Newton method [4], Borsdorf and Higham showed that the preconditioned MINRES can lead to the better numerical performance compared to the preconditioned CG used in [14].

(ii) \( V_0^k \in \mathbb{R}^l \), \( l \leq n \), are chosen as the diagonal elements in \( Z^0 \) in order to include as much information of the generalized Hessian as possible.

From (i) and (ii), we need to calculate the diagonal elements in \( V \in \tilde{A}^T \tilde{\theta} (y^*) \), i.e., \( V_0, l = 1, \ldots, m \). The calculation is similar to that in [9, Section 5]. Note that

\[
\begin{align*}
V_0 & = \left( \tilde{h}_i, V_{i \Omega} \right) \\
& = \left( \tilde{h}_i, A(P(\Omega \circ (P^T (A^* \hat{\theta}))P^T))) \right) \\
& = \langle P^T (A^* \hat{\theta})P, \Omega \circ (P^T (A^* \hat{\theta})P) \rangle
\end{align*}
\]

and

\[
A_t = \begin{pmatrix} \frac{1}{2} (e_{1i}e_{1i}^T + e_{1j}e_{1j}^T) & 0 \\ 0 & \frac{1}{2} (e_{1i}e_{1i}^T + e_{1j}e_{1j}^T) \end{pmatrix}, \quad I \in \{p + q_1 + 1, \ldots, m\}.
\]

where \( \hat{\theta} \) is the \( l \)-th column of the identity matrix \( I \in \mathbb{R}^n \) and

\[
(i, j) \in \begin{cases} \Omega, & \text{if } i = 1, \ldots, p, \\
\Omega_2, & \text{if } i = p + 1, \ldots, p + q_1, \\
\Omega_3, & \text{if } i = p + q_1 + 1, \ldots, m.
\end{cases}
\]

Due to the special structure of \( A_t \), there is \( V_{i \Omega} = \frac{1}{2} (a_{i1}^T \Omega (a_{i1})^T + (a_{i2} \circ a_{i2}) \Omega (a_{i2} \circ a_{i2})^T) \) where \( a_i \) is the \( l \)-th row of \( P, a_{i1} = a_i, a_{i2} = a_i, i = 1, \ldots, n \). In this way, the cost of calculating the diagonal matrix of \( V \) is \( O(n^2) \). When \( m = O(n^2) \), it is still too expensive. Therefore, we borrow the idea from [9] to estimate \( V_{i \Omega} \) by \( \hat{V}_{i \Omega} := a_{i1}^T \Omega (a_{i1})^T \), which will reduce the cost to \( O(n^3) \) to calculate the estimated diagonal preconditioner.

(b) Test examples

**Example 1.** \( n = 387 \). The matrix \( G \) is the \( 387 \times 387 \) 1-day correlation matrix (as of October 10, 2008) from the lagged datasets of RiskMetrics (www.riskmetrics.com/stddownload.edu.html). For the test purpose, we perturb \( G \) to \( G := (I - v)G + vR \) where \( v \in (0, 1) \) and \( R \) is a randomly generated symmetric matrix.
with entries in $[-1, 1]$. The Matlab code for generating the random matrix $R$ is: $R = 2.0 * rand(387, 387) - ones(387, 387)$; $R = \text{triu}(R) + \text{triu}(R, 1)$; for $i = 1 : n; R(i,i) = 1; \text{end}$. Here we take $v = 0.1$. $B_i = \{(i, i) \mid i = 1, \ldots, n\}$. The index sets $B_l, B_r$ consist of the indices of $\min(h, n - i)$ randomly generated elements at the $i$-th row of $G$, $i = 1, \ldots, n$, with $h = 1, 5, 10, 20, 50, 100, 150$.

**Example 2.** $G$ is a randomly generated $n \times n$ symmetric matrix with entries in $[-1, 1]$. $B_l = \{(i, i) \mid i = 1, \ldots, n\}$. Given an integer $n_1 \in [1, n]$, we choose $B_l, B_r$ as block inequality constraints, i.e., $B_l = \{(i, j) \mid 1 \leq i \leq n_1, 1 \leq j < i\} \cup \{(i, j) \mid n_1 + 1 \leq i \leq n, 1 \leq j < i\}$. We test cases (a) and (b) as in Example 1 and choose $n_1 = n/10, n/5, n/4, n/2$ with $n = 500, 1000, 1500, 2000, 2500$. Parameters are chosen as: $\alpha = 0, \text{Tol} = 1.0 \times 10^{-6}$, $\rho = 0.5$, $\sigma = 1.0 \times 10^{-4}$, $\epsilon = 0.01$, $\eta = 0.2$, $B_l = I$, $M = I$. We report the number of iterations $\text{Iter}$, the CPU time $t$ in the form of $hh:mm:ss$ and the residual $\text{Res} = \|h^k\|$

Before comparing $\text{PSE}n$W on with CaliMat, we first investigate the role that estimated preconditioner plays in $\text{PSE}n$W on. We report the results on Example 2 where a large number of constraints is involved. From Table 1, it is quite clear that using estimated preconditioner leads to great improvement in CPUtime. Now we compare $\text{PSE}n$W on using estimated preconditioner with CaliMat on Examples 1–3. The results in Tables 2–4 confirm the efficiency of $\text{PSE}n$W on. Such good performance may be explained by the following reasons: (i) The idea of the projected Newton method is effective to deal with simple bound constraints; (ii) The active set is well estimated by $\text{PS}n$W on; (iii) The semismooth Newton method has been proved to be highly efficient in several papers, such as [14,15]. For most cases, CaliMat has a better performance. However, there are cases that $\text{PSE}n$W on is slightly faster. This may be partly explained by the more iterations in CaliMat. In conclusion, $\text{PSE}n$W on is comparable to CaliMat on the tested problems.

**Acknowledgements**

The second author’s research is supported by the major project of the Ministry of Education of China Grant 309023 and the NSF of China Grant 11071087 and 10771057.
### Table 4
Results on Example 3.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n_0$</th>
<th>$a$</th>
<th>PSemiNewton</th>
<th>CaliMat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Iter</td>
<td>$t$</td>
<td>Res</td>
</tr>
<tr>
<td>500</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>1.37E−07</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>4.98E−07</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>6.08E−07</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>9</td>
<td>29</td>
<td>8.49E−08</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>9</td>
<td>39</td>
<td>1.69E−07</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8</td>
<td>38</td>
<td>4.51E−07</td>
</tr>
<tr>
<td>1500</td>
<td>1</td>
<td>9</td>
<td>1:26</td>
<td>7.47E−08</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>9</td>
<td>1:31</td>
<td>9.92E−07</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>9</td>
<td>1:59</td>
<td>5.02E−07</td>
</tr>
<tr>
<td>2000</td>
<td>1</td>
<td>9</td>
<td>3:20</td>
<td>6.28E−08</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>9</td>
<td>3:41</td>
<td>3.93E−07</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>9</td>
<td>4:38</td>
<td>4.44E−07</td>
</tr>
</tbody>
</table>

### References


