ISSN 1346-5597

National Institute of Informatics

NII Technical Report

Modulus-Type Inner Outer Iterative Methods for Nonnegative Constrained Least Squares Problems

Ning Zheng, Ken Hayami, and Jun-Feng Yin

NII-2015-001E
Jan. 2015

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

MODULUS-TYPE INNER OUTER ITERATIVE METHODS FOR NONNEGATIVE CONSTRAINED LEAST SQUARES PROBLEMS*

NING ZHENG†, KEN HAYAMI‡, AND JUN-FENG YIN§

Abstract. For the solution of large sparse nonnegative constrained least squares (NNLS) problems, a new iterative method is proposed by using conjugate gradient least squares (CGLS) method for inner iterations and the modulus-type iterative method in the outer iterations for the solution of linear complementarity problem (LCP) resulting from Karush-Kuhn-Tucker (KKT) conditions of the NNLS problem. Theoretical convergence analysis including the optimal choice of the parameter matrix is presented for the proposed method. Numerical experiments show the efficiency of the proposed method compared to projection-type methods with less iteration steps and CPU time.

Key words. least squares problems, nonnegative constraints, convergence, linear complementarity problem, CGLS method

AMS subject classifications. 65F10, 65F20, 65F22, 65F50

1. Introduction.

Consider the nonnegative constrained linear least squares problem [5], abbreviated as NNLS,

$$\min_{\boldsymbol{x}\in\mathbf{R}^n} \|A\boldsymbol{x}-\boldsymbol{b}\|_2, \quad \text{subject to} \quad \boldsymbol{x}\geq\mathbf{0}, \tag{1.1}$$

where $A\in\mathbf{R}^{m\times n}$, $\boldsymbol{b}\in\mathbf{R}^{m\times 1}$, $m\geq n$ or $m<n$, and the inequalities are to be interpreted componentwise. The rank-deficient case is allowed, when the equality in $\mathrm{rank}A\leq\min(m,n)$ does not hold. Not only do the NNLS problems arise in many scientific computing and engineering applications, e.g., reconstruction problems in geodesy and tomography, contact problems for mechanical systems, and the modeling of ocean circulation, but it is even argued that any minimization problem is only realistic when its variables are constrained within meaningful intervals [4].

Algorithms for the solution of unconstrained linear least squares problem

$$\min_{\boldsymbol{x}\in\mathbf{R}^n} \|A\boldsymbol{x}-\boldsymbol{b}\|_2 \tag{1.2}$$

fall into two classes: direct methods, which are usually based on some matrix factorizations and may not be so practical when matrix A is large and sparse, and iterative methods, among which the (preconditioned) CGLS method, which is mathematically equivalent to the conjugate gradient (CG) method applied to the normal equation

$$A^{\mathrm{T}}A\boldsymbol{x}=A^{\mathrm{T}}\boldsymbol{b}, \quad m\geq n, \tag{1.3}$$

and the CGNE method equivalent to CG applied to the normal equation

$$AA^{\mathrm{T}}\boldsymbol{y}=\boldsymbol{b}, \quad \boldsymbol{x}=A^{\mathrm{T}}\boldsymbol{y}, \quad m\leq n, \tag{1.4}$$

*This work was supported by the Grant-in-Aid for Scientific Research (C) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

†Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies (Sokendai), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan/Department of Mathematics, Tongji University, 1239 Siping Road, Shanghai, 200092, People's Republic of China (nzheng@nii.ac.jp).

‡National Institute of Informatics/Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies (Sokendai), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan (hayami@nii.ac.jp).

§Department of Mathematics, Tongji University, 1239 Siping Road, Shanghai, 200092, People's Republic of China (yinjf@tongji.edu.cn).

and (preconditioned) GMRES type methods by Hayami, Yin and Morikuni [11, 12] play important roles. However, the approximate solutions determined by the above methods are not guaranteed to satisfy the nonnegative constraints in (1.1). Therefore, special techniques must be added to the algorithms that handles the status of variables with respect to their nonnegativity.

A class of inner outer iterative methods is widely discussed for the solution of NNLS problems, where a series of unconstrained least squares problems are solved in the inner iteration, and the obtained solution is updated to satisfy the nonnegative constraints, and then the outer iteration is restarted until convergence. Remark that the inner outer iteration methods contain two tasks including how to update the solution of the unconstrained least squares problem when some of its components violate the bounds, and when to terminate the inner iteration and restart the outer iteration. For example, by restricting the step size in each CG iteration to satisfy constraints, O'Leary [16] proposed a generalized CG method for solving general box constrained quadratic programming problems with a symmetric positive definite matrix, which can be naturally applied for solving NNLS problems. Similar algorithm called restricted LSQR method was presented by Lötstedt [13], where LSQR is a stabilized version of CGLS, and Bierlaire, Toint and Tuyttens [4] introduced a variant of the algorithm. In addition, instead of shrinking the step size, some researchers considered so-called projection-type methods, which orthogonally project the iterated solution into the feasible region. For example, the projected Landweber method, and the projected steepest descent method with a suitable preconditioner are proposed by Bertero and Boccacci [2], and Nagy et al. [15], respectively. Bierlaire et al. [4] also discuss projected gradient methods with active set strategy. However, the disadvantage is that the iteration in the inner iteration is terminated as soon as a component of a computed iterate violates a constraint, which forces frequent restart of the outer iteration and thus slows down convergence. Another undesirable feature is that the active set type algorithm allows only one variable to leave a bound at a given outer iteration, which allows to add or delete one index from the active set at a time. This is a very inefficient feature when the number of variables is large.

In order to avoid these disadvantages, Calvetti et al. [6] proposed a projected restarted iteration method for nonnegative constrained ill-posed problems by allowing more consecutive iterations in the inner iteration. The algorithm is given as follows.

ALGORITHM 1.1. **Projected Restarted Iteration Method**

1. *Choose an initial approximate solution* $\boldsymbol{x}^0$ *and compute* $\boldsymbol{r}^0 = \boldsymbol{b} - A\boldsymbol{x}^0$.
2. *For* $k = 0, 1, 2, \ldots$ *until convergence*
3. *Compute an approximate solution* $\boldsymbol{w}^k$ *by an iterative method*

$$\min_{\boldsymbol{w} \in \mathbf{R}^n} \|A\boldsymbol{w} - \boldsymbol{r}^k\|_2.$$

4. *Compute* $\hat{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^k + \boldsymbol{w}^k$ *and project it on the nonnegative region*

$$x_j^{k+1} = \begin{cases} 0, & \hat{x}_j^{k+1} < 0; \\ \hat{x}_j^{k+1}, & \hat{x}_j^{k+1} \geq 0. \end{cases} \qquad j = 1, 2, ..., n.$$

5. *Compute* $\boldsymbol{r}^{k+1} = \boldsymbol{b} - A\boldsymbol{x}^{k+1}$.
6. *Endfor*

Here, the unconstrained least squares problem for each loop is solved with CGLS, GMRES and RRGMRES iterative methods until the stopping criterion is satisfied.

In addition, Morigi et al. [14] proposed an active set restarted projected CG method for general box constrained ill-posed problems, which can be applied to nonnegative constrained problems, where the components of the solution that equal their bounds are referred to as the active set and identified in the outer iteration, and the reduced unconstrained least squares problem is solved in the inner iteration by keeping the identified components fixed. The nonnegative constrained version of the algorithm is given as follows.

ALGORITHM 1.2. **Active Set Projected Restarted Iteration Method**

1. *Choose an initial approximate solution* $\boldsymbol{x}^0$ *and compute* $\boldsymbol{r}^0 = \boldsymbol{b} - A\boldsymbol{x}^0$.
2. *For* $k = 0, 1, 2, \ldots$ *until convergence*
3. *Define Lagrange multipliers* $\boldsymbol{\lambda}^k = -A^T\boldsymbol{r}^k$.
4. *Define active set* $\mathcal{B}$ *and free variable set* $\mathcal{F}$

$$\mathcal{B} = \{j : x_j^k = 0,\ \lambda_j^k \geq 0\}, \quad \mathcal{F} = \{1, 2, ..., n\}\backslash\mathcal{B}.$$

5. *Compute an approximate solution* $\boldsymbol{w}^k$ *by an iterative method*

$$\min_{\boldsymbol{w}\in\mathbf{R}^{\check{n}}} \|A_{\mathcal{F}}\boldsymbol{w} - \boldsymbol{r}^k\|_2.$$

6. *Compute*

$$\hat{\boldsymbol{x}}_{\mathcal{F}}^{k+1} = \boldsymbol{x}_{\mathcal{F}}^k + \boldsymbol{w}^k,\ \hat{\boldsymbol{x}}_{\mathcal{B}}^{k+1} = \boldsymbol{x}_{\mathcal{B}}^k,$$

and project it on the nonnegative region

$$x_j^{k+1} = \begin{cases} 0, & \hat{x}_j^{k+1} < 0; \\ \hat{x}_j^{k+1}, & \hat{x}_j^{k+1} \geq 0. \end{cases} \qquad j = 1, 2, ..., n.$$

7. *Compute* $\boldsymbol{r}^{k+1} = \boldsymbol{b} - A\boldsymbol{x}^{k+1}$.
8. *Endfor*

Here, $\check{n}$ denotes the number of elements in set $\mathcal{F}$, and $A_{\mathcal{F}}$ denotes the submatrix of A consisting of the columns of A whose indices belong to $\mathcal{F}$. These methods are shown to require low storage requirement and are easy to implement, and numerical examples arising from constrained linear ill-posed problems as well as image restoration indicate their fairly rapid convergence. However, there is no theoretical analysis to guarantee the convergence, and the norm of consecutively generated residual vectors might not be monotonically decreasing.

In this paper, instead of using shrinking step size or the projection techniques, we apply a modulus transformation to constrain the nonnegativity of the variable, and the solution of NNLS problem (1.1) can be replaced by the solution of a sequence of unconstrained least squares problems, for which numerous efficient solvers can be exploited. Therefore, a new class of inner outer iterative methods is proposed by using CGLS method for inner iterations and the modulus-based iterative method in the outer iterations for the solution of LCP (Linear Complementarity Problem) resulting from KKT (Karush-Kuhn-Tucker) conditions of the NNLS. Theoretical convergence analysis is presented, and the choice of the parameter matrix is discussed for the proposed method. Numerical experiments show the efficiency of the proposed method compared to projection-type methods with less iteration steps and CPU time.

The rest of the paper is organized as follows. In Section 2, the modulus iterative method and the corresponding active set modulus method are proposed for the solution of the NNLS problem. In Section 3, the convergence analysis of the proposed methods are presented, and the choice of the parameter matrix is discussed. In Section 4, numerical results are presented, and Section 5 concludes the paper.

2. Modulus Iterative Methods. In this section, we show that the solution of the NNLS problem can be transformed to a series of unconstrained least squares problems by applying a modulus transformation on the variables. First, the equivalence between the nonnegative constrained quadratic programming and the linear complementarity problem (LCP) is shown in the following theorem, when the coefficient matrix is symmetric positive semidefinite.

THEOREM 2.1. *The nonnegative constrained quadratic programming* NNQP(B, c)

$$\min_{\boldsymbol{x}\in\mathbf{R}^n}\left(\frac{1}{2}\boldsymbol{x}^T B\boldsymbol{x}+\boldsymbol{c}^T\boldsymbol{x}\right), \quad \textit{subject to} \quad \boldsymbol{x}\geq\mathbf{0} \tag{2.1}$$

is equivalent to the linear complementarity problem LCP$(B, \boldsymbol{c})$

$$\boldsymbol{x}\geq\mathbf{0}, \quad B\boldsymbol{x}+\boldsymbol{c}\geq\mathbf{0}, \quad \textit{and} \quad \boldsymbol{x}^T(B\boldsymbol{x}+\boldsymbol{c})=0, \tag{2.2}$$

provided that B is a symmetric positive semidefinite matrix.

Proof. If $\boldsymbol{x}^*$ is a solution of LCP$(B, \boldsymbol{c})$, then it holds that

$$\boldsymbol{x}^*\geq\mathbf{0}, \quad B\boldsymbol{x}^*+\boldsymbol{c}\geq\mathbf{0}, \quad \text{and} \quad (\boldsymbol{x}^*)^{\mathrm{T}}(B\boldsymbol{x}^*+\boldsymbol{c})=0.$$

It is observed that for any $\boldsymbol{x}\geq 0$,

$$\begin{aligned}
&\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}B\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}-\left(\frac{1}{2}(\boldsymbol{x}^*)^{\mathrm{T}}B\boldsymbol{x}^*+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*\right)\\
=&\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathrm{T}}B(\boldsymbol{x}-\boldsymbol{x}^*)+\boldsymbol{x}^{\mathrm{T}}(B\boldsymbol{x}^*+\boldsymbol{c})-(\boldsymbol{x}^*)^{\mathrm{T}}(B\boldsymbol{x}^*+\boldsymbol{c})\\
=&\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathrm{T}}B(\boldsymbol{x}-\boldsymbol{x}^*)+\boldsymbol{x}^{\mathrm{T}}(B\boldsymbol{x}^*+\boldsymbol{c})\geq 0.
\end{aligned}$$

The last inequality holds by the fact that B is symmetric positive semidefinite. Hence, we have

$$\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}B\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}\geq\frac{1}{2}(\boldsymbol{x}^*)^{\mathrm{T}}B\boldsymbol{x}^*+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*,$$

which indicates that $\boldsymbol{x}^*$ is a minimization solution of NNQP$(B, \boldsymbol{c})$.

If $\boldsymbol{x}^*$ is a solution of NNQP$(B, \boldsymbol{c})$, then $\boldsymbol{x}^*$ satisfies the necessary KKT conditions as follows.

Stationarity

$$\nabla\left(\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}B\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{f}^{\mathrm{T}}\boldsymbol{x}\right)=B\boldsymbol{x}+\boldsymbol{c}-\boldsymbol{f}=\mathbf{0},$$

Primal and Dual feasibility

$$\boldsymbol{x}\geq\mathbf{0}, \quad \boldsymbol{f}\geq\mathbf{0},$$

Complementarity slackness

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{f}=0.$$

By collecting the KKT conditions above, it is derived that $\boldsymbol{x}^*$ satisfies LCP$(B, \boldsymbol{c})$. ■

COROLLARY 2.2. *If matrix B is symmetric positive definite, then both* NNQP$(B, \boldsymbol{c})$ *and* LCP$(B, \boldsymbol{c})$ *have the same unique solution.*

COROLLARY 2.3. *The NNLS problem* (1.1) *is equivalent to* LCP$(A^TA, -A^T\boldsymbol{b})$

$$\boldsymbol{x}\geq\mathbf{0}, \quad \boldsymbol{w}\equiv A^TA\boldsymbol{x}-A^T\boldsymbol{b}\geq\mathbf{0}, \quad \textit{and} \quad \boldsymbol{x}^T\boldsymbol{w}=0. \tag{2.3}$$

Proof. Set $B = A^{\mathrm{T}}A$ and $\boldsymbol{c} = -A^{\mathrm{T}}\boldsymbol{b}$ in Theorem 2.1. ■

Furthermore, the following theorem, where the proof can be easily obtained by Theorem 2.1 in [1], implies that $\mathrm{LCP}(A^{\mathrm{T}}A, -A^{\mathrm{T}}\boldsymbol{b})$ is equivalent to the implicit fixed-point equation

$$(\Omega + A^{\mathrm{T}}A)\boldsymbol{z} = (\Omega - A^{\mathrm{T}}A)|\boldsymbol{z}| + A^{\mathrm{T}}\boldsymbol{b} \tag{2.4}$$

with modulus transformation $\boldsymbol{x} = \boldsymbol{z} + |\boldsymbol{z}|$, where Ω is a positive diagonal parameter matrix. Hence, it is equivalent to solve the implicit fixed-point equation (2.4) for the solution of (1.1) by Corollary 2.3.

THEOREM 2.4. *Let Ω be an $n\times n$ positive diagonal matrix. For the* $\mathrm{LCP}(A^{T}A, -A^{T}\boldsymbol{b})$, *the following statements hold:*

(i) *if $(\boldsymbol{x}, \boldsymbol{w})$ is a solution of the* $\mathrm{LCP}(A^{T}A, -A^{T}\boldsymbol{b})$, *then $\boldsymbol{z} = (\boldsymbol{x} - \Omega^{-1}\boldsymbol{w})/2$ satisfies the implicit fixed-point equation* (2.4)*;*

(ii) *if $\boldsymbol{z}$ satisfies the implicit fixed-point equation* (2.4), *then*

$$\boldsymbol{x} = |\boldsymbol{z}| + \boldsymbol{z} \quad \textit{and} \quad \boldsymbol{w} = \Omega(|\boldsymbol{z}| - \boldsymbol{z})$$

is a solution of the $\mathrm{LCP}(A^{T}A, -A^{T}\boldsymbol{b})$.

Based on the equivalence in Theorem 2.4, the modulus-type iterative scheme

$$(\Omega + A^{\mathrm{T}}A)\boldsymbol{z}^{k+1} = (\Omega - A^{\mathrm{T}}A)|\boldsymbol{z}^{k}| + A^{\mathrm{T}}\boldsymbol{b} \tag{2.5}$$

is naturally derived for the solution of the fixed-point equation (2.4). If $\boldsymbol{z}^*$ is a fixed point of (2.5), then by Theorems 2.1 and 2.4, the solution of the NNLS problem (1.1) can be obtained straightforwardly by $\boldsymbol{x}^* = \boldsymbol{z}^* + |\boldsymbol{z}^*|$. Therefore, the solution of the NNLS problem (1.1) is transformed to the solution of a series of fixed-point equations (2.5), which can be solved directly by matrix decompositions, or by iterative methods, such as the preconditioned conjugate gradient (PCG) method.

The modulus restarted iterative method for NNLS problem (1.1) is described as follows.

ALGORITHM 2.5. **Modulus Restarted Iteration Method**

1. *Choose an initial approximate solution $\boldsymbol{z}^0$ and a parameter matrix Ω.*
2. *Compute $\boldsymbol{x}^0 = \boldsymbol{z}^0 + |\boldsymbol{z}^0|$.*
3. *For $k = 0, 1, 2, \ldots$ until convergence*
4. *Compute a solution $\boldsymbol{z}^{k+1}$ by solving the fixed-point equation* (2.5).
5. *Compute $\boldsymbol{x}^{k+1} = \boldsymbol{z}^{k+1} + |\boldsymbol{z}^{k+1}|$.*
6. *Endfor*

We remark that the modulus method derived from (2.4) is not only a special case of accelerated modulus-based matrix splitting methods when $M_1 = M_2 = A^{\mathrm{T}}A$ and $N_1 = N_2 = 0$ in [21], but also includes the original modulus method [19, 20], the modified modulus method [8], and the extrapolated modulus method [10] when $\Omega = I$, αI and $(1/\alpha)I$ with $\alpha > 0$, respectively. For more numerical methods of LCP, see [21] and the references therein.

Finally, it is noted that the iterative scheme (2.5) can be reorganized as the normal equations

$$\widetilde{A}^{\mathrm{T}}\widetilde{A}\boldsymbol{z}^{k+1} = \widetilde{A}^{\mathrm{T}}\tilde{\boldsymbol{b}}^{k}, \tag{2.6}$$

of the unconstrained least squares problem

$$\min_{\boldsymbol{z}^{k+1}\in\mathbf{R}^n} \|\widetilde{A}\boldsymbol{z}^{k+1} - \tilde{\boldsymbol{b}}^{k}\|_2 \tag{2.7}$$

for any fixed $k = 0, 1, 2, ...$, where

$$\widetilde{A} = \begin{bmatrix} A \\ \Omega^{\frac{1}{2}} \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{b}}^k = \begin{bmatrix} -A|\boldsymbol{z}^k| + \boldsymbol{b} \\ \Omega^{\frac{1}{2}}|\boldsymbol{z}^k| \end{bmatrix},$$

Therefore, the solution of the NNLS problem (1.1) is transformed to the solution of a series of unconstrained least squares problems (2.7), which can be solved efficiently by CGLS or GMRES type methods [11] with various preconditioning techniques [12].

The modulus-type inner outer iterative method for NNLS problem (1.1) is described as follows.

ALGORITHM 2.6. **Modulus-Type Inner Outer Iteration Method**

Choose an initial approximate solution $\boldsymbol{z}^0$ *and a parameter matrix* Ω.
Set $\widetilde{A} = [A^T,\ \Omega^{1/2}]^T$ *and* $\tilde{\boldsymbol{b}}^0 = [(-A|\boldsymbol{z}^0| + \boldsymbol{b})^T,\ (\Omega^{1/2}|\boldsymbol{z}^0|)^T]^T$.
Compute $\boldsymbol{x}^0 = \boldsymbol{z}^0 + |\boldsymbol{z}^0|$ *and* $\boldsymbol{r}^0 = \boldsymbol{b} - A\boldsymbol{x}^0$.
Compute $\tilde{\boldsymbol{r}}^0 = [(\boldsymbol{r}^0)^T,\ (\Omega^{1/2}(|\boldsymbol{z}^0| - \boldsymbol{z}^0))^T]^T (= \tilde{\boldsymbol{b}}^0 - \widetilde{A}\boldsymbol{z}^0)$.
For $k = 0, 1, 2, \ldots$ *until convergence*
    Compute an approximate solution $\boldsymbol{w}^k$ *by solving*

$$\min_{\boldsymbol{w} \in \mathbf{R}^n} \|\widetilde{A}\boldsymbol{w} - \tilde{\boldsymbol{r}}^k\|_2. \tag{2.8}$$

    Compute $\boldsymbol{z}^{k+1} = \boldsymbol{z}^k + \boldsymbol{w}^k$.
    Compute $\boldsymbol{x}^{k+1} = \boldsymbol{z}^{k+1} + |\boldsymbol{z}^{k+1}|$ *and* $\boldsymbol{r}^{k+1} = \boldsymbol{b} - A\boldsymbol{x}^{k+1}$.
    Set $\tilde{\boldsymbol{b}}^{k+1} = [(-A|\boldsymbol{z}^{k+1}| + \boldsymbol{b})^T,\ (\Omega^{1/2}|\boldsymbol{z}^{k+1}|)^T]^T$.
    Compute $\tilde{\boldsymbol{r}}^{k+1} = [(\boldsymbol{r}^{k+1})^T,\ (\Omega^{1/2}(|\boldsymbol{z}^{k+1}| - \boldsymbol{z}^{k+1}))^T]^T (= \tilde{\boldsymbol{b}}^{k+1} - \widetilde{A}\boldsymbol{z}^{k+1})$.
Endfor

Here, the iterative solution of the unconstrained least squares problems (2.8) for each $k = 0, 1, 2, ...$ is referred to as the inner iteration of the algorithm, while the for loop is referred to as the outer iteration. Note that Algorithm 2.5 is a special version of Algorithm 2.6 when the inner linear least squares problems (2.8) are solved based on their normal equations (2.5).

Similar to [14], the corresponding active set version of modulus iterative algorithm is given as follows.

ALGORITHM 2.7. **Active Set Modulus-Type Inner Outer Iteration Method**

Choose an initial approximate solution $\boldsymbol{z}^0$ *and a parameter matrix* Ω.
Set $\widetilde{A} = [A^T,\ \Omega^{1/2}]^T$ *and* $\tilde{\boldsymbol{b}}^0 = [(-A|\boldsymbol{z}^0| + \boldsymbol{b})^T,\ (\Omega^{1/2}|\boldsymbol{z}^0|)^T]^T$.
Compute $\boldsymbol{x}^0 = \boldsymbol{z}^0 + |\boldsymbol{z}^0|$ *and* $\boldsymbol{r}^0 = \boldsymbol{b} - A\boldsymbol{x}^0$.
Compute $\tilde{\boldsymbol{r}}^0 = [(\boldsymbol{r}^0)^T,\ (\Omega^{1/2}(|\boldsymbol{z}^0| - \boldsymbol{z}^0))^T]^T (= \tilde{\boldsymbol{b}}^0 - \widetilde{A}\boldsymbol{z}^0)$.
For $k = 0, 1, 2, \ldots$ *until convergence*
    Define Lagrange multipliers $\lambda^k = -A^T\boldsymbol{r}^k$.
    Define active set $\mathcal{B}$ *and free variable set* $\mathcal{F}$

$$\mathcal{B} = \{j : x_j^k = 0,\ \lambda_j^k \geq 0\}, \quad \mathcal{F} = \{1, 2, ..., n\} \backslash \mathcal{B}.$$

    Compute an approximate solution w^k *by an iterative method*

$$\min_{\boldsymbol{w} \in \mathbf{R}^{\tilde{n}}} \|\tilde{A}_{\mathcal{F}}\boldsymbol{w} - \tilde{\boldsymbol{r}}^k\|_2.$$

    Compute $\boldsymbol{z}_{\mathcal{F}}^{k+1} = \boldsymbol{z}_{\mathcal{F}}^k + \boldsymbol{w}^k,\ \boldsymbol{z}_{\mathcal{B}}^{k+1} = \boldsymbol{z}_{\mathcal{B}}^k$.
    Set $\tilde{\boldsymbol{b}}^{k+1} = [(-A|\boldsymbol{z}^{k+1}| + \boldsymbol{b})^T,\ (\Omega^{1/2}|\boldsymbol{z}^{k+1}|)^T]^T$.

11. *Compute* $\boldsymbol{x}^{k+1} = \boldsymbol{z}^{k+1} + |\boldsymbol{z}^{k+1}|$ *and* $\boldsymbol{r}^{k+1} = \boldsymbol{b} - A\boldsymbol{x}^{k+1}$.

12. *Compute* $\tilde{\boldsymbol{r}}^{k+1} = [(\boldsymbol{r}^{k+1})^{\mathrm{T}},\ (\Omega^{1/2}(|\boldsymbol{z}^{k+1}| - \boldsymbol{z}^{k+1}))^{\mathrm{T}}]^{\mathrm{T}} (= \tilde{\boldsymbol{b}}^{k+1} - \widetilde{A}\boldsymbol{z}^{k+1})$.

13. *Endfor*

Here, $\check{n}$ denotes the number of elements in set $\mathcal{F}$, and $\tilde{A}_{\mathcal{F}}$ denotes the submatrix of $\tilde{A}$ consisting of the columns of $\tilde{A}$ whose indices belong to $\mathcal{F}$.

3. Convergence Analysis. In this section, we establish the convergence theory of the Algorithm 2.5 in which the inner unconstrained least squares problems (2.8) are solved based on the normal equations (2.5). Specifically, we would discuss the cases when the inner systems are solved exactly or inexactly, respectively, as well as the theoretically optimal choice of the iteration parameter matrix Ω.

Assume that $\boldsymbol{z}^* \in \mathbf{R}^n$ is a solution of implicit fixed-point equation (2.5)

$$(\Omega + A^{\mathrm{T}}A)\boldsymbol{z}^* = (\Omega - A^{\mathrm{T}}A)|\boldsymbol{z}^*| + A^{\mathrm{T}}\boldsymbol{b}, \tag{3.1}$$

and $\boldsymbol{z}^{k+1}$ is calculated exactly from $\boldsymbol{z}^k$ by (2.5). After subtracting (3.1) from (2.5), we obtain

$$\boldsymbol{z}^{k+1} - \boldsymbol{z}^* = (\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)(|\boldsymbol{z}^k| - |\boldsymbol{z}^*|), \tag{3.2}$$

provided that $\Omega + A^{\mathrm{T}}A$ is nonsingular. The error relationship (3.2) is the basis for us to establish convergence theorems about Algorithm 2.5. The following analysis is based on the condition that A is of full column rank and thus $A^{\mathrm{T}}A$ is symmetric positive definite.

3.1. Scalar matrix case. Consider the case when $\Omega = \omega I$ with $\omega > 0$, it follows from taking vector norm $\|\cdot\|_2$ of both sides of (3.2) that

$$\begin{aligned}\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^*\|_2 &\le \|(\omega I + A^{\mathrm{T}}A)^{-1}(\omega I - A^{\mathrm{T}}A)\|_2 \||\boldsymbol{z}^k| - |\boldsymbol{z}^*|\|_2 \\ &\le \|(\omega I + A^{\mathrm{T}}A)^{-1}(\omega I - A^{\mathrm{T}}A)\|_2 \|\boldsymbol{z}^k - \boldsymbol{z}^*\|_2\end{aligned}$$

By simple calculations, it can be easily obtained that $(\omega I + A^{\mathrm{T}}A)^{-1}(\omega I - A^{\mathrm{T}}A)$ is symmetric. Therefore,

$$\|(\omega I + A^{\mathrm{T}}A)^{-1}(\omega I - A^{\mathrm{T}}A)\|_2 = \max_{\lambda_i \in \sigma(A^{\mathrm{T}}A)} \left|\frac{\omega - \lambda_i}{\omega + \lambda_i}\right|,$$

where $\sigma(A^{\mathrm{T}}A)$ denotes the set of all eigenvalues of $A^{\mathrm{T}}A$. As A is of full column rank, it follows that $\lambda_i > 0$ and

$$\left|\frac{\omega - \lambda_i}{\omega + \lambda_i}\right| < 1,$$

for any i, and thus

$$\|(\omega I + A^{\mathrm{T}}A)^{-1}(\omega I - A^{\mathrm{T}}A)\|_2 < 1.$$

Consequently, the iteration sequence $\{\boldsymbol{z}^k\}_{k=0}^{+\infty}$ generated by (2.5) converges to the unique solution $\boldsymbol{z}^*$ for any initial vector.

Let $\lambda_{\min}$ and $\lambda_{\max}$ be the minimum and maximum eigenvalues of $A^{\mathrm{T}}A$, respectively. It can be easily calculated that the optimal ω^* is

$$\omega^* \equiv \arg\min_{\omega} \left\{ \max_{\lambda_{\min} \le \lambda \le \lambda_{\max}} \left|\frac{\omega - \lambda}{\omega + \lambda}\right| \right\} = \sqrt{\lambda_{\min}\lambda_{\max}}$$

and

$$\|(\omega^* I + A^{\mathrm{T}}A)^{-1}(\omega^* I - A^{\mathrm{T}}A)\|_2 = \frac{\sqrt{\lambda_{\max}} - \sqrt{\lambda_{\min}}}{\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}}} = \frac{\sqrt{\kappa(A^{\mathrm{T}}A)} - 1}{\sqrt{\kappa(A^{\mathrm{T}}A)} + 1},$$

where $\kappa(A^{\mathrm{T}}A)$ denotes the spectral condition number of matrix $A^{\mathrm{T}}A$.

3.2. General positive diagonal matrix case. Consider the general case when Ω is a positive diagonal matrix. We define a norm that is useful in the sequel discussions. For all $\boldsymbol{x} \in \mathbf{R}^n$, $\|\boldsymbol{x}\|_P \equiv \sqrt{\boldsymbol{x}^{\mathrm{T}} P \boldsymbol{x}^{\mathrm{T}}}$ and $\|\boldsymbol{x}\|_{P,q} \equiv \|P\boldsymbol{x}\|_q$ define vector norms on $\mathbf{R}^n$, where $P \in \mathbf{R}^{n\times n}$ is an arbitrary nonsingular matrix; see [1, 17]. Moveover, if $X \in \mathbf{R}^{n\times n}$, then $\|X\|_{P,q} \equiv \|PXP^{-1}\|_q$. It follows from taking vector norm $\|\cdot\|_{\Omega^{1/2},2}$ of both sides of (3.2) that

$$\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^*\|_{\Omega^{1/2},2} \le \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2} \||\boldsymbol{z}^k| - |\boldsymbol{z}^*|\|_{\Omega^{1/2},2}. \tag{3.3}$$

Note that

$$\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^*\|_{\Omega^{1/2},2} = \|\Omega^{1/2}(\boldsymbol{z}^{k+1} - \boldsymbol{z}^*)\|_2 = \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^*\|_\Omega$$

and

$$\begin{aligned}
&\|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2} \\
&= \|\Omega^{-1/2}(I + (A\Omega^{-1/2})^{\mathrm{T}}(A\Omega^{-1/2}))^{-1}\Omega^{-1/2}\Omega^{1/2}(I - (A\Omega^{-1/2})^{\mathrm{T}}(A\Omega^{-1/2}))\Omega^{1/2}\|_{\Omega^{1/2},2} \\
&= \|(I + (A\Omega^{-1/2})^{\mathrm{T}}(A\Omega^{-1/2}))^{-1}(I - (A\Omega^{-1/2})^{\mathrm{T}}(A\Omega^{-1/2}))\|_2 \\
&\equiv \|(I + \widehat{A}^{\mathrm{T}}\widehat{A})^{-1}(I - \widehat{A}^{\mathrm{T}}\widehat{A})\|_2,
\end{aligned}$$

where $\widehat{A} \equiv A\Omega^{-1/2}$. Therefore, (3.3) gives

$$\begin{aligned}
\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^*\|_\Omega &\le \|(I + \widehat{A}^{\mathrm{T}}\widehat{A})^{-1}(I - \widehat{A}^{\mathrm{T}}\widehat{A})\|_2 \||\boldsymbol{z}^k| - |\boldsymbol{z}^*|\|_\Omega \\
&\le \|(I + \widehat{A}^{\mathrm{T}}\widehat{A})^{-1}(I - \widehat{A}^{\mathrm{T}}\widehat{A})\|_2 \|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega.
\end{aligned}$$

Notice that $\widehat{A} \equiv A\Omega^{-1/2}$ is of full column rank as A is of full column rank and Ω is a positive diagonal matrix. Hence $\widehat{A}^{\mathrm{T}}\widehat{A}$ is symmetric positive definite and

$$\|(I + \widehat{A}^{\mathrm{T}}\widehat{A})^{-1}(I - \widehat{A}^{\mathrm{T}}\widehat{A})\|_2 = \max_{\hat{\lambda}_i \in \sigma(\widehat{A}^{\mathrm{T}}\widehat{A})} \left|\frac{1 - \hat{\lambda}_i}{1 + \hat{\lambda}_i}\right| < 1.$$

Consequently, the iteration sequence $\{\boldsymbol{z}^k\}_{k=0}^{+\infty}$ generated by (2.5) converges to the unique solution $\boldsymbol{z}^*$ for any initial vector.

Next, the choice of the parameter matrix Ω is discussed. Set $\Omega = \bar{\omega}D$, where $D \equiv \mathrm{diag}(A^{\mathrm{T}}A)$ denotes the diagonal part of $A^{\mathrm{T}}A$ and $\bar{\omega}$ is a positive scalar parameter. Then $\widehat{A} = \bar{\omega}^{-1/2}AD^{-1/2} \equiv \bar{\omega}^{-1/2}\bar{A}$ and

$$\begin{aligned}
&\|(I + \widehat{A}^{\mathrm{T}}\widehat{A})^{-1}(I - \widehat{A}^{\mathrm{T}}\widehat{A})\|_2 \\
&= \|(I + \bar{\omega}^{-1}\bar{A}^{\mathrm{T}}\bar{A})^{-1}(I - \bar{\omega}^{-1}\bar{A}^{\mathrm{T}}\bar{A})\|_2 \\
&= \|(\bar{\omega}I + \bar{A}^{\mathrm{T}}\bar{A})^{-1}(\bar{\omega}I - \bar{A}^{\mathrm{T}}\bar{A})\|_2.
\end{aligned}$$

Similar to the previous analysis, the optimal parameter can be obtained by

$$\bar{\omega}^* = \sqrt{\bar{\lambda}_{\min}\bar{\lambda}_{\max}},$$

where $\bar{\lambda}_{\min}$ and $\bar{\lambda}_{\max}$ are the minimum and maximum eigenvalues of $\bar{A}^{\mathrm{T}}\bar{A}$, respectively. In addition,

$$\|(\bar{\omega}^* I + \bar{A}^{\mathrm{T}}\bar{A})^{-1}(\bar{\omega}^* I - \bar{A}^{\mathrm{T}}\bar{A})\|_2 = \frac{\sqrt{\bar{\lambda}_{\max}} - \sqrt{\bar{\lambda}_{\min}}}{\sqrt{\bar{\lambda}_{\max}} + \sqrt{\bar{\lambda}_{\min}}} = \frac{\sqrt{\kappa(\bar{A}^{\mathrm{T}}\bar{A})} - 1}{\sqrt{\kappa(\bar{A}^{\mathrm{T}}\bar{A})} + 1},$$

where $\kappa(\bar{A}^{\mathrm{T}}\bar{A})$ denotes the spectral condition number of matrix $\bar{A}^{\mathrm{T}}\bar{A}$.

Remark that $\bar{A}^{\mathrm{T}}\bar{A} = D^{-1/2}A^{\mathrm{T}}AD^{-1/2}$ can be regarded as a symmetric diagonal scaling preconditioning of $A^{\mathrm{T}}A$. Hence, it may be more efficient to choose $\Omega = \omega D$ than to choose $\Omega = \omega I$ in the modulus iteration Algorithm 2.5.

3.3. Convergence of inexact inner iteration. Finally, the convergence analysis based on the inexact solution of the implicit fixed-point equation (2.5) is considered. Suppose $\boldsymbol{z}^k$ has already been computed. Then, $\boldsymbol{z}^{k+1}$ is computed by applying iterative methods, such as the PCG method, to (2.5). Thus, we have

$$(\Omega + A^{\mathrm{T}}A)\boldsymbol{z}^{k+1} = (\Omega - A^{\mathrm{T}}A)|\boldsymbol{z}^k| + A^{\mathrm{T}}\boldsymbol{b} + \boldsymbol{e}^k. \tag{3.4}$$

In addition, we define

$$\boldsymbol{\varepsilon}^k = (\Omega + A^{\mathrm{T}}A)\boldsymbol{z}^k - (\Omega - A^{\mathrm{T}}A)|\boldsymbol{z}^k| - A^{\mathrm{T}}\boldsymbol{b}.$$

Note that if $\boldsymbol{\varepsilon}^k = 0$ for some fixed k, then $\boldsymbol{x}^* = \boldsymbol{x}^k$ is an exact solution of the fixed-point equation (2.4).

Assume that $\|\boldsymbol{e}^k\| \le \gamma_k\|\varepsilon^k\|$. Then it follows by subtracting (3.1) from (3.4) that

$$\begin{aligned}
&\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^*\|_\Omega \\
&= \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)(|\boldsymbol{z}^k| - |\boldsymbol{z}^*|) + (\Omega + A^{\mathrm{T}}A)^{-1}\boldsymbol{e}^k\|_\Omega \\
&\le \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2}\||\boldsymbol{z}^k| - |\boldsymbol{z}^*|\|_\Omega + \|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}\|\boldsymbol{e}^k\|_\Omega \\
&\le \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2}\|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega + \gamma_k\|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}\|\boldsymbol{\varepsilon}^k\|_\Omega \\
&= \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2}\|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega \\
&+ \gamma_k\|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}\|(\Omega + A^{\mathrm{T}}A)\boldsymbol{z}^k - (\Omega - A^{\mathrm{T}}A)|\boldsymbol{z}^k| - A^{\mathrm{T}}\boldsymbol{b}\|_\Omega \\
&= \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2}\|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega \\
&+ \gamma_k\|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}\|(\Omega + A^{\mathrm{T}}A)(\boldsymbol{z}^k - \boldsymbol{z}^*) - (\Omega - A^{\mathrm{T}}A)(|\boldsymbol{z}^k| - |\boldsymbol{z}^*|)\|_\Omega \\
&\le \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2}\|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega \\
&+ \gamma_k\|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}(\|\Omega + A^{\mathrm{T}}A\|_{\Omega^{1/2},2} + \|\Omega - A^{\mathrm{T}}A\|_{\Omega^{1/2},2})\|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega \\
&=: L_k\|\boldsymbol{z}^k - \boldsymbol{z}^*\|_\Omega.
\end{aligned}$$

Hence, we only need to verify that $L_k \le \theta < 1$, where θ is a scalar constant independent of k.

Set

$$\begin{aligned}
\kappa &\equiv \kappa(\Omega + A^{\mathrm{T}}A) = \|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}\|\Omega + A^{\mathrm{T}}A\|_{\Omega^{1/2},2}, \\
\delta &\equiv \|(\Omega + A^{\mathrm{T}}A)^{-1}(\Omega - A^{\mathrm{T}}A)\|_{\Omega^{1/2},2}, \\
\mu &\equiv \|(\Omega + A^{\mathrm{T}}A)^{-1}\|_{\Omega^{1/2},2}\|\Omega - A^{\mathrm{T}}A\|_{\Omega^{1/2},2}.
\end{aligned}$$

By the fact that $\delta < 1$, we have

$$\theta \equiv \alpha + (1 - \alpha)\delta < 1,$$

where $\alpha \in [0, 1)$. If there exists an integer k_0 such that for all $k \geq k_0$,

$$L_k = \delta + \gamma_k(\kappa + \mu) \leq \theta \quad \Rightarrow \quad \gamma_k \leq \frac{\alpha(1-\delta)}{\kappa + \mu},$$

then it follows that for $k \geq k_0$, $L_k \leq \theta < 1$, which guarantees the convergence of the iteration sequence $\{\boldsymbol{z}^k\}_{k=0}^{+\infty}$ generated by inexact modulus iteration for any initial vector.

Combining the analysis above, we have the following theorem.

THEOREM 3.1. *If A is of full column rank, then the iteration sequence $\{\boldsymbol{x}^k\}_{k=0}^{\infty}$ generated by modulus-type inner outer iteration Algorithm 2.6 converges to the unique solution $\boldsymbol{x}^*$ for any initial vector when*

- *the inner system is solved exactly;*
- *or the inner system is solved iteratively with*

$$\|\boldsymbol{e}^k\|_\Omega \leq \gamma_k \|\varepsilon^k\|_\Omega \quad \textit{with} \quad \gamma_k \leq \frac{\alpha(1-\delta)}{\kappa + \mu},$$

for $k \geq k_0$, where k_0 is an integer and $\alpha \in [0, 1)$.

4. Numerical Experiments. Finally, numerical experiment results are presented to show the performance of modulus-type inner outer iteration methods. We compare them to the projection-type inner outer iteration methods for overdetermined problems. All of the computation are run on a personal computer with 2.20 GHz CPU and 2 GB memory. The programming language is Matlab 7.8 with machine precision $\epsilon = 1.1 \times 10^{-16}$.

In addition, all initial vectors for outer and inner iteration are chosen to be $\boldsymbol{x}^0 = [0, 0, ..., 0]^{\mathrm{T}} \in \mathbf{R}^n$, and the parameter matrix $\Omega = \omega I$ and $\Omega = \omega \mathrm{diag}(A^{\mathrm{T}}A)$ are chosen for modulus iteration methods, respectively, where ω is a positive parameter. The inner least squares problems (2.8) are solved by either backslash in Matlab or CGLS method. We define the residual as

$$\mathrm{Res}(\boldsymbol{x}^k) \equiv \| \min(A^{\mathrm{T}}A\boldsymbol{x}^k - A^{\mathrm{T}}\boldsymbol{b}, \boldsymbol{x}^k)\|_2, \tag{4.1}$$

set the relative residual

$$\frac{\mathrm{Res}(x^k)}{\mathrm{Res}(x^0)} < tol \tag{4.2}$$

to be the stopping criterion for the outer iteration, and set

$$\frac{\|A^{\mathrm{T}}(\boldsymbol{b} - A\boldsymbol{x}^k)\|_2}{\|A^{\mathrm{T}}(\boldsymbol{b} - A\boldsymbol{x}^0)\|_2} < tol_{in} \equiv 10^{-2}/k \tag{4.3}$$

to be the stopping criterion for the inner iteration. Hence, the inner systems (2.8) are solved more and more accurately as k increases. Note that (4.2) is also frequently used as a stopping criterion of LCPs [21], and $\mathrm{Res}(\boldsymbol{x}^k) = 0$ if and only if $\boldsymbol{x}^k$ is a solution of the NNLS problem (1.1) by Corollary 2.3. In order to perform a fair comparison among different methods, we use the same stopping criterion, and the maximum number k of outer iteration steps (20000), for all methods.

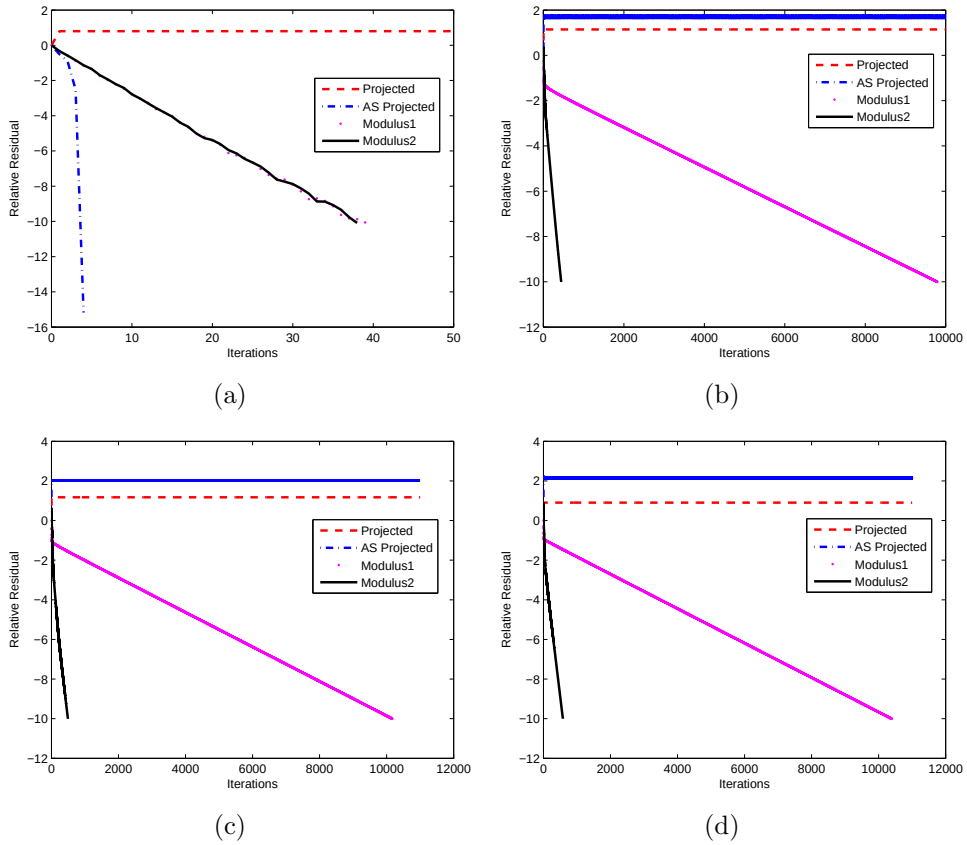


FIG. 1. *Relative residual vs. iterations for (a)* $\rho = 1$*, (b)* $\rho = 0.8$*, (c)* $\rho = 0.6$ *and (d)* $\rho = 0.4$ *with* $\sigma_n = 0.01$ $(\kappa(A) = 100)$.

4.1. Dense full rank case. First, we show how the condition number and the distribution of singular values of A influence the convergence of the modulus-type and projection-type methods with a class of dense matrices of the form $A = U\Sigma V^{\mathrm{T}}$, where $U \in \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$ are orthogonal matrices obtained from the QR decomposition of random matrices, and $\Sigma \in \mathbf{R}^{m\times n}$ is a rectangular diagonal matrix with diagonal entries $\sigma_1 > \sigma_2 > ... > \sigma_n$, where the ith smallest singular value is

$$\sigma_{n-i+1} = \sigma_n + \frac{i-1}{n-1}(\sigma_1 - \sigma_n)\rho^{n-i}, \quad i = 1, ..., n,$$

with the parameter $\rho \in (0, 1]$. Note that when ρ decreases, the singular values are tightly clustered towards the smallest singular value σ_n and are far apart towards the largest singular value σ_1. The idea of generating this kind of matrices is from [9, 11].

In our numerical experiments, we set $m = 200$, $n = 100$, $\sigma_1 = 1$, $\sigma_n = 0.01$ or 0.0001, $\rho = 1,\ 0.8,\ 0.6,\ 0.4$, and form inconsistent NNLS problems where the elements of vector $\boldsymbol{b}$ are generated randomly following the normal distribution with mean zero and variance 1, using the Matlab function randn$(m, 1)$. The same $\boldsymbol{b}$ is used for all the cases.

When the inner unconstrained least squares problems (2.8) are solved by backslash "\" in Matlab, we compare the four testing methods, projected method (denoted by

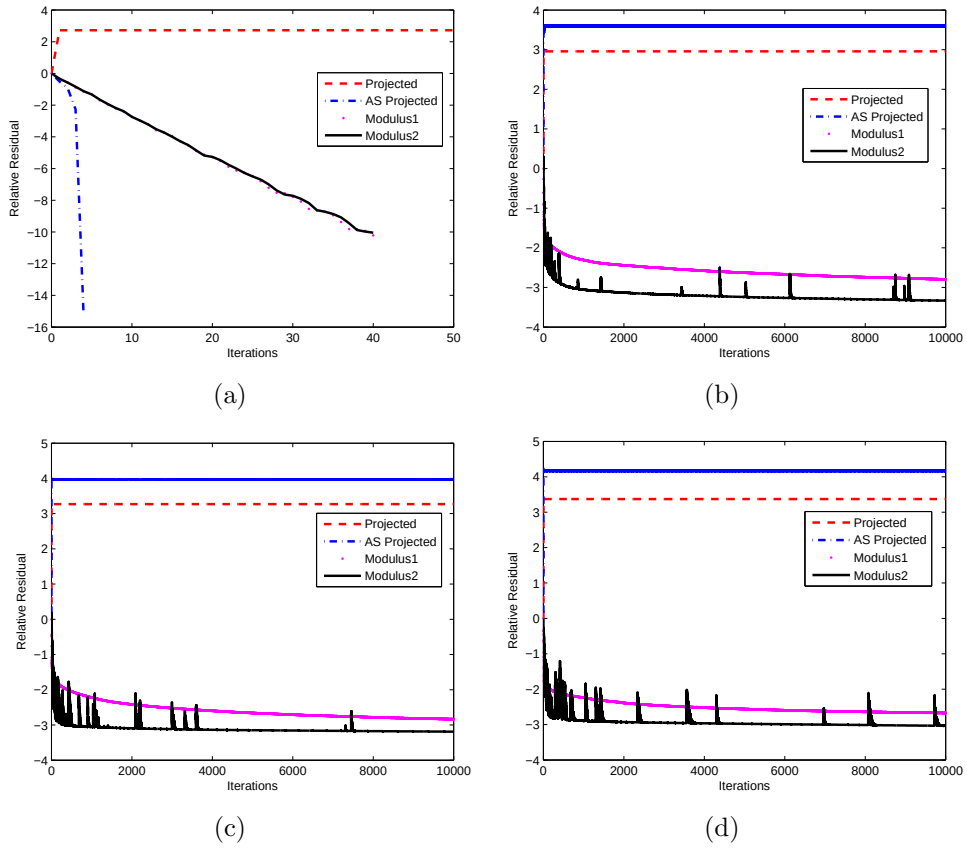


FIG. 2. *Relative residual vs. iterations for (a)* $\rho = 1$*, (b)* $\rho = 0.8$*, (c)* $\rho = 0.6$ *and (d)* $\rho = 0.4$ *with* $\sigma_n = 0.0001$ $(\kappa(A) = 10^4)$.

"Projected"), active set projected method (denoted by "AS-Projected"), modulus method with $\Omega = \omega I$ (denoted by "Modulus1"), and modulus method with $\Omega = \omega \mathrm{diag}(A^{\mathrm{T}}A)$ (denoted by "Modulus2"). The optimal ω was chosen by changing it from 0.1 to 2 with an interval of 0.1, so that it gives the minimum number of outer iterations.

In Figures 1 and 2, we depict the curves of the relative residual $\mathrm{Res}(x^k)/\mathrm{Res}(x^0)$ of the testing methods versus the number of outer iteration steps with $\sigma_n = 0.01$ and $\sigma_n = 0.0001$, respectively. In each figure, there are four diagrams denoted by (a), (b), (c) and (d) corresponding to $\rho = 1,\ 0.8,\ 0.6$ and 0.4, respectively.

From Figure 1, it is observed that the active set projected method outperforms other iterative methods for the case when the singular values are uniform distributed with $\rho = 1$, while fails to converge when the singular values cluster towards the smallest singular value in (b), (c) and (d). The projected method fails to converge for any ρ. In addition, the relative residual of the Modulus2 method declines much more rapidly than any other iterative methods as the iteration steps increase except for the case $\rho = 1$ in (a). This shows that the choice of $\Omega = \omega \mathrm{diag}(A^{\mathrm{T}}A)$ in Modulus2 is more efficient than the choice of $\Omega = \omega I$ in Modulus1, which confirms our convergence analysis.

TABLE 1
Comparison of the iterative methods (full rank and inconsistent problem).

(ρ, σ_n, tol)	AS-PCGLS	MCGLS1	AS-MCGLS1	MCGLS2	AS-MCGLS2
$(1, 10^{-2}, 10^{-10})$	6(13.00) 175 ∗0.00	39(8.62) 790 0.03	45(5.20) 604 0.02	38(9.03) 801 0.02	45(7.07) 772 0.03
$(0.8, 10^{-2}, 10^{-10})$	–	9,262(6.18) 142,269 2.22	9,790(3.81) 103,975 2.70	433(45.30) 40,526 ∗0.62	904(24.86) 47,665 0.69
$(0.6, 10^{-2}, 10^{-10})$	–	10,167(4.53) 122,636 1.95	10,167(3.33) 98,184 2.68	497(37.77) 39,036 ∗0.59	1,210(25.97) 66,469 0.95
$(0.4, 10^{-2}, 10^{-10})$	–	10,386(5.67) 148,999 2.34	10,386(2.97) 92,933 2.70	552(33.54) 38,687 ∗0.59	1,226(28.10) 72,585 1.03
$(1, 10^{-4}, 10^{-10})$	6(13.17) 177 ∗0.00	40(8.70) 817 0.02	47(5.17) 628 0.02	40(9.15) 853 0.03	49(6.67) 802 0.02
$(0.8, 10^{-4}, 10^{-3})$	–	–	–	669(42.39) 58,732 0.89	765(33.07) 52,900 ∗0.80
$(0.6, 10^{-4}, 10^{-3})$	–	–	–	392(43.44) 35,237 ∗0.58	547(34.28) 39,148 0.59
$(0.4, 10^{-4}, 10^{-2})$	–	179(4.26) 2,062 0.05	176(3.24) 1,669 0.06	25(32.96) 1,724 0.03	16(26.63) 901 ∗0.02

First row: number of outer iterations (IT) and average inner iterations (Inner).
Second row: number of matrix vector multiplication (MV).
Third row: computational time in seconds (CPU).

Figure 2 shows similar convergence phenomena as in Figure 1. As the iteration steps increase, the relative residual of the active set projected method decreases much more rapidly than any other iterative methods for the case $\rho = 1$ in (a), while the Modulus2 method is most efficient when the singular values cluster towards the smallest singular value in (b), (c) and (d). Note that the convergence behavior of all four methods deteriorate as the condition number of the matrix A increases, as can be seen by comparing Figures 1 and 2.

The optimal ω used for the Modulus1 and the Modulus2 methods in diagram (a) of Figure 1 were 0.2 and 0.6, respectively. For the other diagrams in Figure 1, the optimal ω was 0.1. The optimal ω for Figure 2 were the same as in Figure 1 for all the corresponding modulus-type methods.

When the inner least squares problems (2.8) are solved by the CGLS method, in Table 1 we compare the six testing methods, projected CGLS method (denoted by "PCGLS"), modulus method with $\Omega = \omega I$ (denoted by "MCGLS1"), modulus method with $\Omega = \omega \mathrm{diag}(A^{\mathrm{T}}A)$ (denoted by "MCGLS2"), and their corresponding active set methods (denoted by "AS-PCGLS", "AS-MCGLS1" and "AS-MCGLS2"),

TABLE 2
Comparison of the iterative methods (full rank and inconsistent problem).

Problem		PCGLS	AS-PCGLS	MCGLS2	AS-MCGLS2
Randn_1	IT(Inner)	–	3(13.33)	16(13.81)	16(12.00)
	MV		90	491	433
	CPU		∗0.03	0.23	0.16
	ω			0.7	0.7
Randn_2	IT(Inner)	–	–	68(69.90)	102(34.91)
	MV			9,711	7,429
	CPU			4.10	∗2.34
	ω			0.2	0.3
Randn_3	IT(Inner)	–	–	333(211.92)	338(122.67)
	MV			142,136	83,941
	CPU			60.70	∗25.85
	ω			0.1	0.1
Randn_4	IT(Inner)	–	–	4,784(102.46)	4,785(67.55)
	MV			994,647	660,784
	CPU			431.94	∗210.13
	ω			0.1	0.1
Randn_5	IT(Inner)	–	–	588(185.13)	592(111.88)
	MV			219,483	134,239
	CPU			94.38	∗41.42
	ω			0.4	0.4
Randn_6	IT(Inner)	–	–	886(200.98)	890(85.72)
	MV			358,803	155,255
	CPU			154.47	∗48.50
	ω			0.3	0.3
Randn_7	IT(Inner)	–	–	975(157.57)	650(103.79)
	MV			310,182	136,875
	CPU			133.32	∗42.68
	ω			0.3	0.2
Randn_8	IT(Inner)	–	–	530(271.40)	265(140.48)
	MV			289,275	75,250
	CPU			123.91	∗23.38
	ω			0.2	0.1

$$\mathrm{Res}(x^k)/\mathrm{Res}(x^0) < 10^{-5}.$$

from the aspects of outer iteration steps (denoted by "IT"), average inner iteration steps (denoted by "Inner"), the number of matrix vector multiplications (denoted by "MV"), and CPU time in seconds (denoted by "CPU"). The optimal ω was used in the modulus-type methods, which minimized the number of matrix vector multiplications. The stopping criteria for outer and inner iteration were chosen as (4.2) and as (4.3) with $tol_{in} = 10^{-2}/k$, respectively.

In Table 1, we list the numerical results for each method. The numerical results for PCGLS is not shown here since it did not converge within the maximum iteration steps for all the cases. The symbol "–" indicates that the iterative method failed

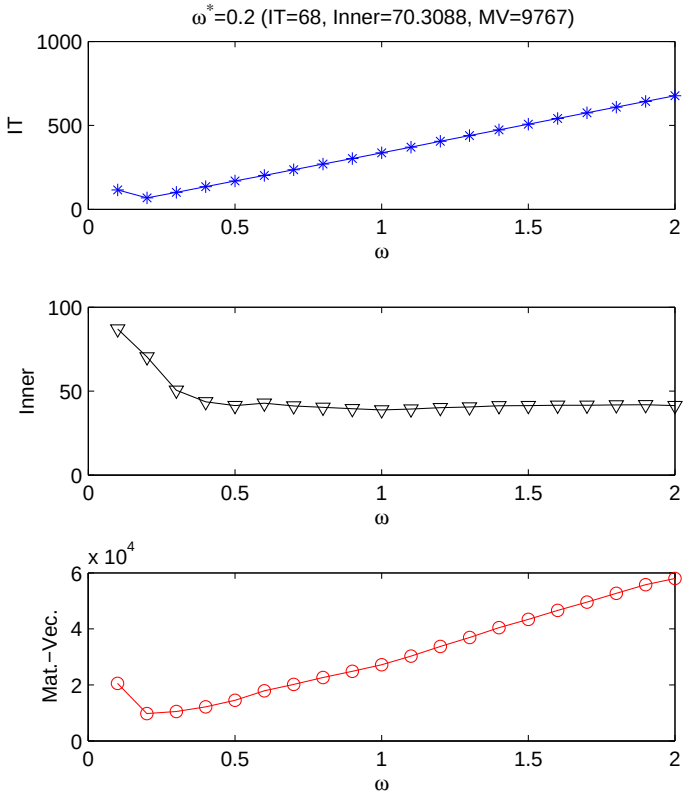


FIG. 3. *Number of outer iterations, inner iterations and matrix vector multiplication vs. ω for MCGLS2 method in Randn_2.*

to converge within the maximum iteration steps, and "$*$" denotes the most efficient method with least CPU time among all the testing methods. For all the methods,

$$\text{MV} = 2\text{IT}(\text{Inner} + 1) + 1 \tag{4.4}$$

holds, which is the relation between number of matrix vector multiplications with the number of inner and outer iterations.

From Table 1, it is observed that AS-PCGLS method is more efficient than other iterative methods with less outer iteration, matrix vector multiplication and CPU time when $\rho = 1$. For $\rho < 1$, the projection-type methods including PCGLS and AS-PCGLS fail to converge, while modulus type methods could obtain the numerical solution in most cases. Also, the MCGLS2 and AS-MCGLS2 methods outperform the MCGLS1 and AS-MCGLS1 methods with less computational costs. In addition, it further confirms that the convergence behavior of the four methods deteriorate as the condition number of the matrix A increases as shown in the Figures 1 and 2. This is the reason why we decrease the *tol* to 10^{-3} and 10^{-2} when $\rho = 0.8,\ 0.6,\ 0.4$ with $\sigma_n = 0.0001$, otherwise, all the numerical methods can not converge within the maximum iteration steps. Note that generally the CPU time show positive correlation

TABLE 3
Information on the practical test matrices.

Problem	m	n	nnz	dens. [%]	rank	$\kappa(A)$
Maragal_3	1,682	858	18,391	1.27	613	1.10×10^3
Maragal_4	1,964	1,027	26,719	1.32	801	9.33×10^6
Maragal_5	4,654	3,296	93,091	0.61	2,147	1.19×10^5
Maragal_6	21,251	10,144	537,694	0.25	8,331	2.91×10^6
Maragal_7	46,845	26,525	1,200,537	0.10	20,843	8.98×10^6
Maragal_8	60,845	33,093	1,308,415	0.06	–	–

with matrix vector multiplications, since matrix vector multiplication is the main computational costs in the algorithms. Therefore, the iterative methods with least CPU time have least number of matrix vector multiplications.

Although the active set type methods require less average inner iterations than non-active set type methods, it does not show significant acceleration of convergence since they require more outer iterations for $\sigma_n = 0.01$.

The optimal ω used for the MCGLS1 and AS-MCGLS1 methods in the case $\rho = 1$ and $\sigma_n = 10^{-2},\ 10^{-4}$ in Table 1 were 0.2, while for the MCGLS2 and AS-MCGLS2 methods the optimal ω were 0.6. For all the other cases, the optimal ω was 0.1.

4.2. Sparse full rank case. Next, we generate a class of sparse full column rank matrices, abbreviated as "Randn_i", $i = 1, 2, 3, 4, 5, 6, 7, 8$, using the Matlab function "sprandn" with $m = 30,000$, $n = 3,000$, and the ratio of nonzero elements density= 0.1%. The condition numbers of these matrices are specified as

$$\kappa(\text{Randn\_i}) = 10^i, \quad i = 1, 2, 3, 4, 5, 6, 7, 8.$$

The nonzero element values were generated by a random number generator following the normal distribution, and the pattern of the nonzero elements is also determined by a random number generator. In these experiments, we form inconsistent NNLS problems where the elements of the right-hand side vector $\boldsymbol{b}$ are generated randomly using the Matlab function randn$(m, 1)$. The same $\boldsymbol{b}$ is used for all the cases.

Since the MCGLS2 method was shown to be more efficient than the MCGLS1 method in the previous subsection, in these experiments we only compare the PCGLS, AS-PCGLS, MCGLS2 and AS-MCGLS2 methods with $\Omega = \omega \text{diag}(A^{\mathrm{T}}A)$. The numerical results are shown in Table 2. The tolerance for the outer iterations was chosen as $tol = 10^{-5}$.

Table 2 shows that the PCGLS method fails to converge for all the cases. The AS-PCGLS method is much faster than modulus-type method for Randn_1 with $\kappa(A) = 10$. However, when the condition number is larger, the projection-type methods do not converge within the maximum iteration numbers while the modulus-type methods converged. Moreover, the AS-MCGLS2 method was more efficient than the MCGLS2 method with less inner iterations, matrix vector multiplications and CPU time. Hence, the active set strategy accelerates the convergence behavior of the modulus method in this full rank inconsistent case. In each cell, the optimal ω is shown in the fourth row.

Figure 3 shows how the number of outer iteration, inner iterations and matrix vector multiplication are affected by the choice of the parameter ω for the MCGLS2 method for Randn_2. The parameter ω varies from 0.1 to 2 with an interval of 0.1.

TABLE 4
Comparison of the iterative methods (rank-deficient and consistent problem).

Problem		PCGLS	AS-PCGLS	MCGLS	AS-MCGLS
Maragal_3	IT(Inner)	2,003(465.24)	35(301.94)	435(26.80)	487(23.62)
	MV	1,869,754	21,242	24,618	24,472
	CPU	115.49	∗1.20	1.78	1.78
	ω			1.0	1.2
Maragal_4	IT(Inner)	1,479(396.90)	416(506.03)	343(26.00)	358(23.54)
	MV	1,178,472	422,263	18,864	17,931
	CPU	91.53	28.78	1.72	∗1.59
	ω			0.9	1.1
Maragal_5	IT(Inner)	259(437.24)	22(395.68)	288(29.58)	351(26.99)
	MV	227,268	17,477	17,905	20,004
	CPU	53.02	∗3.59	4.56	5.09
	ω			1.4	1.8
Maragal_6	IT(Inner)	367(716.96)	24(725.29)	331(28.29)	374(27.47)
	MV	527,354	34,887	19,724	21,671
	CPU	941.75	59.78	∗39.27	44.71
	ω			1.7	1.9
Maragal_7	IT(Inner)	168(535.28)	18(788.22)	411(26.87)	434(28.57)
	MV	180,359	28,431	23,322	26,103
	CPU	816.73	116.66	∗114.72	127.25
	ω			1.9	2.2
Maragal_8	IT(Inner)	980(753.88)	17(1,154.12)	251(24.16)	292(22.67)
	MV	1,480,543	39,292	12,882	14,115
	CPU	7,551.91	182.85	∗71.25	79.69
	ω			2.1	2.6

$$\mathrm{Res}(x^k)/\mathrm{Res}(x^0) < 10^{-5}.$$

From Figure 3, it is observed that the number of outer iteration decreases at first, then increases, while the number of inner iteration always decreases as ω increases. This is because as ω increases, the coefficient matrix $\Omega + A^{\mathrm{T}}A$ in (2.5) becomes more diagonally dominant, and the number of inner iterations decreases. As a result, the optimal parameter with the least number of matrix vector multiplications was $\omega^* = 0.2$. It is noted that the dependence of MV on ω is moderate.

4.3. Sparse rank deficient case. In the following, we test a class of rectangular matrices from the University of Florida Sparse Matrix Collection [7]. We construct the rank-deficient overdetermined systems by deleting all the zero rows and zero columns. The resulting number of rows m, column n, nonzero elements nnz, as well as the rank, are given in Table 3.

For the consistent case, we form NNLS problems where the right-hand side vector $\boldsymbol{b} = A\boldsymbol{x}^*$ and $\boldsymbol{x}^* = [1, 0, 1, 0, ...]^{\mathrm{T}} \in \mathbf{R}^{n\times 1}$. The numerical results are shown in Table 4, where the tolerance for the outer iteration is chosen to be $tol = 10^{-5}$.

Table 4 shows that all the testing methods converge for all the cases, although the PCGLS method requires the most number of iterations and CPU time. For problems Maragal_3 and Maragal_5, the AS-PCGLS is slightly more efficient than

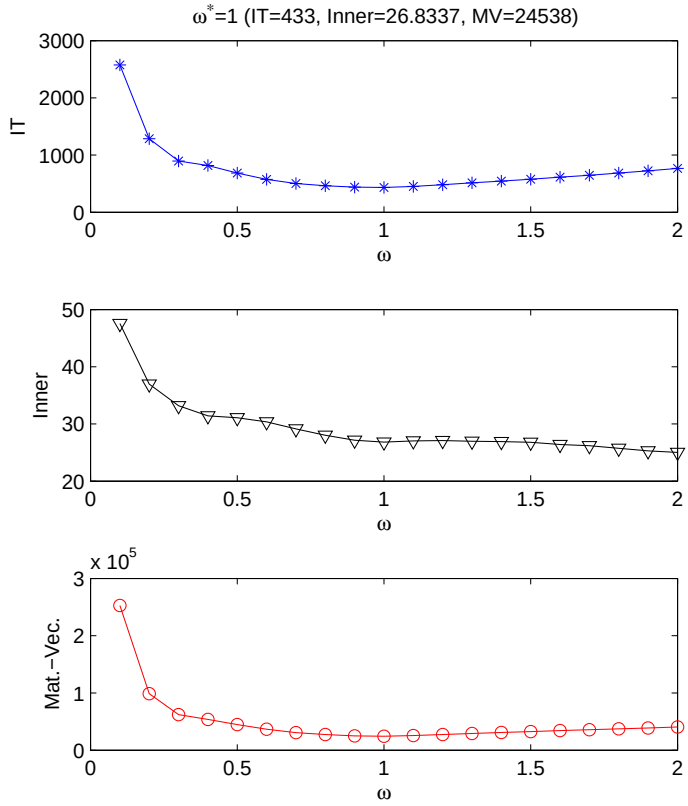


FIG. 4. *Number of outer iterations, inner iterations and matrix vector multiplication vs. ω for MCGLS method in Maragal_3.*

modulus-type methods. This may be related to the relatively small condition numbers $\kappa(A(\text{Maragal\_3})) \sim \mathcal{O}(10^3)$ and $\kappa(A(\text{Maragal\_5})) \sim \mathcal{O}(10^5)$ compared to other problems. As is shown in Section 4.1, the AS-PCGLS method outperforms other methods if the condition number is small and the singular values are uniformly distributed. For problem Maragal_i, $i = 4, 6, 7, 8$, the modulus-type methods are more efficient than projection-type methods. In addition, apart from Maragal_4, the active set strategy does not show significant acceleration compared to the MCGLS method, which is different from the sparse full rank case in Section 4.2.

Figure 4 shows the number of outer iterations, inner iterations and matrix vector multiplications vs. ω for the MCGLS method for Maragal_3.

As in Figure 3, the number of outer iterations decreases at first, and then increases, while the number of inner iterations always decreases as ω increases. The optimal parameter with the least number of matrix vector multiplications was $\omega^* = 1$.

For the inconsistent case, we form inconsistent NNLS problems where the right-hand side vector $\boldsymbol{b} = [1, 1, ..., 1]^{\mathrm{T}} \in \mathbf{R}^{n\times 1}$. The numerical results are shown in Table 5, where the tolerance for the outer iterations is chosen to be $tol = 10^{-3}$.

Table 5 shows that the projection-type methods do not converge for most cases,

TABLE 5
Comparison of the iterative methods (rank-deficient and inconsistent problem).

Problem		PCGLS	AS-PCGLS	MCGLS	AS-MCGLS
Maragal_3	IT(Inner)	–	–	261(30.60)	407(17.28)
	MV			16,758	15,286
	CPU			1.34	∗1.05
	ω			1.0	1.6
Maragal_4	IT(Inner)	–	53(425.19)	250(34.92)	464(17.75)
	MV		45,230	18,209	17,869
	CPU		2.81	1.61	∗1.42
	ω			0.7	1.4
Maragal_5	IT(Inner)	–	–	262(32.39)	448(21.84)
	MV			17,759	20,911
	CPU			4.71	∗4.62
	ω			1.3	2.3
Maragal_6	IT(Inner)	–	–	189(33.16)	296(23.70)
	MV			13,104	14,921
	CPU			25.79	∗24.55
	ω			1.5	2.3
Maragal_7	IT(Inner)	–	–	181(28.32)	208(24.84)
	MV			10,796	10,959
	CPU			52.70	∗48.89
	ω			2.4	2.7
Maragal_8	IT(Inner)	–	–	2,665(48.16)	652(45.97)
	MV			264,694	61,907
	CPU			1,464.66	∗193.86
	ω			0.6	0.6

$$\mathrm{Res}(x^k)/\mathrm{Res}(x^0) < 10^{-3}.$$

which is different from the consistent case in Table 4. For the modulus-type methods, it requires more iterations and CPU time compared to the consistent case. For problems Maragal_5, Maragal_6 and Maragal_7, the AS-MCGLS method requires slightly more matrix vector multiplication than MCGLS method, nevertheless, it requires least CPU time. This can be explained as in the inner iteration of AS-MCGLS method, a series of reduced least squares problem is solved which requires reduced size of matrix vector multiplication. Hence, AS-MCGLS is the most efficient method among all the testing methods.

4.4. Image restoration. Lastly, we apply the proposed method to the solution of nonnegative constrained ill-posed problems

$$A\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0}, \tag{4.5}$$

where $A \in \mathbf{R}^{n\times n}$ is a matrix with ill-determined rank and has many singular values of different orders of magnitude close to the origin. In many linear discrete ill-posed problems that arise in science and engineering, such as the restoration of an image, the right hand side vector is contaminated by blur and noise [6]. Hence, (4.5) is generally inconsistent and thus one has to solve a NNLS problem.

TABLE 6
Definitions in image restoration.

A	blurring operator
$\hat{\boldsymbol{x}}$	noise- and blur-free image
$\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}}$	blurred noise-free image
$\boldsymbol{e}$	noise
$\boldsymbol{b} = \hat{\boldsymbol{b}} + \boldsymbol{e}$	blurred and noisy image
$\gamma = \|\boldsymbol{e}\|_2/\|\hat{\boldsymbol{b}}\|_2$	noise level

(a) (b)

FIG. 5. *Exact image for (a) satellite and (b) variant motion.*

We test the numerical methods PCGLS and MCGLS2 on image restoration, where the data consists of the noise- and blur-free images shown in Figure 5, which come from Nagy's Matlab toolbox "RestoreTools" [3]. Some basic definitions in image restoration are shown in Table 6. Note that the matrix A is determined by the point spread function. Vector $\boldsymbol{e}$ is generated with normally distributed entries with zero mean by Matlab. The iteration is terminated when the numerical solution $\boldsymbol{x}^k$ satisfies the discrepancy principle as

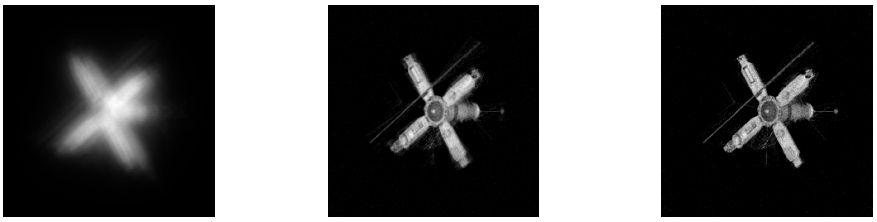
$$\|\boldsymbol{b} - A\boldsymbol{x}^k\|_2 \leq \eta\delta, \quad \delta = \|\boldsymbol{e}\|_2 = \|\boldsymbol{b} - \hat{\boldsymbol{b}}\|_2, \tag{4.6}$$

where $\eta \geq 1$ is a specified constant, or k reaches the maximal number of iteration steps, e.g., 10. The noise level δ is set to be 5% and the discrepancy factor $\eta = 1$ for all cases.

In Figures 6 and 7, the medium and large blurred and noisy image, the restored images of satellite by PCGLS and MCGLS2 are shown, respectively. The relative error of the image is defined as

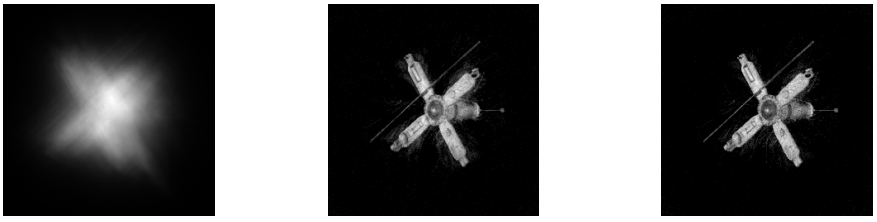
$$\text{Error} = \frac{\|\boldsymbol{x}^k - \hat{\boldsymbol{x}}\|_2}{\|\hat{\boldsymbol{x}}\|_2}.$$

The figures show that the MCGLS2 method could obtain more accurate numerical solutions and thus clearer images compared to the PCGLS method with the same computational costs.



Medium Blur Error=0.2230 Error=0.1829

FIG. 6. *Medium blurred and noisy image (left), restored image by PCGLS (middle), and restored image by MCGLS2 (right) of satellite image.*



Large Blur Error=0.2564 Error=0.2019

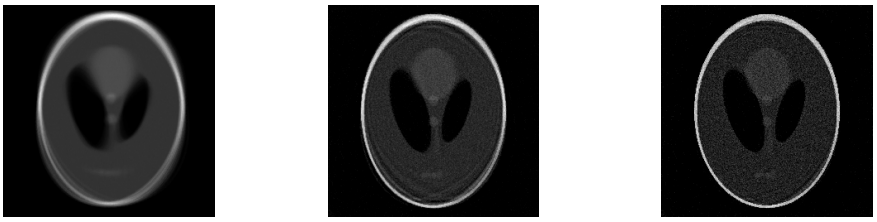
FIG. 7. *Large blurred and noisy image (left), restored image by PCGLS (middle), and restored image by MCGLS2 (right) of satellite image.*

In Figures 8 and 9, the medium and large blurred and noisy image, the restored images of variant motion by PCGLS and MCGLS2 are shown, respectively.

From the figures, the same conclusion can be reached that the MCGLS2 method could obtain more accurate numerical solutions than the PCGLS method with the same computational costs.

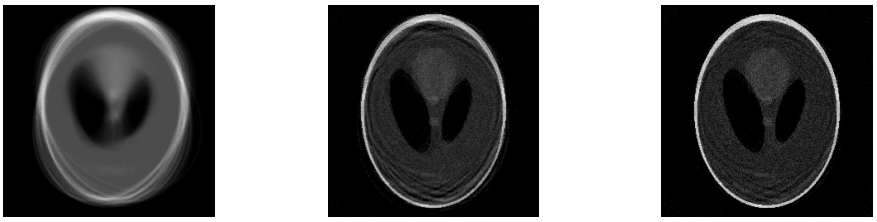
5. Concluding Remarks. A new class of inner outer iterative methods for nonnegative constrained least squares (NNLS) problem (1.1) was proposed based on the modulus transformation on the nonnegative variables. Thus, the solution of the NNLS problem (1.1) can be transformed into the solution of a sequence of unconstrained least squares problems. Theoretical convergence analysis was presented when the inner system is solved either exactly or iteratively, and the choice of the parameter matrix was discussed for the proposed method. Numerical experiments showed that the modulus-type methods are feasible for full column rank and rank deficient overdetermined NNLS problems. In addition, the modulus-type methods outperform projection-type methods with less iteration steps and CPU time when the coefficient matrix has ill-determined rank with large condition number and the singular values cluster near zero. We also applied our methods to nonnegative constrained ill-posed image restoration problems, and the numerical results showed that the proposed method is more accurate than projected method.

Acknowledgement. The authors would like to thank Dr. Keiichi Morikuni for valuable remarks which greatly improved the quality of the paper. The authors



Medium Blur Error=0.2453 Error=0.2375

FIG. 8. *Medium blurred and noisy image (left), restored image by PCGLS (middle), and restored image by MCGLS2 (right) of variant motion image.*



Large Blur Error=0.2754 Error=0.2296

FIG. 9. *Large blurred and noisy image (left), restored image by PCGLS (middle), and restored image by MCGLS2 (right) of variant motion image.*

also would like to thank Professors Jose Mas, Lothar Reichel, Per Christian Hansen, James Nagy, Zhong-Zhi Bai, Wei Xu, Miss Yiran Cui and Mr. Kota Sugihara for valuable discussions. The first author would like to express his gratitude to Professor Ken Hayami for his courteous attention and National Institute of Informatics for the financial support during the six-month internship in Tokyo.

REFERENCES

[1] Z.-Z. BAI, *Modulus-based matrix splitting iteration methods for linear complementarity problems*, Numer. Linear Algebra Appl., 6 (2010), pp. 917–933.
[2] M. BERTERO AND P. BOCCACCI, *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, Bristol, 1998.
[3] S. BERISHA AND J. G. NAGY, *Iterative Methods for Image Restoration*, (2012), http://www.mathcs.emory.edu/~nagy/RestoreTools.
[4] M. BIERLAIRE, PH. L. TOINT, AND D. TUYTTENS, *On iterative algorithms for linear least squares problems with bound constraints*, Linear Algebra Appl. 143 (1991), pp. 111–143.
[5] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[6] D. CALVETTI, G. LANDI, L. REICHEL, AND F. SGALLARI, *Nonnegativity and iterative methods for ill-posed problems*, Inverse Problems, 20 (2004), pp. 1747–1758.
[7] T. DAVIS, *The University of Florida Sparse Matrix Collection*, available online at http://www.cise.ufl.edu/research/sparse/matrices.
[8] J.-L. DONG AND M.-Q. JIANG, *A modified modulus method for symmetric positive-definite linear complementarity problems*, Numer. Linear Algebra Appl., 16 (2009), pp. 129–143.
[9] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
[10] A. HADJIDIMOS AND M. TZOUMAS, *Nonstationary extrapolated modulus algorithms for the solu-*

tion of the linear complementarity problem, Linear Algebra Appl., 431 (2009), pp. 197–210.
[11] K. Hayami, J.-F. Yin, and T. Ito, *GMRES methods for least squares problems*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2400–2430.
[12] K. Morikuni and K. Hayami, *Inner-iteration Krylov subspace methods for least squares problems*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1–22.
[13] P. Lötstedt, *Soving the minimal least squares problem subject to bounds on the variables*, BIT Numerical Mathematics, 24 (1984), pp. 206–224.
[14] S. Morigi, L. Reichel, F. Sgallari, and F. Zama, *An iterative method for linear discrete ill-posed problems with box constraints*, J. Comput. Appl. Math., 198 (2007), pp. 505–520.
[15] J. Nagy and Z. Strakoš, *Enforcing nonnegativity in image reconstruction algorithms*, in Mathematical Modeling, Estimation and Imaging, ed. D. C. Wilson et al., of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 4121, The International Society for Optical Engineering, Bellingham, WA, 2000, pp. 182–190.
[16] D. P. O'Leary, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra and its Applications, 34 (1980), pp. 371–399.
[17] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press: New York, 1970.
[18] B. T. Polyak, *The conjugate gradient method in extremal problems*, U.S.S.R. Computational Mathematics and Mathematical Physics, 9 (1969), pp. 94–112.
[19] W. M. G. van Bokhoven, *A Class of Linear Complementarity Problems is Solvable in Polynomial Time*, unpublished paper, Dept. of Electrical Engineering, University of Technology, The Netherlands (1980).
[20] W. M. G. van Bokhoven, *Piecewise-linear modelling and analysis*, Proeschrift, Eindhoven (1981).
[21] N. Zheng and J.-F. Yin, *Accelerated modulus-based matrix splitting iteration methods for linear complementarity problems*, Numer. Algorithms 64 (2013), pp. 245–262.