Research Article

Newton-PGSS and Its Improvement Method for Solving Nonlinear Systems with Saddle Point Jacobian Matrices

Yao Xiao $^{\dagger}$, Qingbiao Wu $^{\ddagger}$, and Yuanyuan Zhang

Department of Mathematics, Zhejiang University, Hangzhou 310027, Zhejiang, China

Correspondence should be addressed to Qingbiao Wu; qbwu@zju.edu.cn

Received 23 November 2020; Revised 11 January 2021; Accepted 21 January 2021; Published 16 February 2021

Copyright © 2021 Yao Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

In this paper, we will explore effective and convenient methods for solving nonlinear nonsymmetric saddle-point problem:

\[ F(x) = 0, \]

where \( F: \mathbb{D} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a continuous differentiable nonlinear function and the function \( F = (F_1, \ldots, F_m)^T \) with \( F_i = F_i(x), i = 1, 2, \ldots \), and \( x = (x_1, \ldots, x_{n+m}) \) is defined on an open convex subset of \((n+m)\)-dimensional real linear space \( \mathbb{R}^m \). Moreover, the Jacobian matrix \( F'(x) \) is large, sparse, and nonsymmetric saddle point with the form

\[ F'(x) = \begin{pmatrix} A(x) & B(x) \\ -B^T(x) & 0 \end{pmatrix}, \]

where \( A(x) \in \mathbb{R}^{m \times n} \) is a real positive definite matrix and \( B(x) \in \mathbb{R}^{m \times n} \) is a full-column rank matrix \((m < n)\). This kind of large sparse nonsymmetric saddle-point nonlinear systems (1) always arises in many scientific and engineering computing areas, such as elastomechanics equations and Stokes equation. Some of them have not been solved analytically, so we can only explore the method to obtain the numerical simulation at our utmost.

In the past, researchers have developed some methods to solve nonlinear function [1–10]. In these methods, the most typical and popular method for solving the nonlinear system (1) is the Newton method. The principle of solving nonlinear equations by the Newton method is very simple. In each step, we expand the nonlinear equation at \( x_k \) by Taylor expansion and take its linear part to construct the approximate equation of the nonlinear equation. Then, we calculate the zero point of the approximate equation as the next iteration point, and it is represented as follows:

\[ x_{k+1} = x_k - F'(x_k)^{-1}F(x_k), \quad k = 0, 1, \ldots \]  (3)

The sequence \( x_k \) calculated by this iteration will converge to the numerical solution eventually as \( k \rightarrow +\infty \) under certain conditions. We know that an excellent algorithm is not only accurate but also efficient. When the dimension \( n \) is large, the cost of each step of the traditional Newton algorithm is very expensive. The reason for this phenomenon is that, at each iterative step, a linear system
must be exactly and accurately solved. We hope to give up a little bit of "precision" in exchange for greater "efficiency." This idea led to the development of inexact Newton methods which were first proposed by Dembo et al. [11]. In recent decades, the inexact Newton method has been extensively studied and applied in many fields. The linear equation (4) can be solved efficiently by some methods which will discard some precision, but the calculation amount and time will be greatly reduced. In addition, we know that the traditional Newton method is second-order convergence, and increasing the order of convergence can make the algorithm converge to the exact solution faster. Therefore, we consider improving the Newton method to improve the order of convergence. Next, we introduce the traditional Newton method and the improved Newton method. In the inexact Newton methods, the termination condition of the Newton equation (4) is

$$\|F'(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\|, \quad k \geq 0,$$

where $s_k = x_{k+1} - x_k$ is obtained by applying some linear iterative methods. The inexact Newton methods usually have the unified form as shown in Algorithm 1.

Here, $F'(x_k)$ is the Jacobian matrix and $\eta_k \in [0, 1]$ is commonly called forcing term which is used to control the level of accuracy. The algorithm mentioned above has R-order of convergence two at least. The researchers present the modified Newton iteration to improve convergence order as shown in Algorithm 2.

From what is mentioned above, inexact-modified Newton methods only need to calculate $F'(x_k)^{-1}$ once per $m$ step and have less computation compared with inexact-modified Newton methods. This kind of method has R-order of convergence $m+1$ at least as the outer iteration and the PGSS iteration method as the inner iteration. In this paper, we can establish the modified Newton-PGSS as $m=2$.

The inexact Newton methods consist of two parts: inner iteration and outer iteration. The outer iteration is the Newton method, which is used to solve nonlinear problems, and each iteration has to solve a linear equation in order to generate the sequence $\{x_k\}$. Linear iterative methods, such as the classical splitting methods or the modern Krylov subspace methods [12, 13], are applied inside the Newton methods to solve the Newton equations approximately. A significant advantage of such inner-outer iterations is that one can reduce the inverse of the Jacobian matrix storage and calculation of each step, so as to improve the operation efficiency. Therefore, this kind of inner-outer iterative methods has been widely studied. Newton–Krylov subspace [3] methods which utilize the Krylov subspace iteration methods as the inner iterations have been effectively and successfully used in many fields, see [14–16].

By introducing the inexact Newton method [1–4, 7, 8], we know that the efficiency of the inner iteration will affect the efficiency of the whole algorithm. Thus, we want to explore the excellent inner iteration to obtain efficient inner-outer iterative methods. In other words, efficient linear iteration should be employed to solve the Newton equation (4) with real nonsymmetric saddle-point Jacobian matrix. There are many ways to solve the saddle point linear problem [3, 17–25]. Recently, Cao et al. [26–29] proposed a method which is based on the shift-splitting iteration method presented by Bai et al. [30] to solve the saddle-point problem. This method is more efficient than other algorithms such as the Uzawa-type iteration methods, the successive over-relaxation (SOR-like) iteration methods [31, 32], and the Hermitian and skew-Hermitian splitting (HSS) iteration methods [33–35]. In addition, the PGSS iteration method is convergent unconditionally and the preconditioner generated by it is also very excellent [26]. When applying the PGSS method for solving complex linear system, at each iterative step, it needs to solve single linear subsystem with their coefficient matrices being the $\mathcal{A}_{PGSS}$ one $(1/2)(\Omega + \mathcal{A})$. Furthermore, in order to increase the efficiency of algorithm, we optimized the outer iteration and then we propose modified Newton-PGSS method to solve the saddle problems. Because there was no Newton method to solve the saddle point system problem, we compare the Newton-PGSS method with the traditional methods, for example, the Newton-RHSS method [31, 36, 37].

The organization of the paper is as follows. In Section 3, we introduce the Newton-PGSS method. In Sections 4 and 5, we offer the convergence properties of this method. We establish local convergence theorem and semilocal convergence properties under some proper hypothesis for the Newton-PGSS method, respectively. We show the modified Newton-PGSS method in Section 6. Numerical examples are presented to confirm the efficiency of our new method in Section 7. Finally, in Section 8, some brief conclusions are given.

### 2. Preliminaries

First of all, we review the PGSS method [26] for solving large sparse nonsymmetric saddle-point linear system:

$$\mathcal{A}x = b, \quad \mathcal{A} \in \mathbb{R}^{(n+m) \times (n+m)}, x, b \in \mathbb{R}^n,$$

where $\mathcal{A} \in \mathbb{R}^{(n+m) \times (n+m)}$ is a real nonsymmetric saddle-point matrix.

**The PGSS Iteration Method** [27]. Given an initial guess $x_0 \in \mathbb{R}^{nm}$, compute $x_{k+1}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{x_k\}$ satisfies the stopping criterion:

**Algorithm 1:** Inexact Newton methods.

1. Let the initial guess $x_0$ be given.
2. For $k = 0$ until "convergence" do:
   - First of all, we review the PGSS method [26] for solving large sparse nonsymmetric saddle-point linear system:
   - \[ \mathcal{A}x = b, \quad \mathcal{A} \in \mathbb{R}^{(n+m) \times (n+m)}, x, b \in \mathbb{R}^n, \]
   - where $\mathcal{A} \in \mathbb{R}^{(n+m) \times (n+m)}$ is a real nonsymmetric saddle-point matrix.
   - The PGSS Iteration Method [27]. Given an initial guess $x_0 \in \mathbb{R}^{nm}$, compute $x_{k+1}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{x_k\}$ satisfies the stopping criterion:
(1) Let the initial guess $x_0$ be given.
(2) For $k = 0$ until "convergence" do:

$$
\begin{align*}
\frac{1}{2} (\Omega + \alpha f) x_{k+1} &= \frac{1}{2} (\Omega - \alpha f) x_k + b, \\
\end{align*}
$$

(7)

where $\Omega$ is a matrix with the form $\begin{pmatrix} \alpha I_1 & 0 \\ 0 & \beta I_2 \end{pmatrix}$, where $I_1$ is a $n \times n$ identity matrix and $I_2$ is a $m \times m$ identity matrix and $\alpha$ and $\beta$ are real numbers greater than 0. We can get $x_{k+1}$ from (7) leading to the PGSS iterative scheme:

$$
\begin{align*}
x_{k+1} &= M_{a,b} x_k + G_{a,b} b, \\
\end{align*}
$$

(8)

where

$$
\begin{align*}
M_{a,b} &= (\Omega + \alpha f)^{-1} (\Omega - \alpha f), \\
G_{a,b} &= (\Omega + \alpha f)^{-1}.
\end{align*}
$$

(9)

Here, $M_{a,b}$ is the iteration matrix of the PGSS iteration method.

**Theorem 1** (see [27]). $\alpha f = b \in \mathbb{R}^{(r \times m) \times (r \times m)}$ is a nonsymmetric saddle-point matrix, $\alpha$ is a nonnegative constant, and $\beta$ is a positive constant. Then, the iteration matrix $M_{a,b}$ of PGSS is

$$
M_{a,b} = (\Omega + \alpha f)^{-1} (\Omega - \alpha f),
$$

(10)

which satisfies

$$
\rho(M_{a,b}) \leq \max_{\lambda \in \lambda(\Omega + \alpha f), \lambda(\Omega - \alpha f)} \frac{1 - \lambda}{1 + \lambda} \leq 1.
$$

(11)

**3. The Newton-PGSS Method**

In this section, we describe an inner-outer iteration method for solving systems of nonlinear equations with complex symmetric Jacobian matrices.

We use Newton methods as outer iteration and apply the PGSS method as the inner solver for the modified Newton method, in other words, the PGSS iteration is employed to solve the following two linear systems:

$$
F'(x_k) d_k = -F(x_k), \quad x_{k+1} = x_k + d_k.
$$

(12)

Then, we get the Newton-PGSS method for solving nonlinear system (1).
(1) Given an initial guess $x_0$, a nonnegative constant $\alpha$, a positive constant $\beta$, and a positive integer sequences $\{t_k\}_{k=0}^{\infty}$.

(2) For $k = 0, 1, \ldots$, until $\|F(x_k)\| \leq \text{tol}\|F(x_0)\|$ do:

(2.1) Set $d_{k,0} = 0$.

(2.2) For $l = 0, 1, \ldots, l_k - 1$, apply algorithm PGSS to the linear system (12):

\[ (\Omega(x_k) + F(x_k))d_{k,l+1} = (\Omega(x_k) - F(x_k))d_{k,l} - F(x_k), \]

and obtain $d_{k,l}$ such that

\[ \|F(x_k) + F(x_k)d_{k,l}\| \leq \eta_k \|F(x_k)\|, \]

for some $\eta_k \in [0, 1)$, where

\[ \Omega(x_k) = \left( \begin{array}{cc} \alpha I_1(x_k) & 0 \\ 0 & \beta I_2(x_k) \end{array} \right). \]

$I_1$ is an $n \times n$ identity matrix and $I_2$ is an $m \times m$ identity matrix.

(2.3) Set

\[ x_{k+1} = x_k + d_{k,l}. \]

obtain the following uniform expressions for $d_{k,l}$,

\[ d_{k,l} = -\sum_{j=0}^{l-1} M_{a,y}(\Omega; x_k) G_{a,y}(\Omega; x_k) F(x_k), \]

where

\[ M_{a,y}(\Omega; x) = (\Omega(x) + F'(x))^{-1} (\Omega(x) - F'(x)), \]

and

\[ G_{a,y}(\Omega; x) = 2(\Omega(x) + F'(x))^{-1}. \]

Then, the Newton-PGSS method can be rewritten as

\[ x_{k+1} = x_k - \sum_{j=0}^{l-1} M_{a,y}(\Omega; x_k) G_{a,y}(\Omega; x_k) F(x_k), \]

$k = 0, 1, 2, \ldots$.

From the definitions of $M_{a,y}(V; x)$ and $G_{a,y}(V; x)$, we can obtain

\[ G_{a,y}(\Omega(x)^{-1} - M_{a,y}(\Omega(x)) = G_{a,y}(\Omega(x)^{-1} - M_{a,y}(\Omega(x)) \]

\[ = (1/2)(\Omega(x) + F'(x)) - (1/2)(\Omega(x) + F'(x))(\Omega(x) + F'(x))^{-1}(\Omega(x) - F'(x)) \]

\[ = (1/2)(\Omega(x) + F'(x)) - (1/2)(\Omega(x) - F'(x)) \]

\[ = F'(x). \]

Then, the Newton-PGSS method can be equivalently expressed as

\[ x_{k+1} = x_k - (I - M_{a,y}(\Omega(x))^k) F'(x) \]

\[ = B_{a,y}(\Omega(x)) - C_{a,y}(\Omega(x), \]

with

\[ B_{a,y}(\Omega(x)) = G_{a,y}(\Omega(x)^{-1}, \]

\[ B_{a,y}(\Omega(x))^{-1} = (I - M_{a,y}(\Omega(x))) F'(x)^{-1}, \]

\[ C_{a,y}(\Omega(x)) = G_{a,y}(V(x)^{-1} M_{a,y}(\Omega(x)), \]

\[ \|A(x) - A(x_*)\| \leq K_1 \|x - x_*\|^p, \]

\[ \|B(x) - B(x_*)\| \leq K_2 \|x - x_*\|^p, \]

with the exponent $p \in (0, 1]$.

**Algorithm 3:** N-PGSS (Newton-PGSS method).

\[ \begin{align*}
\|A(x) - A(x_*)\| & \leq K_1 \|x - x_*\|^p, \\
\|B(x) - B(x_*)\| & \leq K_2 \|x - x_*\|^p,
\end{align*} \]  

Lemma 1. Under Assumption 1, for all $x, y \in \mathbb{R}(x_*, r)$, if

\[ r \in (0, (1/(yK)))^{1/p}, \]

then $F'(x)^{-1}$ exists. And, the following inequalities hold with $K_1 = K_a + K_b$ for all $x, y \in \mathbb{R}(x_*, r)$:

\[ \|F'(x) - F'(x_*)\| \leq K_1 \|x - x_*\|^p, \]

\[ \|F'(x)^{-1}\| \leq \frac{y}{1 - yK} \|x - x_*\|^p, \]

\[ \|F(x)\| \leq \frac{K_1}{1 + p} \|x - x_*\|^{1+p} + 2\eta \|x - x_*\|. \]
Proof

\[
\|F'(x) - F'(x_*)\| \leq \begin{pmatrix}
A(x) - A(x_*) & B(x) - B(x_*) \\
-B'(x) - B'(x_*) & 0
\end{pmatrix}
\]

\[
= \begin{pmatrix}
A(x) - A(x_*) & 0 \\
0 & 0
\end{pmatrix} + \begin{pmatrix}
0 & B(x) - B(x_*) \\
-B'(x) - B'(x_*) & 0
\end{pmatrix}
\]

\[
= \|A(x) - A(x_*)\| + \|B(x) - B(x_*)\| \leq K_\|x - x_*\|^p + K_\|x - x_*\|^p
\]

\[
= K\|x - x_*\|^p.
\]

Since

\[
\|F'(x)^{-1}(F'(x) - F'(x_*)\| \leq \frac{\gamma K\|x - x_*\|^p}{1 - \gamma K\|x - x_*\|^p},
\]

by Banach lemma, \(F(x)^{-1}\) exists and inequality

\[
\|F'(x)^{-1}\| \leq \frac{\gamma}{1 - \gamma K\|x - x_*\|^p}
\]

holds, and

\[
\|\bar{F}(x_*)\| = \begin{pmatrix}
A(x_*) & B(x_*) \\
-B'(x_*) & 0
\end{pmatrix}
\]

\[
\leq \begin{pmatrix}
A(x_*) & 0 \\
0 & 0
\end{pmatrix} + \begin{pmatrix}
0 & B(x_*) \\
-B'(x_*) & 0
\end{pmatrix}
\]

\[
\leq \|A(x_*)\| + \|B(x_*)\| \leq 2\delta.
\]

Moreover, since

\[
F(x) = F(x) - F(x_*) - F'(x_*) (x - x_*) + F'(x_*) (x - x_*)
\]

\[
= \int_0^1 F'(x_*) + (x - x_*) - F'(x_*) dt (x - x_*)
\]

\[
+ F'(x_*) (x - x_*)
\]

it holds that

\[
\|F(x)\| \leq \int_0^1 \|F'(x_*) + (x - x_*) - F'(x_*)\| dt (x - x_*)
\]

\[
+ \|F'(x_*)\| \|x - x_*\| dt (x - x_*)
\]

\[
\leq \int_0^1 k \|x - x_*\|^p + 2\delta \|x - x_*\|
\]

\[
\leq \frac{K}{1 + p} \|x - x_*\|^{1+p} + 2\delta \|x - x_*\|
\]

\[
\]
From the bounded condition, we have
\[
\|B_{a,\beta}(\Omega; x) - B_{a,\beta}(\Omega; x_*)\| \leq \frac{1}{2} \|F' (x) - F' (x_*)\| \\
\leq \frac{K}{2} \|x - x_*\|^p,
\]
and we can get the inequality
\[
\|B_{a,\beta}(x_*)^{-1}(B_{a,\beta}(\Omega; x) - B_{a,\beta}(\Omega; x_*))\| \leq K\|x - x_*\|^p < 1.
\] (30)

Hence, by making use of the Banach lemma, we can obtain
\[
\|B_{a,\beta}(x)^{-1}\| \leq \frac{2\gamma}{1 - \gamma K\|x - x_*\|^p}.
\] (31)

Similarly,
\[
\|C_{a,\beta}(\Omega; x) - C_{a,\beta}(\Omega; x_*)\| \leq \frac{1}{2} \|F' (x) - F' (x_*)\| \leq \frac{K}{2} \|x - x_*\|^p
\]
\[
\|C_{a,\beta}(\Omega; x)^{-1}\| \leq \frac{2\gamma}{1 - \gamma K\|x - x_*\|^p}.
\] (32)

Then, we have

\[
\begin{align*}
\|M_{a,\beta}(\Omega; x) - M_{a,\beta}(\Omega; x_*)\| \\
= \|B_{a,\beta}(x)^{-1}C_{a,\beta}(x) - B_{a,\beta}(x_*)^{-1}C_{a,\beta}(x_*)\| \\
= \|B_{a,\beta}(x)^{-1}(C_{a,\beta}(x) - C_{a,\beta}(x_*)) + (B_{a,\beta}(x)^{-1} - B_{a,\beta}(x_*)^{-1})C_{a,\beta}(x_*)\| \\
\leq \|B_{a,\beta}(x)^{-1}\| \left(\|C_{a,\beta}(\Omega; x) - C_{a,\beta}(\Omega; x_*)\| + \|B_{a,\beta}(\Omega; x) - B_{a,\beta}(\Omega; x_*)\|\|M_{a,\beta}(\Omega; x_*)\|\right) \\
\leq \frac{2\gamma}{1 - \gamma K\|x - x_*\|^p} \left(\frac{K}{2} \|x - x_*\|^p + \frac{K}{2} \|x - x_*\|^p \theta\right) \\
= \frac{(1 + \theta)\gamma K\|x - x_*\|^p}{1 - \gamma K\|x - x_*\|^p} \leq \tau \theta.
\end{align*}
\] (33)

We can use (33); hence,
\[
\begin{align*}
\|M_{a,\beta}(\Omega; x)\| &\leq \|M_{a,\beta}(\Omega; x) - M_{a,\beta}(\Omega; x_*)\| \\
&\leq \|M_{a,\beta}(\Omega; x_*)\| \leq (1 + \tau)\theta < 1.
\end{align*}
\] (34)

Now, we turn to estimate the error about the Newton–PGSS iteration \{x_k\}_{0}^{\infty} defined above. Clearly, it holds that
\[
x_{k+1} - x_* = x_k - x_* - (I - M_{a,\beta}(\Omega; x_*))^{k+1} F' (x_k)^{-1} F(x_k)
\leq -F' (x_k)^{-1}(F' (x_k) - F' (x_*) (x_k - x_*))
\leq F' (x_k)^{-1} F(x_k) (x_k - x_*)
+ M_{a,\beta}(\Omega; x_k) F' (x_k)^{-1} F(x_k).
\] (35)

Hence, we can obtain
\[
M_{a,\beta}(\Omega; x_k)^{k+1} F' (x_k)^{-1} F(x_k) = M_{a,\beta}(\Omega; x_k)^{k} F' (x_k)^{-1}
\cdot (F(x_k) - F(x_*) - F' (x_*) (x_k - x_*))
\cdot (x_k - x_*)
\]
\[
+ M_{a,\beta}(\Omega; x_k)^{k} F' (x_k)^{-1} F(x_k).
\] (36)
\[ \| M_{a,b}(\Omega; x_k)^k F'(x_*) (x_k - x_*) \| \leq \| M_{a,b}(\Omega; x_k)^k \| \| F'(x_*) \| \| (F(x_k - F(x_*)) (x_k - x_*)) \| + \| F'(x_*) (x_k - x_*) \| \]

\[ \leq \frac{\gamma((\tau + 1)\theta)^k}{1 - \gamma K}\| x_k - x_* \|^{1+p} + 2\delta \| x_k - x_* \|, \]

\[ \| x_{k+1} - x_* \| \leq \frac{\gamma}{1 - \gamma K}\| x_k - x_* \|^{1+p} + \| x_k - x_* \| \]

\[ + \frac{\gamma((\tau + 1)\theta)^k}{1 - \gamma K}\| x_k - x_* \|^{1+p} + 2\delta \| x_k - x_* \| \]

\[ = \frac{\gamma}{1 - \gamma K}\| x_k - x_* \|^{1+p} + 2\delta [(\tau + 1)\theta]^k \] \| x_k - x_* \|

\[ = g(\| x_k - x_* \|^{1+p}; l_k) \| x_k - x_* \|. \] (37)

where

\[ g(t^p; c) = \frac{\gamma}{1 - \gamma K t^p} \left( \frac{3 + p}{1 + p} K t^p + 2\beta (\tau + 1) \theta^c \right). \] (38)

This function is about \( t \) increasing and about \( c \) decreasing; hence,

\[ g(\| x_k - x_* \|^{1+p}; l_k) \leq g(r_0^p; l_0) < 1. \] (39)

In fact, for \( k = 0 \), we have \( \| x_0 - x_* \| < r < r_0 \), as \( x^{(0)} \in \mathbb{N}(x_*, r) \). It follows that

\[ \| x_1 - x_* \| \leq g(r_0^p; l_0) \| x_0 - x_* \| \leq \| x_0 - x_* \| < r_0. \] (40)

Hence, \( x_1 \in \mathbb{N}(x_*, r) \), and by making use of mathematical methods of induction, suppose \( x_m \in \mathbb{N}(x_*, r) \) is valid for some positive integer \( k = m \). Then, by making use of the function above again, we can straightforwardly deduce the estimate

\[ \| x_{m+1} - x_* \| \leq g(r_0^p; l_0) \| x_m - x_* \| \] \( \| x_m - x_* \| \leq x_0 - x_* \| < r_0. \) (41)

which show that it also holds true for \( k = m + 1 \) as the following. In addition, we have

\[ \| x_{m+1} - x_* \| \leq g(r_0^p; l_0) \| x_m - x_* \| \leq \| x_0 - x_* \| < r_0, \] (42)

and, hence, \( x_{m+1} \in \mathbb{N}(x_*, r_0) \). Now, the conclusion we are proving above is as follows.

5. Semilocal Convergence of the Newton-PGSS Method

Assumption 2. For all \( x \in \mathbb{N}(x_0, r) \subset \mathbb{N}_0 \), where \( r < (1/2) \sqrt{\| L \|} \), assume the following conditions hold:

(A1) The bounded condition: there exist positive constants \( \delta \) and \( \gamma \) such that

\[ \max\{ \| A(x_0) \|, \| B(x_0) \| \} \leq \beta, \]

\[ \| F'(x_*)^{-1} \| \leq \gamma, \]

\[ \| F(x_0) \| \leq \delta. \] (43)

(A2) The Hölder condition: there exist nonnegative constants \( L_a \) and \( L_b \) for all \( x, y \in \mathbb{N}(x_0, r) \in \mathbb{N}_0 \)

\[ \| A(x) - A(y) \| \leq L_a\| x - y \|^p, \]

\[ \| B(x) - B(y) \| \leq L_b\| x - y \|^p, \] (44)

with the exponent \( p \in (0, 1] \), and we define \( L = L_a + L_b \).

Lemma 2. Under Assumption 2, for all \( x, y \in \mathbb{N}(x_*, r) \), then \( F'(x)^{-1} \) exists, and we have the following inequalities:

\[ \| F'(x) - F'(y) \| \leq L\| x - y \|^p, \]

\[ \| F'(x) \| \leq L\| x - x_0 \|^p + 2\beta, \]

\[ \| B_{a,b}(\Omega; x) - B_{a,b}(\Omega; y) \| = \| C_{a,b}(\Omega; x) - C_{a,b}(\Omega; y) \| \]

\[ \| B_{a,b}(\Omega; x)^{-1} \| \leq \frac{2\| x \|^p}{1 - L\| x - x_0 \|^p} \] (45)

Proof. The proof is omitted since it is the same as Lemma 1.

Theorem 3. Under Assumption 2, for all \( x, y \in \mathbb{N}(x_*, r) \), then \( F'(x)^{-1} \) exists, and we have the inequalities in (45).

Now, we construct the following sequence of functions:
Lemma 3. Assume that constants satisfy
\[
\delta^p y^{p+1} L \leq \frac{1 - \eta}{2^p (1 + \eta^2)}.
\] (49)

Denote \( t_* = \sqrt[2p]{b/a} \), and then, when \( t \in [0, t_*] \), the following inequalities hold that
\[
\begin{align*}
g(t) &\geq 0, \\
g'(t) &< 0, \\
g''(t) &> 0, \\
h(t) &< g'(t) < 0.
\end{align*}
\] (51)

Proof. The proof is omitted since it is straightforward.

Theorem 4. Under the assumptions of lemma in this section, \( r_* = \min(r_1, r_2) \) with
\[
\begin{align*}
r_1 &= \sqrt{\frac{\theta r}{\gamma L (1 + \tau + \theta r)}}, \\
r_2 &= \sqrt{\frac{b}{a}},
\end{align*}
\] (52)

satisfying
\[
r_1 < r_2.
\] (53)

And, define \( l_0 = \liminf_{k \to \infty} l_k \), and the constant \( l_0 \) satisfies
\[
l_0 > 1 - \frac{-\ln \eta}{\ln ((r + 1)/\theta)},
\] (54)

where the symbol \([\cdot]\) is used to denote the smallest integer no less than the corresponding real number, \( r \in (0, (1 - \theta)/\theta) \) a prescribed positive constant, and
\[
\theta = \theta(a, \beta; x_0) = \|M_{a, \beta}(x_0)\| < 1.
\] (55)

Then, the iteration sequence \( \{t_k\}_{k=0}^\infty \) generated by the Newton–PGSS is well defined and converges to \( x_* \), which satisfies \( F(x_*) \).

Proof. Firstly, we construct the sequence
\[
\begin{align*}
t_0 &= 0, \\
t_{k+1} &= t_k - \frac{g(t_k)}{h(t_k)}.
\end{align*}
\] (56)

We have
\[
g\left( \sqrt[2p]{b/a} \right) > 0. \tag{57}
\]

Furthermore, \( g(0) = c > 0 \); hence, we have \( r_* \) which satisfies \( g(r_*) = 0 \), where \( t_1 = 2\gamma \delta \) because (49) and (50). Hence, we have
\[
g(2\gamma \delta) > 0,
\]

\[
t_1 = 2\gamma \delta < r_* < \sqrt[2p]{b/a}.
\] (58)

Therefore, we have
\[
t_0 < t_1 < r_*.
\] (59)

Now, we assume that \( t_{k-1} < t_k < r_* \), and by making use of mathematical methods of induction, we have
\[
t_{k+1} = t_k - \frac{g(t_k)}{h(t_k)}.
\] (60)

Because
\[
h(t_k) \leq g'(t_k) < 0,
\] (61)

hence
\[
t_{k+1} > t_k.
\] (62)

Furthermore, \( m(t) = t - (g(t)/g'(t)) = m'(t) = (g(t) g''(t))/g'(t)^2 \); then, \( m'(t) \) is an increasing function in \((0, \sqrt[2p]{b/a})\) and \(-1/h(t_k)) \leq -1/g'(t_k)) \); hence, we have
\[
t_{k+1} = t_k - (g(t_k)/h(t_k)) \leq t_k - (g(r_*)/g'(r_*)) \leq r_* - (g(r_*)/g'(r_*)) < r_* < r_2,
\]

and it exists as point \( t_* = \liminf_{k \to \infty} t_k \).

Next, prove the following inference by mathematical induction:
\[ \|x_{k+1} - x_k\| \leq t_{k+1} - t_k, \quad \|F'(x_k)\| \leq \frac{1 - \eta yL_t^p}{\eta(1 + \eta)} (t_{k+1} - t_k), \] 
(63)

where
\[ \|x_1 - x_0\| \leq \|F'(x_0)^{-1}F(x_0)\| + \|M_{a,b}(\Omega;x_0)^{1/2}F'(x_0)^{-1}F(x_0)\| \leq \gamma(1 + \theta^4)\delta, \]
\[ t_1 - t_0 = 2\gamma\delta, \]
\[ \|F(x_0)\| \leq \delta \leq \frac{2\delta}{1 + \eta} = \frac{1 - \eta yL_t^p}{\eta(1 + \eta)} (t_1 - t_0), \]
\[ \|F(x_k)\| \leq \|F(x_k) - F(x_{k-1}) - F'(x_k)(x_k - x_{k-1})\| + \|F(x_k) + F'(x_{k-1})(x_k - x_{k-1})\| \]
\[ \leq \frac{L}{1 + p}\|x_k - x_{k-1}\|^{p+1} + \eta\|F(x_{k-1})\|. \]
(64)

Because
\[ \delta^p y^{p+1}L \leq \frac{1 - \eta}{2^p(1 + \eta)} \leq \frac{1}{2^p}, \]
(65)

we can derive inequality
\[ \frac{(1 - \eta)^2}{1 - \eta yL_t^p} \leq \frac{a}{-h(t_k)}. \]
(66)

Hence,
\[ \frac{\gamma(1 + \eta)^2}{1 - \gamma yL_t^p} \|F(x_k)\| \leq \gamma(1 + \eta)^2 \left( \frac{2}{1 + p} (t_k - t_{k-1})^{1+p} + \eta L_t^{k-1} (t_k - t_{k-1}) \right) \]
\[ \leq \frac{1}{1 + p} \frac{a}{-h(t_k)} (t_k - t_{k-1})^{p+1} + \frac{\eta}{h(t_k)} (t_k - t_{k-1}). \]
(67)

And because
\[ \frac{a}{1 + p}(t_k - t_{k-1})^{p+1} + \eta(t_k - t_{k-1}) \]
\[ \leq g(t_k) - g(t_{k-1}) - h(t_{k-1})(t_k - t_{k-1}) \]
\[ = \frac{a}{1 + p}(t_k^{p+1} - t_{k-1}^{p+1}) - b(t_k - t_{k-1}) - (at_k^p - 1)(t_k - t_{k-1}), \]
\[ = \frac{1}{1 + p}a(t_k^{p+1} - t_{k-1}^{p+1}) - at_k^p(t_k - t_{k-1}) + \eta(t_k - t_{k-1}), \]
\[ \leq 1, \]
(68)

we can give
\[ \frac{\gamma(1 + \eta)^2}{1 - \gamma yL_t^p} \|F(x_k)\| \leq \gamma(1 + \eta)^2 \left( 1 + ((\tau + 1)\theta)^4 \right) L_t^{k-1} (t_k - t_{k-1}). \]
\[ \|x_{k+1} - x_k\| \leq \left( (1 - M_{a,b}(\Omega;x_0)^{1/2}) F'(t_k)^{-1} F(x_k) \right) \|F(x_k)\| \]
\[ \leq \left( 1 + ((\tau + 1)\theta)^4 \right) \frac{\gamma}{1 - \gamma yL_t^p} \|F(x_0)\|, \]
(70)

and we can have inequality
\[ \frac{(1 + \eta)^p}{1 - \eta yL_t^p} \|F(x_k)\| \leq t_{k+1} - t_k. \]
(71)

Since the sequence \( \{t_k\}_{k=0}^\infty \) converges to \( t_\ast \) and
\[ \|x_{k+1} - x_0\| \leq \|x_{k+1} - x_k\| + \|x_k - x_{k-1}\| + \cdots + \|x_1 - x_0\| \]
\[ \leq t_{k+1} - t_k + t_k - t_{k-1} + \cdots + t_1 - t_0 \leq r_\ast, \]
(72)

where \( r_\ast < \sqrt{b/a} \), the sequence \( x_k \) also converges to \( x_\ast \). The proof has been completed as above.

6. The Modified Newton-PGSS Method and Its Local Convergence

In this section, we improve Newton-PGSS and introduce the modified Newton-PGSS and prove the local convergence of the modified Newton-PGSS method briefly.
The modified Newton method is a kind of algorithm based on the Newton method. Its principle is to reduce the calculation times of the inverse matrix of Jacobian matrix, making the algorithm more efficient. It only needs to calculate inverse matrix once every two steps. The format of the algorithm is shown below:

\[
F'(x_k)d_k = -F(x_k), \quad y_k = x_k + d_k, \\
F'(x_k)h_k = -F(y_k), \quad x_{k+1} = y_k + h_k. \tag{73}
\]

Then, we get the modified Newton-PGSS method for solving nonlinear system (1) (Algorithm 4).

**Assumption 3.** For all \( x \in \mathbb{N}(x_*, r) \subset \mathbb{N}_0 \), assume the following conditions hold.

(A1) The bounded condition: there exist positive constants \( \delta \) and \( \gamma \) such that

\[
\max \| A(x) \|, \| B(x) \| \leq \delta, \\
\| F'(x) \|^{-1} \leq \gamma. \tag{74}
\]

(A2) The Hölder condition: there exist nonnegative constants \( K_w \) and \( K_r \) such that

\[
\| A(x) - A(x_*) \| \leq K_w\| x - x_* \|^p, \\
\| B(x) - B(x_*) \| \leq K_r\| x - x_* \|^p, \tag{75}
\]

with the exponent \( p \in (0, 1] \).

**Lemma 4.** Under Assumption 3, for all \( x, y \in \mathbb{N}(x_*, r) \), if \( r \in (0, (1/(\sqrt{\gamma}K))^1/p) \), then \( F'(x)^{-1} \) exists. And, the following inequalities hold with \( K := K_w + K_r \) for all \( x, y \in \mathbb{N}(x_*, r) \):

\[
\| F'(x) - F'(x_*) \| \leq K\| x - x_* \|^p, \\
\| F'(x) \|^{-1} \leq \frac{\gamma}{1 - \gamma K\| x - x_* \|^p}, \\
\| F(x) \| \leq \frac{K}{1 + p}\| x - x_* \|^{1+p} + 2\| y - x_* \|. \tag{76}
\]

**Theorem 5.** Under the assumptions of Lemma 4, suppose \( r \in (0, r_0) \) and define \( r_0^* = \min\{r_0, r^{(1)}_+\} \), where

\[
r_+^{(1)} = \sqrt{\frac{\tau \delta}{yK(1 + \theta + \tau \theta)}}. \\
r_+^{(2)} = \sqrt{\frac{1 - 2\delta((r + 1)\theta)^{1/2}}{((4 + 2p)/(1 + p))Ky^p}}. \\
r_+^{(3)} = \frac{1}{\gamma K}. \tag{77}
\]

with \( l_0 = \liminf_{k \to \infty} l_k \), and the constant \( u \) satisfies

\[
l_0 > \left[ -\frac{\ln 2 \delta}{\ln((r + 1)\theta)} \right], \tag{78}
\]

where the symbol \( \lfloor \cdot \rfloor \) is used to denote the smallest integer no less than the corresponding real number, \( \tau \in (0, ((1 - \theta)/(\theta))) \) is a prescribed positive constant, and

\[
\theta \equiv \theta(a, \beta; x_*) = \| M_{a, \beta}(x_*) \| \\
\leq \max_{\lambda \in \lambda(\Omega^{(1/2)p}(x_*, l_0^{1/2}))} \frac{1 - \lambda^2}{1 + \lambda^2} \equiv \sigma(a, \beta; x_*) < 1, \tag{79}
\]

with \( a \) and \( \beta \) are more than 0.

Then, for any \( x \in \mathbb{N}(x_*, r) \subset \mathbb{N}_0 \), \( t \in (0, r) \) and \( c > u \), it holds that

\[
\| M_{a, \beta}(V; x) \| \leq (r + 1)\theta < 1, \\
g(t^p; c) = \frac{\gamma}{1 - \gamma Kt^p} \left( \frac{3 + p}{1 + p} + 2\| (r + 1)\theta \|^p \right) \leq g(r_0^*; u) < 1. \tag{80}
\]

**Proof.** It is the same as Theorem 2.

In Theorem 5, we get the fact that \( \| x^{(1)}(x_*) \| \leq g(r_0^*; u)\| x_0 \| - x_* \| \) which is the modified Newton-PGSS has the similar result as the following.

**Theorem 6.** Under the conditions of Theorem 5, we have the fact that, for any \( x_0 \in \mathbb{N}(x_*, r) \) with corresponding \( \{l_k\}_{k=0}^{\infty}, \{m_k\}_{k=0}^{\infty} \) of positive integers, the iteration sequence \( \{x_k\}_{k=0}^{\infty} \) which is generated by the modified Newton-PGSS method is well defined and converges to \( x_* \). Furthermore, it has the following properties:

\[
\lim_{t \to \infty} \sup_{x_k \in \mathbb{N}(x_*, r)} \| x_k - x_* \|^{1/k} \leq g(r_0^*; u)^2. \tag{81}
\]

**Proof.** The proof of \( \| y_k - x_* \| \leq g(\| x_k - x_* \|^{1/p}; l_k)\| x_k - x_* \| \) is the same as \( \| x_{k+1} - x_* \| \leq g(\| x_k - x_* \|^{1/p}; l_k)\| x_k - x_* \| \) in
(1) Given an initial guess $x_0$, a nonnegative constant $\alpha$, a positive constant $\beta$, and two positive integer sequences $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$.
(2) For $k = 0, 1, \ldots$, until $\|F(x_k)\| \leq \|F(x_0)\|$ do:
(2.1) Set $d_{k,0} = h_{k,0} = 0$.
(2.2) For $l = 0, 1, \ldots, l_k - 1$, apply algorithm PGSS to the linear system (12):
$$\Omega (x_k) + F'(x_k) d_{k,l+1} = (\alpha \Omega (x_k) - F' (x_k)) d_{k,l} - F(x_k),$$
and obtain $d_{k,l}$ such that
$$\|F(x_k) + F'(x_k) d_{k,l}\| \leq \eta_k \|F(x_k)\|, \text{ for some } \eta_k \in [0, 1).$$
(2.3) Set
$$y_k = x_k + d_{k,l}.$$ 
(2.4) For $m = 0, 1, \ldots, m_k - 1$, apply algorithm PGSS to the linear system (12):
$$\Omega (x_k) + F'(x_k) h_{k,m+1} = (\alpha \Omega (x_k) - F' (x_k)) d_{k,l} - F(y_k),$$
and obtain $h_{k,m}$ such that
$$\|F(x_k) + F'(x_k) h_{k,m}\| \leq \eta_k' \|F(y_k)\|, \text{ for some } \eta_k' \in [0, 1).$$
(2.5) Set
$$x_{k+1} = y_k + h_{k,m}.$$ Where
$$\Omega (x_k) = \begin{pmatrix} \alpha I_1 (x_k) & 0 \\ 0 & \beta I_2 (x_k) \end{pmatrix}$$
and $I_1$ is an $n \times n$ identity matrix and $I_2$ is a $m \times m$ identity matrix.

Algorithm 4: MN-PGSS (modified Newton-PGSS method).

Theorem 2. And, from the definition of $x_k$ and in Lemma 4, we can easily get that

$$\|x_{k+1} - x_k\| = \|y_k - x_k - (I - M (\alpha, \beta; x_k)^{m_k}) F'(x_k)^{-1} F(y_k)\|$$
$$= \|y_k - x_k - F'(x_k)^{-1} F(y_k)\| + \|M (\alpha, \beta; x_k)^{m_k} F'(x_k)^{-1} F(y_k)\|,$$ (82)
where

\[
y_k - x_0 - F'(x_k)^{-1}F(y_k) = -F'(x_k)^{-1}(F(y_k - F(x_k) - F'(x_k)(y_k - x_0) + F'(x_k)(y_k - x_0)).
\]

(83)

By Lemma 4 and similar to the proof of \(\|F(x)\|\), we can get it:

\[
y_k - x_0 - F'(x_k)^{-1}F(y_k) \leq \frac{\gamma}{1 - \gamma K}\|x_k - x_*\|^p \left( \frac{1 + ((r + 1)\theta)^{m_k}}{1 + p} \|y_k - x_*\|^p + K\|x_k - x_*\|^p \right)\|y_k - x_*\|
\]

(84)

Combining (84) with (85), we can obtain

\[
\|x_{k+1} - x_*\| \leq \left( \frac{\gamma K}{1 - \gamma K}\|x_k - x_*\|^p \left( \frac{1 + ((r + 1)\theta)^{m_k}}{1 + p} \|y_k - x_*\|^p + K\|x_k - x_*\|^p \right) \right)
\]

\[
+ \frac{2\delta((r + 1)\theta)^{m_k}}{1 - \gamma K}\|x_k - x_*\|^p \|y_k - x_*\|^{p+1}
\]

(85)

By utilizing mathematical induction, we can get the fact that any \(x_0 \in \mathbb{N}_0(x_0, r)\) and nonnegative integer \(k\), and we have

\[
\|x_{k+1} - x_*\| \leq g \left( \frac{\gamma K}{1 - \gamma K}\|x_k - x_*\|^p \left( \frac{1 + ((r + 1)\theta)^{m_k}}{1 + p} \|y_k - x_*\|^p + K\|x_k - x_*\|^p \right) \right)\|x_{k+1} - x_*\|
\]

\[
+ \frac{2\delta((r + 1)\theta)^{m_k}}{1 - \gamma K}\|x_k - x_*\|^p \|y_k - x_*\|^{p+1}
\]

(86)

Because \(g(r_0^*; u < 1\), we can make a conclusion that \(x_0\) converges to \(x_*\) as \(n \rightarrow +\infty\) from (86). The proof of theorem is completed.

7. Numerical Example

In this section, we show the efficiency of the modified Newton-PGSS method. Because such problems have not been analyzed before, in this paper, the first step is that we just compare the modified Newton-PGSS method with the
Newton-GMRES algorithm. The numerical results in iterations are splitting methods. And, the second step, we will discuss which is more effective as preconditioner in Newton-GMRES algorithm. The numerical results in Example 1 were computed using MATLAB Version R2011b, on an iMac with a 3.20 GHz Intel Core i5-6500 CPU, and 8.00 GB RAM, with machine accuracy $\varepsilon = 2.22 \times 10^{-16}$.

### Table 1: Numerical results of inexact Methods for $\nu = 0.1$ and $\eta = 0.4$.

| $n$ | Method     | Optimal values of $\alpha/\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|-----|------------|----------------------------------|-----------------|--------------|----------|----------|
| 16  | Newton-PGSS | $(9.7, 7.8)$                     | $2.2284 \times 10^{-4}$ | 3.89         | 11       | 5        |
|     | Newton-MPGSS| $(9.7, 7.8)$                     | $6.1064 \times 10^{-5}$ | 4.36         | 7        | 5        |
|     | Newton-RHSS | 14.4                              | $2.3976 \times 10^{-4}$ | 7.12         | 13       | 10       |
| 20  | Newton - PGSS | $(11.3, 9.2)$                | $3.3053 \times 10^{-4}$ | 15.02        | 12       | 6        |
|     | Newton-MPGSS| $(11.3, 9.2)$                     | $1.6762 \times 10^{-4}$ | 14.7         | 5        | 6.6      |
|     | Newton - RHSS | 17                            | $3.3234 \times 10^{-4}$ | 26.92        | 13       | 13.6     |
| 32  | Newton - PGSS | $(15.5, 9.9)$                | $5.31 \times 10^{-4}$ | 227.52       | 13       | 10.3     |
|     | Newton-MPGSS| $(11.3, 9.2)$                     | $1.4157 \times 10^{-4}$ | 227.8        | 7        | 14.4     |
|     | Newton - RHSS | 20.6                        | $3.5082 \times 10^{-4}$ | 453.84       | 15       | 28.2     |

### Table 2: Numerical results of inexact methods for $\nu = 1$ and $\eta = 0.4$.

| $n$ | Method     | Optimal values of $\alpha/\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|-----|------------|----------------------------------|-----------------|--------------|----------|----------|
| 16  | Newton-PGSS | $(8.8, 7.4)$                     | $1.6253 \times 10^{-4}$ | 20.74        | 15       | 44.7     |
|     | Newton-MPGSS| $(11.3, 9.2)$                     | $1.1582 \times 10^{-4}$ | 30.8         | 8        | 39       |
|     | Newton-RHSS | 13.2                              | $1.7534 \times 10^{-4}$ | 82.2         | 13       | 124.9    |
| 20  | Newton - PGSS | $(7.7)$                        | $2.6262 \times 10^{-4}$ | 195.01       | 14       | 85.9     |
|     | Newton-MPGSS| $(13.6)$                          | $3.2972 \times 10^{-4}$ | 140.59       | 7        | 45.4     |
|     | Newton - RHSS | 18                            | $2.695 \times 10^{-4}$ | 253          | 13       | 167.5    |
| 32  | Newton - PGSS | $(9.4, 10.2)$                | $2.297 \times 10^{-4}$ | 356.2        | 15       | 153.3    |
|     | Newton-MPGSS| $(11.4, 9.8)$                     | $8.8723 \times 10^{-5}$ | 3593         | 8        | 135.5    |
|     | Newton - RHSS | 21.7                         | $3.596 \times 10^{-4}$ | 5488.3       | 13       | 327.5    |

### Table 3: Numerical results of inexact methods for $\nu = 0.1$ and $\eta = 0.2$.

| $n$ | Method     | Optimal values of $\alpha/\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|-----|------------|----------------------------------|-----------------|--------------|----------|----------|
| 16  | Newton-PGSS | $(7.7)$                        | $7.6864 \times 10^{-5}$ | 6.57         | 8        | 11.5     |
|     | Newton-MPGSS| $(7.7, 9.6)$                     | $2.5763 \times 10^{-4}$ | 4.66         | 4        | 9        |
|     | Newton-RHSS | 18.6                              | $9.8486 \times 10^{-5}$ | 8.34         | 9        | 16.1     |
| 20  | Newton - PGSS | $(8.2, 9)$                      | $3.3090 \times 10^{-4}$ | 15.43        | 8        | 12.4     |
|     | Newton-MPGSS| $(12.6)$                          | $2.5400 \times 10^{-4}$ | 14.4         | 4        | 8.4      |
|     | Newton - RHSS | 14.6                          | $2.0703 \times 10^{-4}$ | 21.44        | 8        | 23.8     |
| 32  | Newton - PGSS | $(8.2, 9.5)$                | $2.2875 \times 10^{-4}$ | 394.4        | 8        | 29.2     |
|     | Newton-MPGSS| $(10.8, 13.5)$                   | $1.7649 \times 10^{-4}$ | 346.78       | 4        | 21       |
|     | Newton - RHSS | 20.6                         | $1.9104 \times 10^{-4}$ | 507.3        | 7        | 48.5     |

### Table 4: Numerical results of inexact methods for $\nu = 1$ and $\eta = 0.2$.

| $n$ | Method     | Optimal values of $\alpha/\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|-----|------------|----------------------------------|-----------------|--------------|----------|----------|
| 16  | Newton-PGSS | $(8.7, 6)$                      | $2.7300 \times 10^{-4}$ | 41.67        | 9        | 75.9     |
|     | Newton-MPGSS| $(20.7)$                          | $1.2180 \times 10^{-4}$ | 33.15        | 5        | 47.2     |
|     | Newton-RHSS | 17.8                              | $8.4271 \times 10^{-5}$ | 86.22        | 8        | 218.8    |
| 20  | Newton - PGSS | $(8.9)$                       | $3.3351 \times 10^{-4}$ | 174          | 9        | 112.4    |
|     | Newton-MPGSS| $(8.9, 6)$                          | $3.5554 \times 10^{-5}$ | 93.2         | 5        | 142.3    |
|     | Newton - RHSS | 18                              | $9.3834 \times 10^{-5}$ | 258          | 8        | 289.6    |
| 32  | Newton - PGSS | $(8.9, 2)$                       | $1.95 \times 10^{-4}$ | 3784         | 9        | 304.9    |
|     | Newton-MPGSS| $(10.8, 13.5)$                    | $1.485 \times 10^{-4}$ | 3679         | 5        | 219.2    |
|     | Newton - RHSS | 21.7                         | $1.4216 \times 10^{-4}$ | 5944         | 8        | 564.5    |
Example 1. Consider the Stokes flow problem. Find \( \mathbf{u} \) and \( \mathbf{w} \) such that

\[
\begin{cases}
-\nu \Delta \mathbf{u} + \nabla \mathbf{w} = \mathbf{f}, & \text{in } \Omega, \\
\nabla \cdot \mathbf{u} = \mathbf{g}, & \text{in } \Omega, \\
u \mathbf{u}(t, x, y) = 0, & \text{on } (0, 1] \in \partial \Omega, \\
\int_{\Omega} \mathbf{w}(x) dx = 0,
\end{cases}
\]

where \( \Omega = (0, 1) \times (0, 1) \), with \( \partial \Omega \) is the boundary of \( \Omega \), \( \mathbf{u} \) is a vector-valued function representing the velocity \( \nu > 0 \) is the viscosity constant, \( \Delta \) is the componentwise Laplace operator, and \( \mathbf{w} \) is a scalar function representing the pressure. By discretizing the function above with the upwind scheme, we obtain the saddle point problem in which

\[
\mathbf{A} = \begin{pmatrix}
I \otimes T + T \otimes I & 0 \\
0 & I \otimes T + T \otimes I
\end{pmatrix} \in \mathbb{R}^{p^2 \times 2p^2},
\]

\[
\mathbf{B} = \begin{pmatrix}
I \otimes F \\
F \otimes I
\end{pmatrix} \in \mathbb{R}^{2p^2 \times p^2},
\]

where

\[
T = \frac{\nu}{h^2} \cdot \text{tridiag} (-1, 2, -1) \in \mathbb{R}^{p \times p},
\]

\[
F = \frac{1}{h} \cdot \text{tridiag} (-1, 1, 0) \in \mathbb{R}^{2p \times 1},
\]

\[
\mathbf{f} = \begin{pmatrix}
\nu_1, \nu_2, \ldots, \nu_p, e_{\nu_1}, e_{\nu_2}, \ldots, e_{\nu_p}, e_{\nu_1}, e_{\nu_2}, \ldots, e_{\nu_p}
\end{pmatrix}^T \in \mathbb{R}^{p \times p},
\]

\[
\mathbf{g} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{2 \times 1},
\]

with \( \otimes \) being the Kronecker product symbol. By applying the centered finite difference scheme on the equidistant discretization grid with the step size \( \Delta t = h = (1/(N + 1)) \), the system of nonlinear equations (1) is obtained with the following form:

\[
F(\mathbf{u}) = \mathbf{M} \mathbf{u} + \Psi(\mathbf{u}) = 0,
\]

where

\[
M = \begin{pmatrix}
\mathbf{A} & \mathbf{B} \\
-\mathbf{B}^T & 0
\end{pmatrix},
\]

\[
\Psi(\mathbf{u}) = \begin{pmatrix}
e_{\nu_1}^1, e_{\nu_2}^1, \ldots, e_{\nu_1}^p, e_{\nu_2}^1, \ldots, e_{\nu_1}^p, e_{\nu_2}^1, \ldots, e_{\nu_1}^p
\end{pmatrix}^T 
\]

Then, the Jacobian matrix is

\[
F'(\mathbf{u}) = \mathbf{M} + \text{diag}(e_{-\nu_1}, e_{-\nu_2}, \ldots, e_{-\nu_1}, \ldots, e_{-\nu_2}, \ldots, e_{-\nu_1}, e_{-\nu_2}, \ldots, e_{-\nu_1}, 1, 1, \ldots, 1).
\]

Firstly, we compare the algorithms whose inner iterations are splitting methods, such as Newton RHSS, Newton PGSS, and modified Newton PGSS. The parameters needed in the problem are chosen by using the traversal method for the purpose of comparison: the initial guess \( \mathbf{u}_0 = 0 \), the stopping criterion for the outer iteration is set to be

\[
\frac{\| F(\mathbf{u}_k) \|_2}{\| F(\mathbf{u}_0) \|_2} \leq 10^{-5},
\]

and the prescribed tolerance \( \eta_k \) and \( \bar{\eta}_k \) for controlling the accuracy of the iteration are both set to be \( \eta \), which satisfies inequality

\[
\frac{\| F'(\mathbf{u}_k) \|_2}{\| F(\mathbf{u}_k) \|_2} \leq \eta.
\]

For different inner tolerance \( \eta = 0.4, 0.2, \) and \( 0.1 \) and problem parameters \( \nu = 1 \) and \( 0.1 \), the results about outer IT, inner IT, and CPU are listed in the numerical tables corresponding to the referred inexact Newton methods. Because the linear matrix of the solution is different in each iteration, there is no way to find the optimal parameters in theory. Thus, we get the most efficient algorithm by traversing for the parameters of different algorithms, and then, we tabulate these results. When the number of steps, time, and error show an earlier increase and later decrease trend, the iteration is stopped to determine the range of parameters.
We use this method to narrow the parameter range and get "the best parameters at present" until the result (such as step) does not change. For the selection of two parameters (denoted them as $\alpha$ and $\beta$), first, we fix the parameter $\alpha$ and traverse the parameter $\beta$ by using the single parameter traversal method. Then, we fix the parameter $\beta$ and traverse the parameter $\alpha$. We repeat the process until the result does not change. We can get information from Tables 1–6 that Newton-PGSS performs better than Newton-RHSS in the iterative CPU. Moreover, the Newton-MPGSS algorithm is

Table 6: Numerical results of inexact methods for $v = 1$ and $\eta = 0.1$.

| $n$  | Method         | Optimal values of $a/\alpha,\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|----------------|-----------------------------------|-----------------|-------------|----------|----------|
| 16   | Newton-PGSS    | (8.2, 8.2)                        | $2.5370 \times 10^{-4}$ | 40.885      | 7        | 96       |
|      | Newton-RHSS   | (11.8, 4)                         | $3.9974 \times 10^{-5}$ | 40.9        | 4        | 85.4     |
| 20   | Newton – PGSS | (8, 10.4)                         | $3.4557 \times 10^{-4}$ | 156         | 7        | 145.1    |
|      | Newton-MPGSS  | (11.7, 3)                         | $7.1642 \times 10^{-6}$ | 134         | 4        | 155.2    |
|      | Newton – RHSS | 17.2                              | $3.9646 \times 10^{-5}$ | 272         | 6        | 404.3    |
| 32   | Newton – PGSS | (10.7, 11.7)                      | $5.1146 \times 10^{-4}$ | 3001        | 7        | 251.6    |
|      | Newton-MPGSS  | (11.4, 7.6)                       | $2.6628 \times 10^{-4}$ | 3241        | 4        | 308.8    |
|      | Newton – RHSS | 21.4                              | $5.8568 \times 10^{-5}$ | 5907        | 6        | 790.7    |

Table 7: Numerical results of preconditioned inexact Newton methods for $v = 0.1$ and $\eta = 0.4$.

| $n$  | Precondition for GMRES | Optimal values of $a/\alpha,\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|------------------------|-----------------------------------|-----------------|-------------|----------|----------|
| 16   | RHSS                   | –                                 | $2.2319 \times 10^{-4}$ | 1.2         | 15       | 3.2      |
|      | PGSS                   | (2, 1)                            | $2.5914 \times 10^{-4}$ | 0.7         | 7        | 1        |
| 20   | RHSS                   | –                                 | $2.4378 \times 10^{-4}$ | 3.97        | 21       | 3.95     |
|      | PGSS                   | (8, 1)                            | $1.6342 \times 10^{-4}$ | 2.58        | 8        | 1        |
| 32   | RHSS                   | –                                 | $5.1384 \times 10^{-4}$ | 16.21       | 16       | 9.9      |
|      | PGSS                   | (5, 1)                            | $1.7622 \times 10^{-4}$ | 16.52       | 8        | 1        |

Table 8: Numerical results of preconditioned inexact Newton methods for $v = 0.1$ and $\eta = 0.2$.

| $n$  | Precondition for GMRES | Optimal values of $a/\alpha,\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|------------------------|-----------------------------------|-----------------|-------------|----------|----------|
| 16   | RHSS                   | –                                 | $6.4114 \times 10^{-3}$ | 1.06        | 9        | 4.4      |
|      | PGSS                   | (2, 2)                            | $1.3215 \times 10^{-4}$ | 1.36        | 7        | 1        |
| 20   | RHSS                   | –                                 | $1.2495 \times 10^{-4}$ | 1.06        | 10       | 6.3      |
|      | PGSS                   | (3, 2)                            | $3.0298 \times 10^{-4}$ | 1.83        | 7        | 1        |
| 32   | RHSS                   | –                                 | $1.5261 \times 10^{-4}$ | 19.6        | 10       | 17.2     |
|      | PGSS                   | (2, 1)                            | $1.3380 \times 10^{-4}$ | 16.69       | 7        | 1        |

Table 9: Numerical results of preconditioned inexact Newton methods for $v = 0.1$ and $\eta = 0.1$.

| $n$  | Precondition for GMRES | Optimal values of $a/\alpha,\beta$ | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|------------------------|-----------------------------------|-----------------|-------------|----------|----------|
| 16   | RHSS                   | –                                 | $1.0703 \times 10^{-4}$ | 1.37        | 8        | 5.3      |
|      | PGSS                   | (3, 2)                            | $2.5352 \times 10^{-4}$ | 1.28        | 6        | 1        |
| 20   | RHSS                   | –                                 | $2.1225 \times 10^{-4}$ | 1.81        | 7        | 8.4      |
|      | PGSS                   | (3, 2)                            | $1.1526 \times 10^{-4}$ | 2.25        | 7        | 1        |
| 32   | RHSS                   | –                                 | $5.0033 \times 10^{-4}$ | 14.63       | 7        | 20.7     |
|      | PGSS                   | (1, 1)                            | $3.2577 \times 10^{-4}$ | 13.98       | 6        | 1        |
Table 10: Numerical results of preconditioned inexact Newton methods for \( v = 1 \) and \( \eta = 0.4 \).

| \( n \) | Preconditioner for GMRES | Optimal values of \( a/(\alpha, \beta) \) | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|--------------------------|----------------------------------|----------------|-------------|----------|---------|
| 16   | –                        | –                                | 1.5522 \times 10^{-4} | 3.14         | 16       | 29.3    |
|      | RHSS                     | 18                               | 1.7172 \times 10^{-4} | 4.07         | 13       | 1       |
|      | PGSS                     | (1, 3)                           | 1.8162 \times 10^{-4} | 1.81         | 11       | 1       |
| 20   | –                        | –                                | 2.0082 \times 10^{-4} | 9.91         | 17       | 50      |
|      | RHSS                     | 18                               | 1.8359 \times 10^{-4} | 10.48        | 13       | 1       |
|      | PGSS                     | (1, 1)                           | 2.8163 \times 10^{-4} | 3.32         | 9        | 1       |
| 32   | –                        | –                                | 4.5286 \times 10^{-4} | 294.57       | 39       | 79.8    |
|      | RHSS                     | 26                               | 3.3912 \times 10^{-4} | 138.79       | 15       | 1.07    |
|      | PGSS                     | (1, 3)                           | 3.8558 \times 10^{-4} | 29.95        | 10       | 1       |

Table 11: Numerical results of preconditioned inexact Newton methods for \( v = 1 \) and \( \eta = 0.2 \).

| \( n \) | Preconditioner for GMRES | Optimal values of \( a/(\alpha, \beta) \) | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|--------------------------|----------------------------------|----------------|-------------|----------|---------|
| 16   | –                        | –                                | 7.5565 \times 10^{-5} | 2.78         | 11       | 39.5    |
|      | RHSS                     | 16                               | 2.3989 \times 10^{-4} | 3.22         | 9        | 1       |
|      | PGSS                     | (3, 1)                           | 2.1344 \times 10^{-4} | 1.48         | 7        | 1       |
| 20   | –                        | –                                | 2.0648 \times 10^{-4} | 9.24         | 11       | 71.5    |
|      | RHSS                     | 19                               | 3.4282 \times 10^{-4} | 8.92         | 9        | 1.2     |
|      | PGSS                     | (1, 2)                           | 1.5814 \times 10^{-4} | 3.27         | 7        | 1       |
| 32   | –                        | –                                | 3.9128 \times 10^{-4} | 264.06       | 35       | 80      |
|      | RHSS                     | 23                               | 5.4007 \times 10^{-4} | 119.75       | 10       | 1.3     |
|      | PGSS                     | (2, 4)                           | 5.0058 \times 10^{-4} | 29.61        | 7        | 1       |

Table 12: Numerical results of preconditioned inexact Newton methods for \( v = 1 \) and \( \eta = 0.1 \).

| \( n \) | Preconditioner for GMRES | Optimal values of \( a/(\alpha, \beta) \) | Error estimates | CPU time (s) | Outer IT | Inner IT |
|------|--------------------------|----------------------------------|----------------|-------------|----------|---------|
| 16   | –                        | –                                | 1.2584 \times 10^{-4} | 2.23         | 8        | 53.9    |
|      | RHSS                     | 21                               | 2.0586 \times 10^{-4} | 2.93         | 7        | 1.3     |
|      | PGSS                     | (1, 4)                           | 1.3197 \times 10^{-4} | 1.09         | 5        | 1       |
| 20   | –                        | –                                | 1.1186 \times 10^{-4} | 9.14         | 10       | 78.8    |
|      | RHSS                     | 18                               | 2.6957 \times 10^{-4} | 8.32         | 7        | 1.6     |
|      | PGSS                     | (3, 2)                           | 1.7341 \times 10^{-4} | 2.7          | 5        | 1       |
| 32   | –                        | –                                | 3.9128 \times 10^{-4} | 265.38       | 35       | 80      |
|      | RHSS                     | 23                               | 2.8212 \times 10^{-4} | 121.84       | 8        | 1.9     |
|      | PGSS                     | (2, 9)                           | 3.2145 \times 10^{-4} | 26.51        | 5        | 1       |

much better than Newton-RHSS in the number of generation steps.

As we known, Krylov subspace method is more efficient than the stationary iterative methods in saddle point. Secondly, we will compare the effects of PGSS and RHSS as preconditioners on Newton-GMRES. In Tables 7–12, we can find it that PGSS and RHSS are more efficient as preprocessing operators than without them as using GMRES methods. Furthermore the PGSS is more efficient than RHSS as preconditioners. In the inner iteration, RHSS and PGSS are treated as preprocessing operators, and then the Krylov subspace method is used to solve the problem, which is better than the Krylov subspace method in CPU and step number. Although the effect of PGSS as preconditioner is not much better than that of RHSS when \( n \) is small, it can be seen that PGSS has great advantages in both steps and CPU compared with RHSS with the increase of \( n \).

8. Conclusions

The Newton-PGSS method is a considerable method for solving large sparse nonlinear system with nonsymmetric saddle point problems with the nonsymmetric Jacobian matrix. This is the first time to solve this kind of problem, and we utilize the PGSS iteration as the inner solver for the Newton equation. And, we establish a modified Newton-PGSS method for solving large sparse nonlinear system with nonsymmetric saddle point problems with the nonsymmetric Jacobian matrix. We give the local convergence and semilocal convergence analysis of the new method under proper conditions. Finally, the numerical results show that the modified Newton-PGSS outperforms the other splitting method in the sense of CPU time and iterative steps. Furthermore, when we apply the Newton-GMRES method to solve the problems, PGSS will accelerate the algorithm as preconditioner and make it more efficient than RHSS.
Data Availability
No data were used to support this study.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (Grant nos. 11771393 and 11632015).

References
[1] Y. Cao, S. Li, and L.-Q. Yao, “A class of generalized shift-splitting preconditioners for nonsymmetric saddle point problems,” Applied Mathematics Letters, vol. 49, pp. 20–27, 2015.
[2] Y. Cao, “Regularized DPSS preconditioners for non-Hermitian saddle point problems,” Applied Mathematics Letters, vol. 84, pp. 96–102, 2018.
[3] Z.-Z. Bai, “On spectral clustering of HSS preconditioner for generalized saddle-point matrices,” Linear Algebra and Its Applications, vol. 555, pp. 285–300, 2018.
[4] K. Wang, J. Di, and D. Liu, “Improved PHSS iterative methods for solving saddle point problems,” Numerical Algorithms, vol. 71, no. 4, pp. 753–773, 2016.
[5] Y. Dou, A.-L. Yang, and Y.-J. Wu, “Modified parameterized inexact Uzawa method for singular saddle-point problems,” Numerical Algorithms, vol. 72, no. 2, pp. 325–339, 2016.
[6] D. K. Salkuyeh, “Shifted skew-symmetric/skew-symmetric splitting method and its application to generalized saddle point problems,” Applied Mathematics Letters, vol. 103, pp. 106–114, 2020.
[7] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, NY, USA, 1970.
[8] D. K. Salkuyeh and M. Masoudi, “A new relaxed HSS preconditioner for saddle point problems,” Numerical Algorithms, vol. 74, no. 3, pp. 781–795, 2017.
[9] W. C. Rheinboldt, “A unified convergence theory for a class of iterative processes,” SIAM Journal on Numerical Analysis, vol. 5, no. 1, pp. 42–63, 1968.
[10] A.-L. Yang, “Scaled norm minimization method for computing the parameters of the HSS and the two-parameter HSS preconditioners,” Numerical Linear Algebra with Applications, vol. 25, no. 4, Article ID e2169, 2018.
[11] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, “Inexact Newton methods,” SIAM Journal on Numerical Analysis, vol. 19, no. 2, pp. 400–408, 1982.
[12] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, PA, USA, 2nd edition, 2003.
[13] W. C. Rheinboldt, Methods of Solving Systems of Nonlinear Equations, SIAM, Philadelphia, PA, USA, 2nd edition, 1998.
[14] H.-B. An and Z.-Z. Bai, “A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations,” Applied Numerical Mathematics, vol. 57, no. 3, pp. 235–252, 2007.
[15] H.-B. An, Z.-Y. Mo, and X.-P. Liu, “A choice of forcing terms in inexact Newton method,” Journal of Computational and Applied Mathematics, vol. 200, no. 1, pp. 47–60, 2007.
[16] D. A. Knoll and D. E. Keyes, “Jacobian-free Newton-Krylov methods: a survey of approaches and applications,” Journal of Computational Physics, vol. 193, no. 2, pp. 357–397, 2004.
[17] Q.-Q. Shen and Q. Shi, “Generalized shift-splitting preconditioners for nonsingular and singular generalized saddle point problems,” Computers & Mathematics with Applications, vol. 72, no. 3, pp. 632–641, 2016.
[18] Y. Cao, Z.-Q. Shi, and Q. Shi, “Regularized DPSS preconditioners for generalized saddle point linear systems,” Computers & Mathematics with Applications, vol. 80, no. 5, pp. 956–972, 2020.
[19] Z.-G. Huang, L.-G. Wang, Z. Xu, and J.-J. Cui, “Improved PPHSS iterative methods for solving nonsingular and singular saddle point problems,” Computers & Mathematics with Applications, vol. 72, no. 1, pp. 92–109, 2016.
[20] X.-P. Guo, “On semilocal convergence of inexact Newton methods,” Journal of Computational Mathematics, vol. 25, pp. 231–242, 2007.
[21] S.-W. Zhou, A.-L. Yang, Y. Dou, and Y.-J. Wu, “The modified shift-splitting preconditioners for nonsymmetric saddle-point problems,” Applied Mathematics Letters, vol. 59, pp. 109–114, 2016.
[22] J. Zhou, S. Zhang, G. Yang, and J. Tan, “A convergence theorem for the inexact Newton methods based on Hölder continuous Fréchet derivative,” Applied Mathematics and Computation, vol. 197, no. 1, pp. 206–211, 2008.
[23] Q. Zheng and L. Lu, “Extended shift-splitting preconditioners for saddle point problems,” Journal of Computational and Applied Mathematics, vol. 313, pp. 70–81, 2017.
[24] Y. Xu and L. Yang, “Optimization of a parameterized inexact Uzawa method for saddle point problems,” International Journal of Computer Mathematics, vol. 95, no. 12, pp. 2539–2548, 2017.
[25] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, “Analysis of the inexact Uzawa algorithm for saddle point problems,” SIAM Journal on Numerical Analysis, vol. 34, no. 3, pp. 1072–1092, 1997.
[26] Y. Cao, M.-Q. Jiang, and Y.-L. Zheng, “A splitting preconditioner for saddle point problems,” Numerical Linear Algebra with Applications, vol. 18, no. 5, pp. 875–895, 2011.
[27] Y. Cao, J. Du, and Q. Niu, “Shift-splitting preconditioners for saddle point problems,” Journal of Computational and Applied Mathematics, vol. 272, pp. 239–250, 2014.
[28] Y. Cao, S.-X. Miao, and Z.-R. Ren, “On preconditioned generalized shift-splitting iteration methods for saddle point problems,” Computers & Mathematics with Applications, vol. 74, no. 4, pp. 859–872, 2017.
[29] Y. Cao, Z. Zhen, and L. Yao, “Improved relaxed positive-definite and skew-Hermitian splitting preconditioners for saddle point problems,” Journal of Computational Mathematics, vol. 37, pp. 95–111, 2018.
[30] Z.-Z. Bai, J.-F. Yin, and Y.-F. Su, “A shift-splitting preconditioner for non-Hermitian positive definite matrices,” Journal of Computational Mathematics, vol. 24, pp. 539–552, 2006.
[31] G. H. Golub, X. Wu, and J.-Y. Yuan, “SOR-like methods for augmented systems,” BIT Numerical Mathematics, vol. 41, no. 1, pp. 71–85, 2001.
[32] Z.-Z. Bai, B. N. Parlett, and Z.-Q. Wang, “On generalized successive overrelaxation methods for augmented linear systems,” Numerische Mathematik, vol. 102, no. 1, pp. 1–38, 2005.
[33] Z.-Z. Bai and G. H. Golub, “Accelerated hermitian and skew-hermitian splitting iteration methods for saddle-point problems,” IMA Journal of Numerical Analysis, vol. 27, no. 1, pp. 1–23, 2007.
[34] Z.-Z. Bai, “Regularized HSS iteration methods for stabilized saddle-point problems,” *IMA Journal of Numerical Analysis*, vol. 39, no. 4, pp. 1888–1923, 2019.

[35] M. Chen, Q. Wu, and R. Lin, “Semilocal convergence analysis for the modified Newton-HSS method under the Hölder condition,” *Numerical Algorithms*, vol. 72, no. 3, pp. 667–685, 2016.

[36] F. Xie, R. F. Lin, and Q. B. Wu, “Modified Newton-DSS method for solving a class of systems of nonlinear equations with complex symmetric Jacobian matrices,” *Numerical Algorithms*, vol. 19, pp. 1–25, 2019.

[37] M.-H. Chen and Q.-B. Wu, “On modified Newton-DGPMHSS method for solving nonlinear systems with complex symmetric Jacobian matrices,” *Computers & Mathematics with Applications*, vol. 76, no. 1, pp. 45–57, 2018.