Improved Inexact Alternating Direction Methods for a Class of Nonlinear Complementarity Problems

Jiewen He\textsuperscript{1,*}, Hua Zheng\textsuperscript{2} and Seakweng Vong\textsuperscript{1}

\textsuperscript{1}Department of Mathematics, University of Macau, Taipa, Macao 999078, China.
\textsuperscript{2}School of Mathematics and Statistics, Shaoguan University, Shaoguan 512000, China.

Received 15 April 2021; Accepted (in revised version) 29 July 2021.

Abstract. Three types of improved inexact alternating direction methods for solving nonlinear complementarity problems with positive definite matrices and nonlinear terms are proposed. The convergence of the methods is proven. Numerical examples confirm the theoretical analysis and show that the methods have advantages over similar existing methods, especially in large size problems.

AMS subject classifications: 90C33, 65F10, 65F50, 65G40

Key words: Inexact alternating direction method, nonlinear complementarity problem, successive overrelaxation, iterative method, symmetric positive definite.

1. Introduction

Complementarity problems (CPs), first introduced by Cottle, are widely used in computing the equilibrium points of bimatrix games [11,28]. Such problems arise in scientific computing, engineering and economy when considering numerous problems in elasticity, traffic network design, optimal control, free boundary problems of fluid dynamics, asset pricing and image processing [14,19,20,23,37,60].

Over the past several decades, various numerical algorithms have been proposed to solve nonlinear complementarity problems (NCPs). On the other hand, Mangasarian [34] showed the equivalence of CPs to systems of nonlinear equations and Mangasarian and Solodov [35] reformulated NCPs as unconstrained and constrained minimisation problems, Noor [38] employed several fixed point methods to complementarity problems. Other approaches include projection-type methods [47], merit functions [16], smooth and nonsmooth Newton methods [10,13,27,40,41], interior-point methods [50], linearisation methods [18], domain decomposition methods [1,26]. Existence and uniqueness results for complementarity problems are summarised in surveys and books [17–19,23]. Using the solvers based on matrix splitting and multisplitting, Bai [2,3] and Bai and Wang [9]
developed parallel nonlinear monotonically convergent multisplitting relaxation methods. Asynchronous parallel nonlinear multisplitting relaxation methods for large sparse NCPs have been studied in [5, 6]. These methods are very efficient on high-speed MIMD multiprocessor systems since they allow to avoid the synchronous delays between corresponding processors. Matrix splitting methods are special cases of these multisplitting methods.

Here, we focus on a class of weakly NCPs. Recall that Bai [4] first considered the systems of weakly nonlinear equations, also called the mildly nonlinear systems [8]. Linear complementarity problems (LCPs) represent a special case of NCPs and in recent years, a number of LCP solvers — viz. modulus-based matrix splitting (MMS) iteration methods, attracted considerable attention [7, 31, 39], since very often they perform better than projected relaxation methods [12] and modified modulus methods [15]. In particular, Mezzadri [36] and Li [32] used MMSs to solve horizontal linear complementarity problems and second-order cone LCPs, respectively. MMS iteration methods have been also applied to a class of weakly NCPs — cf. [25, 45]. It turns out that MMSs are more efficient than the fixed point method [38] and the Fischer-Burmeister semismooth Newton method [13]. Zheng [53] improved the convergence of MMS iteration method for NCPs with $H$-matrices. The convergence results have been then extended from $H$-compatible splitting to $H$-splitting of NCPs with $H_+$-matrices [30]. Extensions of MMS methods, such as modified MMS method [33], accelerated MMS method [29], relaxed MMS method [58], two-step MMS method [46, 55], preconditioned MMS method [54, 59] and modulus-based matrix multisplitting methods [43, 44, 48, 49], developed in recent years improved the original one. Besides, a modulus-based nonsmooth Newton method and sign-based methods are studied in [57] and [56], respectively.

In recent years, alternating direction methods (ADMs) have been applied to LCPs of the free boundary problems of mathematical physics [21, 22, 42]. Zhang et al. [51, 52] developed inexact ADMs of multipliers for solving a kind of LCPs. These methods were applied to weakly NCPs in He et al. [24]. The main contribution of our work is as follows:

- Three new numerical methods for solving weakly NCPs are proposed and their convergence is analysed.

- The methods have their own advantages and are much more efficient than the existing methods.

In practical applications, the system matrix $A$ of weakly NCPs is usually large, sparse, symmetric and positive definite. This motivates the current study on different inexact ADMs. In Section 2, a direct inexact ADM, a symmetric successive overrelaxation (SSOR) based inexact ADM and a modified SSOR-based inexact ADM are introduced. Convergence of these methods is studied in Section 3. The numerical examples presented in Section 4, illustrate the theoretical results. Section 5 contains concluding remarks.

2. Improved Inexact ADMs

A nonlinear complementarity problem with respect to a vector-valued function $F : D \rightarrow \mathbb{R}^n$, where $D$ is a subset of $\mathbb{R}^n$ containing at least the nonnegative cone, consists in finding
a vector $u \in \mathbb{R}^n$, which satisfies the following conditions:

$$F(u) \geq 0, \quad u \geq 0, \quad u^T F(u) = 0.$$ 

In this paper, we consider a class of NCP derived from the following problem in a region $D \subset \mathbb{R}^2$:

$$-\mathcal{L} v(x, y) + \psi(v(x, y)) + f(x, y) \geq 0,$$
$$v(x, y) \geq 0,$$
$$v(x, y)[ -\mathcal{L} v(x, y) + \psi(v(x, y)) + f(x, y)] = 0,$$

and such that $v(x, y) = g(x, y)$ on the boundary $D$. The function $f$, difference operator $\mathcal{L}$, nonlinear function $\psi$ and the boundary condition $g$ are given. In numerical solution of the problem, the unknown function $v(x, y)$ is approximated by a vector $u$, which is defined on a grid and satisfies the conditions

$$Au + \phi(u) \geq 0,$$
$$u \geq 0,$$
$$u^T [Au + \phi(u)] = 0,$$

(2.1)

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix. The nonlinear term $\phi(u)$ can be written as

$$\phi(u) = \left( \psi_1(u_1), \psi_2(u_2), \ldots, \psi_n(u_n) \right)^T + b$$

with a vector $b$ generated by the boundary condition and the function $f(x, y)$. The coefficient matrix $A$ in (2.1) usually has a special structure.

In this section, we propose various improvements of inexact alternating direction methods for finding the solution of a class of NCPs. It is worth noting that the condition

$$\frac{d\psi_i}{du_i} \geq 0, \quad i = 1, 2, \ldots, n$$

(2.2)

is assumed throughout the paper. If $A$ is a symmetric positive definite matrix and $d\psi_i/du_i$, $i = 1, 2, \ldots, n$ are nonnegative, then according to [23], NCP has a unique solution $u^*$. Recall that $u^*$ is a solution of (2.1) if and only if $u^*$ solves the constrained programming problem

$$\min S(u) = \frac{1}{2} u^T Au + \Phi(u),$$
$$\text{subject to: } u \geq 0,$$

(2.3)

where

$$\Phi(u) = \sum_{i=1}^{n} \int_{-\infty}^{u_i} (\psi_i(u_i) + b_i) du_i$$

is convex on $u$ because of the condition (2.2). Therefore we can consider the problem (2.3) instead of (2.1).
Let \( \mathbb{R}_+^n := \{ u \in \mathbb{R}^n : u_i \geq 0, \ i = 1, 2, \ldots, n \} \) and \( G(u) \) be the indicator function of \( \mathbb{R}_+^n \), i.e.

\[
G(u) = \begin{cases} 
0, & \text{if } u \in \mathbb{R}_+^n, \\
+\infty, & \text{otherwise.}
\end{cases}
\]

The arguments similar to [52, Lemma 1] show that the problem (2.3) can be equivalently written as

\[
\min \{ S(u) + G(w) \},
\]

subject to: \( \Lambda (w - u) = 0, \ w \in \mathbb{R}_+^n \), (2.4)

where \( \Lambda \) is a given diagonal matrix with positive diagonal entries. For simplicity, here we set \( \Lambda = \mu I, \ \mu > 0 \).

The augmented Lagrangian function of (2.4) has the form

\[
L(u, w, \lambda) = S(u) + G(w) + \lambda^T \Lambda (w - u) + \frac{\beta}{2} \| \Lambda (w - u) \|^2,
\]

where \( \lambda \in \mathbb{R}^n \) is the Lagrangian multiplier and \( \beta > 0 \) is the penalty parameter. The classical alternating direction method of multipliers for the problem (2.4) or, equivalently, for the problem (2.3) is

\[
\begin{align*}
    u^{(k+1)} &= \arg\min_{u \in \mathbb{R}^n} L(u, w^{(k)}, \lambda^{(k)}), \\
    w^{(k+1)} &= \arg\min_{w \in \mathbb{R}_+^n} L(u^{(k+1)}, w, \lambda^{(k)}), \\
    \lambda^{(k+1)} &= \lambda^{(k)} + \beta \mu (w^{(k+1)} - u^{(k+1)}).
\end{align*}
\]

Considering the minimisation problems in (2.5), we have

\[
(A + \beta \Lambda^2) u^{(k+1)} + \phi(u^{(k+1)}) = \Lambda \lambda^{(k)} + \beta \Lambda^2 w^{(k)}. \tag{2.6}
\]

For the second minimisation problem in (2.5), we have

\[
w^{(k+1)} = \left( u^{(k+1)} - \frac{\Lambda^{-1} \lambda^{(k)}}{\beta} \right)_+, \tag{2.7}
\]

where \((.)_+\) denotes the projection on the set \( \mathbb{R}_+^n \).

We note that it is difficult to determine \( u^{(k+1)} \) directly from (2.6). Therefore, we will use an inexact approach — viz. we use the system of linear equations

\[
(A + \beta \Lambda^2) u^{(k+1)} \approx \Lambda \lambda^{(k)} + \beta \Lambda^2 w^{(k)} - \phi(u^{(k)}), \tag{2.8}
\]

where \( u^{(k)}, w^{(k)} \) and \( \lambda^{(k)} \) are given. The relations (2.7), (2.8) lead to the following algorithm.
Algorithm 2.1 Direct inexact ADM (DADM)

**Input:** $A$, $\phi$, $\Lambda = \mu I$, $\mu > 0$, $\alpha > 0$ and $\beta > 0$; give initial points $u^{(0)}$, $w^{(0)}$ and $\Lambda^{(0)}$.

**While** not converged, **do**

- Generate $u^{(k+1)}$ according to (2.8);
- Generate $w^{(k+1)}$ according to (2.7);
- Update $\lambda^{(k)}$ according to $\lambda^{(k+1)} = \lambda^{(k)} + \beta \Lambda (w^{(k+1)} - u^{(k+1)})$;

**End While**

It will be proven in the next section that $(u^{(k+1)}, w^{(k+1)})$ generated by (2.7), (2.8) converges to the solution $(u^*, w^*)$ of the problem (2.4).

At each iteration in Algorithm 2.1, we can find the solution of the system of linear equations (2.8) by linear solvers. However, it is not always possible to solve it directly, especially for large sparse problems. This motivates us to exploit more implementable iterative methods, thus leading to another inexact alternating direction method. Here, we use SSOR iterations to solve (2.8) approximately, and propose the SSOR-based inexact ADM for the symmetric positive definite NCPs.

Noting that $A$ is symmetric positive definite, we consider the splitting

$$A = D - L - U = D - U^T - L^T,$$

where $D$ is a diagonal matrix, $-L$ and $-U$ are strictly lower-triangular and upper-triangular matrices of $A$, respectively. After that we apply a SSOR method to (2.8). More exactly, given $u^{(k)}$, $w^{(k)}$ and $\lambda^{(k)}$, we determine $u^{(k+1)}$ from the equations

$$\begin{align*}
(D - \alpha L + \alpha \beta \Lambda^2)u^{(k+1/2)} &= ((1 - \alpha)D + \alpha U)u^{(k)} + \alpha \left( \Lambda \lambda^{(k)} + \beta \Lambda^2 w^{(k)} - \phi(u^{(k)}) \right), \\
(D - \alpha U + \alpha \beta \Lambda^2)w^{(k+1)} &= ((1 - \alpha)D + \alpha L)u^{(k+1/2)} + \alpha \left( \Lambda \lambda^{(k)} + \beta \Lambda^2 w^{(k)} - \phi(u^{(k)}) \right).
\end{align*}$$

Taking into account (2.7) and (2.9), we arrive at another algorithm.

Algorithm 2.2 SSOR-based inexact ADM (SADM)

**Input:** $D$, $L$, $U$ with $A = D - L - U$, $\phi$, $\Lambda = \mu I$, $\mu > 0$, $\alpha > 0$ and $\beta > 0$; give initial points $u^{(0)}$, $w^{(0)}$ and $\Lambda^{(0)}$.

**While** not converged, **do**

- Generate $u^{(k+1)}$ according to (2.9);
- Generate $w^{(k+1)}$ according to (2.7);
- Update $\lambda^{(k)}$ according to $\lambda^{(k+1)} = \lambda^{(k)} + \beta \Lambda (w^{(k+1)} - u^{(k+1)})$;

**End While**

We propose another type of efficient SSOR-based inexact ADM by using a different SSOR iteration for solving $u^{(k+1)}$ in (2.8). It is called the modified SSOR-based inexact ADM for
solving symmetric positive definite NCPs. Its effectiveness and comparison to SADM is tested by numerical experiments in Section 4.

Noting that $\tilde{A} = A + \beta \Lambda^2$ is also a symmetric positive definite matrix, we consider the splitting

$$\tilde{A} = \tilde{D} - L - U = \tilde{D} - U^T - L^T,$$

where $\tilde{D}$ is a diagonal matrix, and $-L, -U$ are strictly lower-triangular and upper-triangular matrices of $A$, respectively. After that, we employ a modified SSOR method to solve (2.8).

More exactly, given $u^{(k)}, w^{(k)}$ and $\lambda^{(k)}$, we find $u^{(k+1)}$ from the equations

$$\begin{align*}
(\tilde{D} - \alpha L)u^{(k+1/2)} &= ((1 - \alpha)\tilde{D} + \alpha U)u^{(k)} + \alpha \Lambda \lambda^{(k)} + \beta \Lambda^2 w^{(k)} - \phi(u^{(k)}), \\
(\tilde{D} - \alpha U)w^{(k+1)} &= ((1 - \alpha)\tilde{D} + \alpha L)u^{(k+1/2)} + \alpha \Lambda \lambda^{(k)} + \beta \Lambda^2 w^{(k)} - \phi(u^{(k)})).
\end{align*}$$

(2.10)

Exploiting (2.7) and (2.10), we finally get Algorithm 2.3.

**Algorithm 2.3 Modified SSOR-based inexact ADM (MSADM)**

**Input:** $\tilde{D}, L, U$ with $(A + \beta \Lambda^2) = \tilde{D} - L - U$, $\phi$, $\Lambda = \mu I$, $\mu > 0$, $\alpha > 0$ and $\beta > 0$; give initial points $u^{(0)}, w^{(0)}$ and $\lambda^{(0)}$.

**While** not converged, **do**

- Generate $u^{(k+1)}$ according to (2.10);
- Generate $w^{(k+1)}$ according to (2.7);
- Update $\lambda^{(k)}$ according to $\lambda^{(k+1)} = \lambda^{(k)} + \beta \Lambda (w^{(k+1)} - u^{(k+1)})$;

**End While**

### 3. Convergence

In this section, we show that the sequence $(u^{(k)}, w^{(k)})$ generated by Algorithms 2.1-2.3 converges to the solution $(u^*, w^*)$ of the problem (2.4). Recall that, according to the discussion in Section 2 the solution $u^*$ of (2.4) is also a solution of (2.1). Note that the solving of (2.4) is equivalent to the finding of $(u^*, w^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$, which satisfies the relations

$$\begin{align*}
(Au^* + \phi(u^*)) - \Lambda \lambda^* &= 0, \\
(w - w^*)^T \Lambda \lambda^* &\geq 0, \quad \forall w \in \mathbb{R}^n_+, \\
\Lambda (u^* - w^*) &= 0.
\end{align*}$$

(3.1)

Let $e_u^{(k)} = u^{(k)} - u^*$, $e_w^{(k)} = w^{(k)} - w^*$, $e_{\lambda}^{(k)} = \lambda^{(k)} - \lambda^*$ and

$$\begin{align*}
D^{(k)}_{\phi} &= \text{diag} \left( \frac{\phi_1(u_1^{(k)}) - \phi_1(u_1^*)}{u_1^{(k)} - u_1^*}, \ldots, \frac{\phi_n(u_n^{(k)}) - \phi_n(u_n^*)}{u_n^{(k)} - u_n^*} \right) \\
&= \text{diag} \left( \phi_1'(\xi_{1}^{(k)}), \ldots, \phi_n'(\xi_{n}^{(k)}) \right),
\end{align*}$$

where $\xi_{i}^{(k)}$ is located between $u_i^{(k)}$ and $u_i^*$, $i = 1, \ldots, n$. 

Theorem 3.1. Let $\phi_i'(x) \geq 0$, $i = 1, 2, \ldots, n$ for any $x \in \mathbb{R}$ and

$$d_\phi := \max_{i=1,\ldots,n} \left\{ \max_{x \in \mathbb{R}} \{ \phi_i'(x) \} \right\}.$$ 

If $S := A - d_\phi I$ is a symmetric positive definite matrix, then the sequence $(u^{(k)}, w^{(k)})$ generated by Algorithm 2.1 converges to the solution $(u^*, w^*)$ of the problem (2.4).

Proof. It follows from (2.8) that

$$(u^*-u^{(k+1)})^T [Au^{(k+1)} + \beta A^2(u^{(k+1)} - w^{(k)}) - \Lambda \lambda^{(k)} + \phi(u^{(k)})] = 0,$$

and (3.1) yields

$$(u^{(k+1)} - u^*)^T [Au^* + \beta A^2(u^* - w^*) - \Lambda \lambda^* + \phi(u^*)] = 0.$$ 

Consequently, we obtain

$$(u^*-u^{(k+1)})^T \left[ A e_u^{(k+1)} + \beta \Lambda^2 (e_u^{(k+1)} - e_u^{(k)}) - \Lambda \phi(e_u^{(k)}) - \phi(u^*) \right] = 0,$$

or

$$2\beta \langle e_u^{(k+1)}, A e_u^{(k+1)} \rangle + 2\beta^2 \Lambda^2 (e_u^{(k+1)} - e_u^{(k)}) - 2\beta^2 \Lambda^2 (e_u^{(k+1)} - e_u^{(k)})$$

$$- 2\beta \langle e_u^{(k+1)}, \Lambda e_\lambda^{(k)} \rangle + 2\beta \langle e_u^{(k+1)}, \phi(u^{(k)}) - \phi(u^*) \rangle = 0,$$ (3.2)

where

$$\langle x, y \rangle := x^T y = \sum_{i=1}^n x_i y_i, \quad x, y \in \mathbb{R}^n$$

is the inner product in $\mathbb{R}^n$. For compactness, the notation $\langle x, x \rangle = \|x\|^2$ is used whenever possible, and $\| \cdot \|$ denotes the Euclidean norm on $\mathbb{R}^n$.

The equation

$$\lambda^{(k+1)} = \lambda^{(k)} + \beta \Lambda \left( w^{(k+1)} - u^{(k+1)} \right)$$

and the third equation in (3.1) give

$$e_\lambda^{(k+1)} = e_\lambda^{(k)} + \beta \Lambda (e_u^{(k+1)} - e_u^{(k+1)}).$$

Therefore,

$$\|e_\lambda^{(k+1)}\|^2 - \|e_\lambda^{(k+1)}\|^2 = -2\beta \langle e_\lambda^{(k)}, \Lambda (e_u^{(k+1)} - e_u^{(k+1)}) \rangle - \beta^2 \|\Lambda (e_u^{(k+1)} - e_u^{(k+1)})\|^2.$$ 

Following [52], we have

$$2\beta^2 \|\Lambda e_u^{(k+1)}\|^2 - 2\beta^2 \langle \Lambda e_u^{(k+1)}, \Lambda e_u^{(k+1)} \rangle + 2\beta \langle \Lambda e_u^{(k+1)}, e_\lambda^{(k)} \rangle \leq 0,$$

which implies

$$\|e_\lambda^{(k)}\|^2 - \|e_\lambda^{(k+1)}\|^2 \geq 2\beta \langle e_\lambda^{(k)}, \Lambda e_u^{(k+1)} \rangle + \beta^2 \|\Lambda e_u^{(k+1)}\|^2 - \beta^2 \|\Lambda e_u^{(k+1)}\|^2.$$ (3.3)
Combining it with (3.2), we arrive at the estimate
\[ \|e^{(k)}_{\lambda}\|^2 - \|e^{(k+1)}_{\lambda}\|^2 \geq \beta^2 \|\Lambda e^{(k+1)}_{w}\|^2 + \beta^2 \|\Lambda e^{(k+1)}_{u}\|^2 - 2\beta^2 \Lambda^2 \langle e^{(k+1)}_{u}, e^{(k)}_{w}\rangle \\
+ 2\beta \langle e^{(k+1)}_{u}, A e^{(k+1)}_{u}\rangle + 2\beta \langle e^{(k+1)}_{u}, \phi u^{(k)} - \phi (u^a)\rangle. \]

Consequently,
\[ \|e^{(k)}_{\lambda}\|^2 - \|e^{(k+1)}_{\lambda}\|^2 + \beta^2 \left( \|\Lambda e^{(k+1)}_{w}\|^2 - \|\Lambda e^{(k+1)}_{u}\|^2 \right) \geq \beta^2 \|\Lambda (e^{(k)}_{w} - e^{(k+1)}_{u})\|^2 + 2\beta \langle e^{(k+1)}_{u}, A e^{(k+1)}_{u}\rangle + 2\beta \langle e^{(k+1)}_{u}, D_{\phi} e^{(k)}_{u}\rangle. \]

Let \( \phi'(x) \geq 0, \ i = 1, 2, \ldots, n \) for any \( x \in \mathbb{R} \). The Cauchy-Schwarz inequality yields
\[ 2\beta \langle e^{(k+1)}_{u}, D_{\phi} e^{(k)}_{u}\rangle \geq -2\beta \left\| \left(D_{\phi}^{(k)}\right)^{\frac{1}{2}} e^{(k+1)}_{u}\right\| \left\| \left(D_{\phi}^{(k)}\right)^{\frac{1}{2}} e^{(k)}_{u}\right\| \geq -\beta \left\| \left(D_{\phi}^{(k)}\right)^{\frac{1}{2}} e^{(k+1)}_{u}\right\|^2 - \beta \left\| \left(D_{\phi}^{(k)}\right)^{\frac{1}{2}} e^{(k)}_{u}\right\|^2, \]

and it follows that
\[ \|e^{(k)}_{\lambda}\|^2 - \|e^{(k+1)}_{\lambda}\|^2 + \beta^2 \left( \|\Lambda e^{(k+1)}_{w}\|^2 - \|\Lambda e^{(k+1)}_{u}\|^2 \right) \geq \beta^2 \|\Lambda (e^{(k)}_{w} - e^{(k+1)}_{u})\|^2 + 2\beta \langle e^{(k+1)}_{u}, A e^{(k+1)}_{u}\rangle - \beta d_{\phi} \|e^{(k+1)}_{u}\|^2 - \beta d_{\phi} \|e^{(k)}_{u}\|^2. \]

This inequality can be written in the form
\[ \|e^{(k)}_{\lambda}\|^2 - \|e^{(k+1)}_{\lambda}\|^2 + \beta^2 \left( \|\Lambda e^{(k+1)}_{w}\|^2 - \|\Lambda e^{(k+1)}_{u}\|^2 \right) + \beta d_{\phi} \left( \|e^{(k)}_{u}\|^2 - \|e^{(k+1)}_{u}\|^2 \right) \geq \beta^2 \|\Lambda (e^{(k)}_{w} - e^{(k+1)}_{u})\|^2 + 2\beta \langle e^{(k+1)}_{u}, (A - d_{\phi} I) e^{(k+1)}_{u}\rangle. \] (3.4)

Setting
\[ E^{(k)} := \|e^{(k)}_{\lambda}\|^2 + \beta^2 \|\Lambda e^{(k)}_{w}\|^2 + \beta d_{\phi} \|e^{(k)}_{u}\|^2, \]
we write (3.4) as
\[ E^{(k)} - E^{(k+1)} \geq \beta^2 \|\Lambda (e^{(k)}_{w} - e^{(k+1)}_{u})\|^2 + 2\beta \langle e^{(k+1)}_{u}, (A - d_{\phi} I) e^{(k+1)}_{u}\rangle. \] (3.5)

Summing the inequalities (3.5) in \( k \) from 0 to \( n \) gives
\[ E^{(0)} - E^{(n+1)} = \|e^{(0)}_{\lambda}\|^2 - \|e^{(n+1)}_{\lambda}\|^2 + \beta^2 \left( \|\Lambda e^{(0)}_{w}\|^2 - \|\Lambda e^{(n+1)}_{w}\|^2 \right) + \beta d_{\phi} \left( \|e^{(0)}_{u}\|^2 - \|e^{(n+1)}_{u}\|^2 \right) \geq \sum_{k=0}^{n} \beta^2 \|\Lambda (e^{(k)}_{w} - e^{(k+1)}_{u})\|^2 + \sum_{k=0}^{n} 2\beta \langle e^{(k+1)}_{u}, (A - d_{\phi} I) e^{(k+1)}_{u}\rangle. \]
Theorem 3.2. Under the conditions of Theorem 3.1 and 0 < α < 2, the sequence \((u^{(k)}, w^{(k)})\) generated by Algorithm 2.2 converges to the solution \((u^*, w^*)\) of the problem (2.4).

Proof: Combining the first equation in (3.1) and (2.9) gives

\[
\begin{align*}
(D - \alpha L + \alpha \beta \Lambda^2) e_u^{(k+1/2)} &= (1 - \alpha) D + a U) e_u^{(k)} + \alpha \left( \Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right), \\
(D - \alpha L + \alpha \beta \Lambda^2) e_u^{(k+1)} &= (1 - \alpha) D + a U) e_u^{(k+1)} + \alpha \left( \Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right).
\end{align*}
\]

Noting that \(U = L^T\), we write

\[
\begin{align*}
D - \alpha L + \alpha \beta \Lambda^2 &= (2 - \alpha) D + a \beta \Lambda^2 - (1 - \alpha) D + a L), \\
D - \alpha L^T + \alpha \beta \Lambda^2 &= (2 - \alpha) D + a \beta \Lambda^2 - (1 - \alpha) D + a L^T).
\end{align*}
\]

Therefore, the Eq. (3.6) takes form

\[
\begin{align*}
\left[ (2 - \alpha) D + a \beta \Lambda^2 \right] e_u^{(k+1/2)} &= (1 - \alpha) D + a L) e_u^{(k)} + \alpha \left( \Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right), \\
\left[ (2 - \alpha) D + a \beta \Lambda^2 \right] e_u^{(k+1)} &= (1 - \alpha) D + a L^T) e_u^{(k+1)} + \alpha \left( \Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right).
\end{align*}
\]

Since 0 < α < 2, the matrix \((2 - \alpha) D + a \beta \Lambda^2\) is invertible. Multiplying (3.7) by the corresponding inverse matrix gives

\[
\begin{align*}
\left[ I - (2 - \alpha) D + a \beta \Lambda^2 \right]^{-1} \left[ (2 - \alpha) D + a L) e_u^{(k+1/2)} \right] &= \alpha \left[ \Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right].
\end{align*}
\]
Combining (3.8) and (3.3), we have

\[
(2 - \alpha)D + \alpha \beta \Lambda^2 \left[ ((1 - \alpha)D + aU) e_u^{(k)} + \alpha \left( \Lambda e_\Lambda^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) \right],
\]

or equivalently,

\[
\Lambda e_\Lambda^{(k+1)} = \beta \Lambda^2 e_w^{(k)}.
\]

Using the notation

\[
W = ((1 - \alpha)D + aL) \left( (2 - \alpha)D + \alpha \beta \Lambda^2 \right)^{-1} (1 - \alpha)D + aL^T,
\]

we write

\[
\left[ ((1 - \alpha)D + aL) - W \right] e_u^{(k+1)} = ((1 - \alpha)D + aL) e_u^{(k+1/2)} - We_u^{(k)}.
\]

This and the second equation in (3.6) give

\[
\left[ ((1 - \alpha)D + aL) - W \right] e_u^{(k+1)} = \left( D - aU + \alpha \beta \Lambda^2 e_u^{(k)} - \alpha \left( \Lambda e_\Lambda^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) - We_u^{(k)}
\]

or

\[
(\alpha \beta \Lambda^2 + \alpha A + W) e_u^{(k+1)} = We_u^{(k)} + \alpha \left( \Lambda e_\Lambda^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right).
\]

Multiplying the last equation by \( e_u^{(k+1)} \) yields

\[
\frac{2\beta}{\alpha} \left( e_u^{(k+1)}, \left( \alpha \beta \Lambda^2 + \alpha A + W \right) e_u^{(k+1)} \right) = \frac{2\beta}{\alpha} \left( e_u^{(k+1)}, We_u^{(k)} + \alpha \left( \Lambda e_\Lambda^{(k)} + \beta \Lambda^2 e_w^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) \right).
\]

Combining (3.8) and (3.3), we have

\[
\|e_\Lambda^{(k)}\|^2 - \|e_\Lambda^{(k+1)}\|^2 \geq \beta^2 \left( \|\Lambda e_w^{(k+1)}\|^2 - 2\left( e_u^{(k+1)}, \Lambda^2 e_w^{(k)} \right) \right) - \frac{2\beta}{\alpha} \left( e_u^{(k+1)}, We_u^{(k)} + \alpha \left[ - \phi(u^{(k)}) + \phi(u^*) \right) \right).
\]
Therefore,
\[
\|e_{\lambda}^{(k)}\|^2 - \|e_{\lambda}^{(k+1)}\|^2 + \beta^2 \left(\|\Lambda e_{w}^{(k)}\|^2 - \|\Lambda e_{w}^{(k+1)}\|^2\right)
\geq \beta^2 \|\Lambda (e_{w}^{(k)} - e_{w}^{(k+1)})\|^2 + \frac{2\beta}{\alpha} \left( e_{u}^{(k+1)}, (\alpha \Lambda + W)e_{u}^{(k+1)}\right)
\]
\[
- \frac{2\beta}{\alpha} \left( e_{u}^{(k+1)}, W e_{u}^{(k)} + \alpha \left[-D_{\phi}\right] e_{u}^{(k)}\right).
\]
Set
\[
V := \left((2 - \alpha)D + \alpha \beta \Lambda^2\right)^{-1/2}\left((1 - \alpha)D + \alpha L^T\right).
\]
Since \(0 < \alpha < 2\) and \(\phi'(x) \geq 0, i = 1, 2, \ldots, n\) for all \(x \in \mathbb{R}\), the Cauchy-Schwartz inequality implies
\[
- \frac{2\beta}{\alpha} \|e_{u}^{(k+1)}, W e_{u}^{(k)}\| \geq - \frac{2\beta}{\alpha} \|Ve_{u}^{(k+1)}\| \|Ve_{u}^{(k)}\|
\]
\[
\geq - \frac{\beta}{\alpha} \|Ve_{u}^{(k+1)}\|^2 - \frac{\beta}{\alpha} \|Ve_{u}^{(k)}\|^2 = - \frac{\beta}{\alpha} \left( e_{u}^{(k+1)}, Ve_{u}^{(k+1)}\right) - \frac{\beta}{\alpha} \|Ve_{u}^{(k)}\|^2.
\]
Finally we get
\[
\|e_{\lambda}^{(k)}\|^2 - \|e_{\lambda}^{(k+1)}\|^2 + \beta^2 \left(\|\Lambda e_{w}^{(k)}\|^2 - \|\Lambda e_{w}^{(k+1)}\|^2\right)
\]
\[
+ \frac{\beta}{\alpha} \|Ve_{u}^{(k)}\|^2 - \frac{\beta}{\alpha} \|Ve_{u}^{(k+1)}\|^2 + \beta d_{\phi} \left(\|e_{u}^{(k)}\|^2 - \|e_{u}^{(k+1)}\|^2\right)
\]
\[
\geq \beta^2 \|\Lambda (e_{w}^{(k)} - e_{w}^{(k+1)})\|^2 + 2\beta \left( e_{u}^{(k+1)}, (\Lambda - d_{\phi} I)e_{u}^{(k+1)}\right).
\]
Similar to the considerations in Theorem 3.1, we obtain that Algorithm 2.2 converges. \(\frak{Q}\)

**Theorem 3.3.** Under the conditions of Theorem 3.1 and \(0 < \alpha < 2\), the sequence \((u^{(k)}, w^{(k)})\) generated by Algorithm 2.3 converges to the solution \((u^*, w^*)\) of the problem (2.4).

**Proof:** Combining the first equation in (3.1) and (2.10) yields
\[
(\tilde{D} - \alpha L)e_{u}^{(k+1/2)} = ((1 - \alpha)\tilde{D} + \alpha U)e_{u}^{(k)} + \alpha \left(\Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*)\right),
\]
\[
(\tilde{D} - \alpha U)e_{u}^{(k+1)} = ((1 - \alpha)\tilde{D} + \alpha L)e_{u}^{(k+1/2)} + \alpha \left(\Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*)\right).
\]
Since \(U = L^T\), we have
\[
\tilde{D} - \alpha L = (2 - \alpha)\tilde{D} - ((1 - \alpha)\tilde{D} + \alpha L),
\]
\[
\tilde{D} - \alpha L^T = (2 - \alpha)\tilde{D} - ((1 - \alpha)\tilde{D} + \alpha L^T).
\]
Thus, the Eq. (3.9) takes form
\[
[(2 - \alpha)\tilde{D} - ((1 - \alpha)\tilde{D} + \alpha L)]e_{u}^{(k+1/2)}
\]
\[
= ((1 - \alpha)\tilde{D} + \alpha U)e_{u}^{(k)} + \alpha \left(\Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*)\right),
\]
\[
[(2 - \alpha)\tilde{D} - ((1 - \alpha)\tilde{D} + \alpha L^T)]e_{u}^{(k+1)}
\]
\[
= ((1 - \alpha)\tilde{D} + \alpha L)e_{u}^{(k+1/2)} + \alpha \left(\Lambda e_{\lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*)\right).
Combining (3.10) and (3.3) yields

$$\left[ I - ((2 - \alpha)\bar{D})^{-1}((1 - \alpha)\bar{D} + aL) \right] e_u^{(k+1/2)}$$

$$= ((2 - \alpha)\bar{D})^{-1} \left[ ((1 - \alpha)\bar{D} + aU)e_u^{(k)} + \alpha \left( \Lambda e_{\Lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) \right],$$

$$= ((2 - \alpha)\bar{D})^{-1} \left[ ((1 - \alpha)\bar{D} + aL)e_u^{(k+1/2)} + \alpha \left( \Lambda e_{\Lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) \right].$$

Subtracting the first equation from the second one gives

$$\left[ I - ((2 - \alpha)\bar{D})^{-1}((1 - \alpha)\bar{D} + aL^T) \right] e_u^{(k+1)}$$

$$= e_u^{(k+1/2)} - ((2 - \alpha)\bar{D})^{-1}((1 - \alpha)\bar{D} + aU)e_u^{(k)}$$

or

$$((1 - \alpha)\bar{D} + aL)\left[ I - ((2 - \alpha)\bar{D})^{-1}((1 - \alpha)\bar{D} + aL^T) \right] e_u^{(k+1)}$$

$$= ((1 - \alpha)\bar{D} + aL)e_u^{(k+1/2)} - ((2 - \alpha)\bar{D})^{-1}((1 - \alpha)\bar{D} + aU)e_u^{(k)}.$$

Setting

$$W = ((1 - \alpha)\bar{D} + aL)((2 - \alpha)\bar{D})^{-1}((1 - \alpha)\bar{D} + aL^T),$$

we can write

$$\left[ ((1 - \alpha)\bar{D} + aL) - W \right] e_u^{(k+1)} = ((1 - \alpha)\bar{D} + aL)e_u^{(k+1/2)} - We_u^{(k)}.$$

Combining this with the second equation in (3.9), we have

$$\left[ ((1 - \alpha)\bar{D} + aL) - W \right] e_u^{(k+1)}$$

$$= (\bar{D} - aU)e_u^{(k+1)} - \alpha \left( \Lambda e_{\Lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) - We_u^{(k)},$$

or

$$\left( \alpha \beta \Lambda^2 + aA + W \right) e_u^{(k+1)} = We_u^{(k)} + \alpha \left( \Lambda e_{\Lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right).$$

Multiplying it by $e_u^{(k+1)}$ gives

$$\frac{2\beta}{\alpha} \left< e_u^{(k+1)}, (\alpha \beta \Lambda^2 + aA + W)e_u^{(k+1)} \right>$$

$$= \frac{2\beta}{\alpha} \left< e_u^{(k+1)}, We_u^{(k)} + \alpha \left( \Lambda e_{\Lambda}^{(k)} + \beta \Lambda^2 e_{w}^{(k)} - \phi(u^{(k)}) + \phi(u^*) \right) \right>.$$  \hspace{1cm} (3.10)

Combining (3.10) and (3.3) yields

$$\|e_{\Lambda}^{(k)}\|^2 - \|e_{\Lambda}^{(k)}\|^2 \geq \beta^2 \left( \|\Lambda e_{\Lambda}^{(k)}\|^2 - 2 \left< e_{\Lambda}^{(k+1)}, \Lambda^2 e_{w}^{(k)} \right> \right) - \beta^2 \|e_{\Lambda}^{(k+1)}\|^2$$

$$+ \frac{2\beta}{\alpha} \left< e_u^{(k+1)}, (\alpha \beta \Lambda^2 + aA + W)e_u^{(k+1)} \right>$$

$$- \frac{2\beta}{\alpha} \left< e_u^{(k+1)}, We_u^{(k)} + \alpha \left[ -\phi(u^{(k)}) + \phi(u^*) \right] \right>.$$
Therefore,
\[
\|e^{(k)}_\lambda\|^2 - \|e^{(k+1)}_\lambda\|^2 + \beta^2 \left(\|\Lambda e^{(k)}_w\|^2 - \|\Lambda e^{(k+1)}_w\|^2\right) \\
\geq \beta^2 \|\Lambda (e^{(k)}_w - e^{(k+1)}_w)\|^2 + \frac{2\beta}{\alpha} \left(e^{(k+1)}_u, (\alpha A + W)e^{(k+1)}_u\right) \\
- \frac{2\beta}{\alpha} \left(e^{(k+1)}_u, We^{(k)}_u + \alpha \left[-\phi^{(k)}\right] e^{(k)}_u\right).
\]

Setting
\[
V := \left((2 - \alpha)\bar{D}\right)^{-1/2}((1 - \alpha)\bar{D} + \alpha L^T)
\]
and using the Cauchy-Schwartz inequality gives
\[
- \frac{2\beta}{\alpha} \left(e^{(k+1)}_u, We^{(k)}_u\right) \geq - \frac{2\beta}{\alpha} \|We^{(k+1)}_u\| \|Ve^{(k)}_u\| \\
\geq - \frac{\beta}{\alpha} \|Ve^{(k+1)}_u\|^2 - \frac{\beta}{\alpha} \|Ve^{(k)}_u\|^2 = - \frac{\beta}{\alpha} \left(e^{(k+1)}_u, We^{(k)}_u\right) - \frac{\beta}{\alpha} \|Ve^{(k)}_u\|^2.
\]

Finally we get
\[
\|e^{(k)}_\lambda\|^2 - \|e^{(k+1)}_\lambda\|^2 + \beta^2 \left(\|\Lambda e^{(k)}_w\|^2 - \|\Lambda e^{(k+1)}_w\|^2\right) \\
+ \frac{\beta}{\alpha} \|Ve^{(k+1)}_u\|^2 - \frac{\beta}{\alpha} \|Ve^{(k)}_u\|^2 + \beta d_\phi \left(\|e^{(k)}_u\|^2 - \|e^{(k+1)}_u\|^2\right) \\
\geq \beta^2 \|\Lambda (e^{(k)}_w - e^{(k+1)}_w)\|^2 + 2\beta \left(e^{(k+1)}_u, (A - d_\phi I)e^{(k+1)}_u\right).
\]

Considerations similar to the ones in Theorem 3.1 show that Algorithm 2.3 converges. □

4. Numerical Experiments

As is shown in [7, 25, 30, 31, 45, 53], the modulus-based matrix splitting iteration methods perform better than certain popular iteration methods such as projected type methods [12] and modified modulus methods [15] applied to LCPs, and they are more efficient than the Noor fixed point approach [38] and the Fischer-Burmeister semismooth Newton method [13] applied to NCPs.

We provide a brief review of the modulus-based Jacobi method (MJ), modulus-based Gauss-Seidel method (MGS), modulus-based successive overrelaxation method (MSOR), modulus-based accelerated overrelaxation method (MAOR) [25, 45] and the inexact alternating direction method of multipliers (IADM) [24] for solving NCPs and compare them with the methods studied in this paper. Abbreviations of all testing methods are listed in Table 1.

We note that the modulus-based matrix splitting iteration methods are based on a splitting $A = F - G$ and the iterations used are
\[
(\Omega + F)x^{(k+1)} = Gx^{(k)} + (\Omega - A)x^{(k)} - \gamma \left[b + \phi \left(\frac{1}{\gamma}(|x^{(k)}| + x^{(k)})\right)\right]
\]
Table 1: Method abbreviations.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DADM</td>
<td>Proposed Algorithm 1</td>
</tr>
<tr>
<td>SADM</td>
<td>Proposed Algorithm 2</td>
</tr>
<tr>
<td>MSADM</td>
<td>Proposed Algorithm 3</td>
</tr>
<tr>
<td>IADM</td>
<td>The inexact alternating direction method of multipliers</td>
</tr>
<tr>
<td>MAOR</td>
<td>The modulus-based accelerated overrelaxation method</td>
</tr>
<tr>
<td>MSOR</td>
<td>The modulus-based successive overrelaxation method</td>
</tr>
<tr>
<td>MGS</td>
<td>The modulus-based Gauss-Seidel method</td>
</tr>
<tr>
<td>MJ</td>
<td>The modulus-based Jacobi method</td>
</tr>
</tbody>
</table>

with

\[ u^{(k+1)} = \frac{1}{\gamma} \left( |x^{(k+1)}| + x^{(k+1)} \right). \]

We also note that:

(a) If \( F = D, G = L + U \), we obtain the modulus-based Jacobi method.

(b) If \( F = D - L, G = U \), we obtain the modulus-based Gauss-Seidel method.

(c) If \( F = (1/\alpha)D - L, G = (1/\alpha - 1)D + U \), we obtain the modulus-based successive overrelaxation method.

(d) If \( F = (1/\alpha)(D - \beta L), G = (1/\alpha)(1 - \alpha)D + (\alpha - \beta)L + \alpha U \), we obtain the modulus-based accelerated overrelaxation method.

In some practical applications, the discretisation matrix can be represented in the form \( A = H + V \), where \( H \) and \( V \) correspond to the discretisation matrices of the differential operator in each spatial direction \([24, 52]\). Furthermore, both \( H \) and \( V \) are symmetric positive definite, large and sparse. The iteration of IADM is given as follows:

\[
\begin{align*}
(H + \beta \mu^2 I)u^{(k+1/2)} &= -Vu^{(k)} + (\mu \lambda(k) + \beta \mu^2 w^{(k)} - \phi(u^{(k)})), \\
(V + \beta \mu^2 I)u^{(k+1)} &= -Hu^{(k+1/2)} + (\mu \lambda(k) + \beta \mu^2 w^{(k)} - \phi(u^{(k)})).
\end{align*}
\]

(4.1)

**Algorithm 4.1** Inexact alternating direction method of multipliers (IADM)

**Input:** \( H, V \) with \( A = H + V \), \( \phi \), \( \mu \) and \( \beta > 0 \); give initial points \( u^{(0)}, w^{(0)} \) and \( \lambda^{(0)} \).

**While** not converged, **do**

- Generate \( u^{(k+1)} \) according to (4.1);
- Generate \( w^{(k+1)} \) according to \( w^{(k+1)} = (u^{(k+1)} - \mu^{-1} \lambda^{(k)}/\beta)_{+} \);
- Update \( \lambda^{(k)} \) according to \( \lambda^{(k+1)} = \lambda^{(k)} + \beta \mu (w^{(k+1)} - u^{(k+1)}) \);

**End While**
We next compare our methods with MJ, MGS, MSOR, MAOR and IADM. Let
\[
\text{RES}(u^{(k)}) := \| \min (Au^{(k)} + \phi(u^{(k)}), u^{(k)}) \|,
\]
where the minimum is taken componentwise. We set \( \mu = 1 \) in DADM, SADM and MSADM. All other parameters are obtained experimentally by minimising the corresponding iteration steps. Moreover, \( \gamma = 2 \) and all initial vectors are \((0, 0, \ldots, 0)^T \in \mathbb{R}^n\). Let IT denote the number of iterations and CPU the elapsed CPU time in seconds. Finally, for all methods the stopping criteria are \( \text{RES} \leq 10^{-6} \) or \( \text{IT} > 10000 \).

**Example 4.1** (cf. Refs. [25, 33, 45, 54, 56, 59]). Consider \( A = \hat{A} + \sigma I \in \mathbb{R}^{n \times n} \) and
\[
b = -Az - (\psi_1(z_1), \psi_2(z_2), \ldots, \psi_n(z_n))^T \in \mathbb{R}^n,
\]
where the matrix \( \hat{A} \in \mathbb{R}^{n \times n} \) has the form
\[
\hat{A} = \begin{pmatrix} S & -I & 0 & \cdots & 0 & 0 \\
-I & S & -I & \cdots & 0 & 0 \\
0 & -I & S & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & -I & S & -I \\
0 & 0 & \cdots & 0 & -I & S \end{pmatrix}
\]
with the identity matrix \( I \in \mathbb{R}^{m \times m} \), the tri-diagonal \( m \times m \) matrix
\[
S = \begin{pmatrix} 4 & -1 & 0 & \cdots & 0 & 0 \\
-1 & 4 & -1 & \cdots & 0 & 0 \\
0 & -1 & 4 & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & -1 & 4 & -1 \\
0 & 0 & \cdots & 0 & -1 & 4 \end{pmatrix},
\]
and with
\[
z = [1, 2, 1, 2, \ldots, 1, 2, \ldots]^T \in \mathbb{R}^n.
\]

For experiments, we choose two options \( \sigma = 0, \psi(z) = \arctan(z) \) and \( \sigma = 4, \psi(z) = \ln(1 + e^z) \).

Tables 2 and 3 show the corresponding numerical results. We note that the new DADM requires the least number of iterations. However, MSADM is the fastest at all sizes in Example 4.1 and it can be almost 2 times faster than MAOR. On the other hand, although SADM is slightly slower than MSADM, it is still faster and takes fewer iterations than the existing methods at all time. If \( \sigma = 0, \psi(z) = \arctan(z) \), the matrix \( S := A - d \phi I \) is not positive definite. Nevertheless, the methods proposed still work.
Table 2: Example 4.1. Numerical results, $\sigma = 0$, $\psi(z) = \arctan(z)$.

<table>
<thead>
<tr>
<th></th>
<th>300</th>
<th></th>
<th>500</th>
<th></th>
<th>700</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>IT</td>
<td>RES</td>
<td>CPU</td>
<td>IT</td>
<td>RES</td>
</tr>
<tr>
<td>DADM</td>
<td>1.67</td>
<td>11</td>
<td>3.47e-07</td>
<td>4.90</td>
<td>11</td>
<td>5.92e-07</td>
</tr>
<tr>
<td>SADM</td>
<td>0.09</td>
<td>17</td>
<td>4.79e-07</td>
<td>0.38</td>
<td>17</td>
<td>6.09e-07</td>
</tr>
<tr>
<td>MSADM</td>
<td>0.09</td>
<td>17</td>
<td>5.35e-07</td>
<td>0.38</td>
<td>17</td>
<td>6.13e-07</td>
</tr>
<tr>
<td>IADM</td>
<td>2.37</td>
<td>42</td>
<td>6.53e-07</td>
<td>9.88</td>
<td>43</td>
<td>6.82e-07</td>
</tr>
<tr>
<td>MAOR</td>
<td>0.17</td>
<td>39</td>
<td>9.55e-07</td>
<td>0.75</td>
<td>40</td>
<td>7.43e-07</td>
</tr>
<tr>
<td>MSOR</td>
<td>0.48</td>
<td>121</td>
<td>0.38e-07</td>
<td>2.15</td>
<td>124</td>
<td>0.76e-07</td>
</tr>
<tr>
<td>MGS</td>
<td>0.49</td>
<td>121</td>
<td>9.96e-07</td>
<td>2.15</td>
<td>125</td>
<td>9.33e-07</td>
</tr>
<tr>
<td>MJ</td>
<td>0.80</td>
<td>219</td>
<td>9.81e-07</td>
<td>3.46</td>
<td>226</td>
<td>9.51e-07</td>
</tr>
</tbody>
</table>

Table 3: Example 4.1. Numerical results, $\sigma = 4$, $\psi(z) = \ln(1 + e^z)$.

<table>
<thead>
<tr>
<th></th>
<th>300</th>
<th></th>
<th>500</th>
<th></th>
<th>700</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>IT</td>
<td>RES</td>
<td>CPU</td>
<td>IT</td>
<td>RES</td>
</tr>
<tr>
<td>DADM</td>
<td>0.87</td>
<td>6</td>
<td>1.56e-08</td>
<td>3.38</td>
<td>6</td>
<td>2.60e-08</td>
</tr>
<tr>
<td>SADM</td>
<td>0.04</td>
<td>6</td>
<td>6.76e-07</td>
<td>0.16</td>
<td>6</td>
<td>4.29e-07</td>
</tr>
<tr>
<td>MSADM</td>
<td>0.04</td>
<td>6</td>
<td>3.99e-07</td>
<td>0.16</td>
<td>6</td>
<td>3.79e-07</td>
</tr>
<tr>
<td>IADM</td>
<td>1.46</td>
<td>26</td>
<td>8.41e-07</td>
<td>5.72</td>
<td>27</td>
<td>8.02e-07</td>
</tr>
<tr>
<td>MAOR</td>
<td>0.07</td>
<td>13</td>
<td>7.78e-07</td>
<td>0.29</td>
<td>13</td>
<td>8.29e-07</td>
</tr>
<tr>
<td>MSOR</td>
<td>0.10</td>
<td>19</td>
<td>9.98e-07</td>
<td>0.42</td>
<td>20</td>
<td>5.42e-07</td>
</tr>
<tr>
<td>MGS</td>
<td>0.10</td>
<td>19</td>
<td>9.85e-07</td>
<td>0.45</td>
<td>20</td>
<td>6.95e-07</td>
</tr>
<tr>
<td>MJ</td>
<td>0.12</td>
<td>26</td>
<td>5.41e-07</td>
<td>0.54</td>
<td>26</td>
<td>8.91e-07</td>
</tr>
</tbody>
</table>

Example 4.2 (cf. Harker & Pang [24]). Let $M$ be a given positive integer, $m = 2^M - 1$, $n = m^2$, $h = 1/m + 1$, and $A = H + V \in \mathbb{R}^{n \times n}$, where $H = I \otimes V_1$, $V = V_1 \otimes I$.

\[ V_1 = \frac{1}{h^2} \times \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \in \mathbb{R}^{m \times m}, \]

$I \in \mathbb{R}^{m \times m}$ is the unit matrix and $\otimes$ the Kronecker product. The nonlinear term is $\phi(u) = u - \sin(u) + b$, in which the entries of $b$ are ranging from 0 to 10. More precisely,

\[ b = -\left[0, h_1, 2h_1, \ldots, 10, 0, h_1, 2h_1, \ldots, 10, \ldots, \ldots, 0, h_1, 2h_1, \ldots, 10\right] \in \mathbb{R}^n, \]

where $h_1 = 10/(m-1)$. 
We test the problem for different mesh-step sizes $h$. Table 4, where the symbol '-' indicates that the number of iterations exceeds 10000, shows that the proposed DADM is the fastest method and it requires the smallest number of iterations in all cases. MSADM and SADM are still faster than MJ, MGS, MSOR and IADM despite longer computation time than MAOR. Besides, DADM can be at least 30 times faster than the existing methods at $M=9$, which is a huge advantage. We also note that MJ, MGS, MSOR were not able to determine the solution within 10000 iterations, and all methods, except of DADM, required more iterations to converge than in Example 4.1. It is related to the difference of vectors $b$, nonlinear terms and the form of the matrix $A$. Our experiments demonstrate that the structure of $A$ is the major factor influencing the behavior of the methods. In contrast to Example 4.1, the problem considered arises from the discretisation of a free boundary problem. Therefore, one needs to divide the matrix $A$ by $h^2$ here and this is the main reason why more iterations than in Example 4.1 are needed.

Table 4: Example 4.2. Numerical results.

<table>
<thead>
<tr>
<th>M</th>
<th>7</th>
<th>CPU</th>
<th>IT</th>
<th>RES</th>
<th>8</th>
<th>CPU</th>
<th>IT</th>
<th>RES</th>
<th>9</th>
<th>CPU</th>
<th>IT</th>
<th>RES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DADM</td>
<td>0.06</td>
<td>3</td>
<td>7.8E-07</td>
<td>0.32</td>
<td>3</td>
<td>5.74E-07</td>
<td>1.51</td>
<td>3</td>
<td>7.43E-07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SADM</td>
<td>0.46</td>
<td>636</td>
<td>9.73E-07</td>
<td>4.54</td>
<td>1329</td>
<td>9.85E-07</td>
<td>67.69</td>
<td>2776</td>
<td>9.99E-07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSADM</td>
<td>0.45</td>
<td>636</td>
<td>9.73E-07</td>
<td>4.54</td>
<td>1329</td>
<td>9.84E-07</td>
<td>65.06</td>
<td>2776</td>
<td>9.98E-07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IADM</td>
<td>3.78</td>
<td>624</td>
<td>9.80E-07</td>
<td>48.92</td>
<td>1257</td>
<td>9.88E-07</td>
<td>577.95</td>
<td>2551</td>
<td>9.98E-07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAOR</td>
<td>0.34</td>
<td>541</td>
<td>9.86E-07</td>
<td>3.23</td>
<td>1161</td>
<td>9.90E-07</td>
<td>49.52</td>
<td>2386</td>
<td>1.00E-06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSOR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MGS</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MJ</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Summarising we note the following issues.

1. All new methods considered here converge in all cases. They have advantage in different situations — cf. the discussion above. In particular, for large size problems they are more efficient than the existing MJ, MGS, MSOR, MAOR and IADM methods.

2. In practical implementation, we can consider a small size problem first to decide which one of new methods is the fastest and use it to solve the corresponding large size problem.

5. Conclusions

In order to find more efficient and feasible ways to solve symmetric positive definite weakly nonlinear complementarity problems, we improve inexact alternating direction methods. We prove the convergence of the methods. Numerical experiments show that the methods proposed have a number of advantages over similar existing methods. They
are more effective, faster, and provide more accurate results in a shorter time, especially for large size problems.

Acknowledgments

This work is funded by the University of Macau (File No. MYRG2018-00047-FST), by the Major Projects of the Guangdong Education Department for Foundation Research and Applied Research (No. 2018KZDXM065), by the Science and Technology Development Fund, Macau SAR (File No. 0005/2019/A) and by the Science Foundation of Shaoguan University (No. SZ2020KJ01).

References


