Nonlinearly preconditioned semismooth Newton algorithms for nonlinear nonsmooth systems

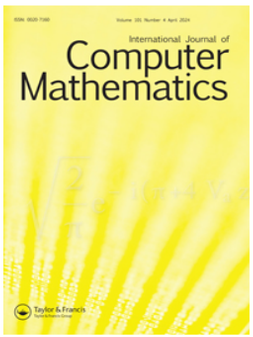
Article *in* International Journal of Computer Mathematics · May 2024

DOI: 10.1080/00207160.2024.2360026

4 authors, including:

Haijian Yang
Hunan University
49 PUBLICATIONS **402** CITATIONS
SEE PROFILE

Feng-Nan Hwang
National Central University
63 PUBLICATIONS **576** CITATIONS
SEE PROFILE



International Journal of Computer Mathematics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/gcom20

Nonlinearly preconditioned semismooth Newton algorithms for nonlinear nonsmooth systems

Haijian Yang, Tian-Hao Ma, Feng-Nan Hwang & Xiao-Chuan Cai

To cite this article: Haijian Yang, Tian-Hao Ma, Feng-Nan Hwang & Xiao-Chuan Cai (30 May 2024): Nonlinearly preconditioned semismooth Newton algorithms for nonlinear nonsmooth systems, International Journal of Computer Mathematics, DOI: 10.1080/00207160.2024.2360026

To link to this article: https://doi.org/10.1080/00207160.2024.2360026

Published online: 30 May 2024.

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS
https://doi.org/10.1080/00207160.2024.2360026

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Nonlinearly preconditioned semismooth Newton algorithms for nonlinear nonsmooth systems

Haijian Yang[a,b], Tian-Hao Ma[c], Feng-Nan Hwang[d] and Xiao-Chuan Cai[c]

[a]Research Institute of Hunan University In Chongqing, Chongqing, China; [b]School of Mathematics, Human University, Changsha, P.R. China; [c]Department of Mathematics, University of Macau, Macau, China; [d]Department of Mathematics, National Central University, Taoyuan City, Taiwan

ABSTRACT

We aim to develop efficient and robust algorithms for nonsmooth nonlinear systems arising from complementarity problems. The semismooth Newton algorithm is popular due to its reliability and efficiency. However, it struggles with issues with imbalanced nonlinearities of the problems, leading to degraded convergence rates or failure despite help from the globalization techniques like linesearch or trust region. We introduce a right nonlinearly preconditioned semismooth Newton algorithm to address this difficulty. The critical success ingredient is that before each global Newton update, a nonlinear preconditioning step implicitly removes the so-called 'bad components' causing trouble via nonlinear subspace correction, inspired by Gaussian elimination but adapted nonlinearly to balance system nonlinearities. Additionally, our method integrates with a domain decomposition framework, enhancing parallelism. Numerical results on two classes of problems demonstrate significantly improved convergence over standard semismooth Newton methods.

ARTICLE HISTORY
Received 28 December 2023
Revised 8 April 2024
Accepted 20 May 2024

KEYWORDS
Semismooth Newton algorithm; nonlinear preconditioning; flow control; complementarity problems; parallel computing

2020 MATHEMATICS SUBJECT CLASSIFICATIONS
49M40; 65K10

1. Introduction

In this paper, we consider the large, sparse, nonsmooth nonlinear system of algebraic equations

$$\mathcal{F}(\mathbf{x}) = 0, \tag{1}$$

where $\mathcal{F}$ is a Lipschitz function in $\mathbb{R}^n$. In particular, we are interested in nonlinear complementarity problems (NCP) [12], which are closely related to variational inequality problems [17], and both problems [21] can be reformulated as systems of nonsmooth equations [40]. Thus, developing fast and robust methods for such problems is important, especially methods that run on large parallel computers. Generally speaking, the family of semismooth Newton methods [22] is a good choice since it is easy to implement and offers rapid local convergence when the user-provided initial guess is close to the solution. To describe an extension of Newton-type methods for the nonsmooth system (1), we begin by introducing some notations. Let $\|\cdot\|$ denote the Euclidean norm throughout the paper unless otherwise explicitly stated.

CONTACT Feng-Nan Hwang hwangf@math.ncu.edu.tw Department of Mathematics, National Central University, Jhongli District, Taoyuan City 320317, Taiwan

Definition 1.1 ([22]): A locally Lipschitzian function $\mathcal{F}: D \subset \mathbb{R}^n \to \mathbb{R}^n$ is called B-differentiable at $\mathbf{x} \in D$ if it is directionally differentiable in the sense that

$$\lim_{\|h\|\to 0} \frac{1}{\|h\|} \|\mathcal{F}(\mathbf{x}+h) - \mathcal{F}(\mathbf{x}) - \mathcal{F}'(\mathbf{x};h)\| = 0 \tag{2}$$

for every $\mathbf{x} \in D$. Here, $\mathcal{F}'(\mathbf{x};h): D \subset \mathbb{R}^n \to \mathbb{R}^n$ is referred to as a directional derivative of $\mathcal{F}$ in D and it is not necessarily unique.

Note that Rademacher's Theorem [43] states a local Lipschitzian function on $\mathbb{R}^n$ is differentiable almost everywhere. Let us define $D_{\mathcal{F}}$ as the set of points where $\mathcal{F}(\mathbf{x})$ is differentiable. Then we construct the generalized Jacobian matrix using the B-subdifferential [12,16,44],

$$\partial_B \mathcal{F}(\mathbf{x}) = \left\{ A \in \mathbb{R}^{n\times n} \,\middle|\, \exists \{\mathbf{x}^{(k)}\} \subseteq D_{\mathcal{F}}, \text{ with } \lim_{\mathbf{x}^{(k)}\to \mathbf{x}} \nabla \mathcal{F}(\mathbf{x}^{(k)}) = A \right\}. \tag{3}$$

A directionally differentiable $\mathcal{F}$ is said to be semismooth at $\mathbf{x}$ if $Vh - \mathcal{F}'(\mathbf{x};h) = o(\|h\|)$ as $\|h\| \to 0$ for $V \in \partial_B \mathcal{F}(\mathbf{x})$. Typically, the following locally superlinear convergence rate of Newton methods can be obtained under appropriate semismoothness and regularity assumptions [22,43,44].

Theorem 1.2: *Let $\mathbf{x}^*$ be a solution to* (1) *and suppose that $\mathcal{F}$ is B-differentiable in an open neighborhood D containing $\mathbf{x}^*$ with the generalized Jacobian $\mathcal{J}(\mathbf{x}) \in \partial_B \mathcal{F}$. If $\mathcal{J}(\mathbf{x})$ is nonsingular for all $\mathbf{x} \in U$ and $\{\|\mathcal{J}(\mathbf{x})^{-1}\|; \mathbf{x} \in U\}$ is bounded, then the Newton iteration*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathcal{J}(\mathbf{x}^{(k)})^{-1} \mathcal{F}(\mathbf{x}^{(k)}) \tag{4}$$

converges superlinearly to $\mathbf{x}^*$, provided that $\|\mathbf{x}^{(0)} - \mathbf{x}^*\|$ is sufficiently small.

If we apply a Newton-type method with a generalized Jacobian in (3) to the system (1), the method is called a semismooth Newton (SN) method. In the method, we first find a direction by solving the Jacobian system and then determine an appropriate step size along this search direction to obtain a new iterate that sufficiently decreases the merit function value

$$\Psi(\mathbf{x}) := \frac{1}{2} \mathcal{F}(\mathbf{x})^T \mathcal{F}(\mathbf{x}). \tag{5}$$

An extension of the globalization linesearch technique from smooth to nonsmooth equations is not straightforward, as the merit function Ψ defined in Equation (5) is usually not differentiable when $\mathcal{F}$ is nonsmooth. However, in some specific cases, such as nonlinear complementarity problems, reformulating the complementarity conditions through the Fischer-Burmeister function leads to a differentiable Ψ due to the special properties of the Fischer-Burmeister function [30]. Assume that $\Phi(\mathbf{x})$ is continuously differentiable. Given an initial guess $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and let $\mathbf{x}^{(k)}$ be the current approximation at the k^{th} SN iteration. The SN method [4,37,45] consists of three key steps for finding the next approximation $\mathbf{x}^{(k+1)}$ as follows.

- Determine the inexact Newton direction $d^{(k)}$ by approximately solving the Jacobian system

$$\mathcal{F}(\mathbf{x}^{(k)}) + \mathcal{J}_k d^{(k)} = 0,$$

  where $\mathcal{J}_k \in \partial_B \mathcal{F}(\mathbf{x})$ is a generalized Jacobian matrix.
- Compute a step length γ such that the Armijo condition is satisfied, i.e.

$$\Psi\left(\mathbf{x}^{(k)} + \gamma\, d^{(k)}\right) \le \Psi(\mathbf{x}^{(k)}) + \sigma\gamma \left(\nabla \Psi(\mathbf{x}^k)\right)^T d^{(k)}, \tag{6}$$

  and set $\alpha = \gamma$ and $\sigma \in (0, 1]$ is a prescribed parameter assuring the sufficient descent.

- Compute the new approximate solution $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha d^{(k)}$, and set $k = k+1$.

We continue the nonlinear iteration until the following convergence criterion is satisfied.

$$\|\mathcal{F}(\mathbf{x}^{(k)})\| \leq \max\{\varepsilon_r \|\mathcal{F}(\mathbf{x}^{(0)})\|, \varepsilon_a\},$$

where ε_r (ε_a) is the relative (absolute) solver tolerance for the Newton iteration.

In the Newton method, α is the damping factor that tells us how far we should go in the selected search direction. As stated in Theorem 1.2, a superlinear convergence rate may be achieved when the initial guess is close enough to the desired solution. Nonetheless, for problems with high nonlinearity or for problems in which the initial guess is far from the solution, the Newton direction may not be descent; thus, the Newton step is unacceptable, and it converges slowly or simply stagnates. In this paper, we employ a nonlinear preconditioning technique to reduce the high nonlinearities that slow down the convergence of Newton iterations. Like linear cases, nonlinear preconditioning is a powerful technique for accelerating the convergence rate of the iterative methods suffering from slow convergence. This acceleration is achieved by reducing the number of iterations required for convergence while imposing minimal overhead for the construction and application of preconditioning. More importantly, when well-designed, nonlinear preconditioning significantly enhances the robustness of these iterative methods, particularly those with local convergence properties such as Newton-type methods.

Here, we briefly review developments in nonlinear preconditioning methods. The first such algorithm is known as the additive Schwarz preconditioned inexact Newton algorithm (ASPIN), in which the function of the system is implicitly changed to a more nonlinearly balanced system, and it has been applied successfully to solve some rather complex nonlinear problems in computational fluid dynamics [8,10,24]. Later, Liu and Keyes and Dolean et al. further developed the multiplicative version (i.e. MSPIN [34]) and the restricted variation (i.e. RASPEN [14]) of the domain decomposition based nonlinearly preconditioned Newton algorithm, respectively. The local convergence analysis of ASPIN and MSPIN is established by An [2] and Liu and Keyes [34], respectively. Under reasonable assumptions, they showed that ASPIN or MSPIN is locally convergent, and the desired superlinear or quadratic convergence can be achieved when the forcing terms are picked appropriately. In addition, the multilevel ASPIN methods were proposed in [9,25], where nonlinear or linear coarse space is added to enhance the scalability of the algorithm. To further improve ASPIN's efficiency, Liu et al. [35] suggested some practical mechanisms to turn off the nonlinear preconditioning step when it is not needed, for example, when the intermediate solution enters the convergence region. In such a situation, the inexact Newton with preconditioning is expected to converge quadratically.

A class of different but closely related algorithms is called the right nonlinear preconditioning technique, including the nonlinear elimination (NE) algorithm [26,27], the nonlinear restricted additive Schwarz algorithm (NKS-RAS) [11], the nonlinear finite element tearing and interconnecting (FETI) method [31]. The right preconditioning method is applied by modifying the variables of the nonlinear function. Compared with the left nonlinear preconditioner, the right one is more attractive and more accessible to implement because it does not change the nonlinear function. Therefore, more sophisticated and efficient solvers can be easily applied to the unchanged system. Among them, the NE approach has attracted more attention since it can handle the imbalance of the nonlinearity by removing variables deemed to cause trouble for the global Newton iterations [35]. Combining with different partitioning strategies leads to the family of domain-based NE algorithms [27,47] and the field-based approaches [18,48], and they have been employed successfully to solve some problems that are challenging for inexact Newton. In addition to model problems defined on simple geometry, in [18,36], Cai and his coworkers demonstrated the capability of NE to complicated 3D fluid flow and structure problems for applications to patient-specific blood flows in the human artery. Recently, Liu et al. [33] proposed a new algorithm, namely the nonlinear elimination preconditioned inexact Newton (NEPIN) algorithm, where NE was used as a left preconditioner for inexact Newton

methods. They showed that left and right NE preconditioned inexact Newton methods have similar convergence behavior.

One of our target applications is the optimal flow control problem [19], reformulated as a nonsmooth system in the form of NCP from the PDE-constrained optimization problem. The main challenge for SN arises from the additional sharp jumps in the residual function near the nonsmooth regions. The stagnation issue becomes worse for problems under consideration in this paper due to the more complicated dynamics of the flow mechanism. This paper aims to develop a NE-based preconditioner for improving the convergence of SN as a numerical solution of nonsmooth systems arising from NCP [45]. Related works [49,50] studied a field-based component-wise or adaptive region-based nonlinear elimination preconditioner for SN used in the numerical simulation of porous media fluid flows, where the governing equation is formulated as variational inequality or NCP forms to guarantee the numerical solution all make physical sense.

The rest of this paper is organized as follows. Section 2 reviews the general framework of a right nonlinear elimination-based preconditioning technique for the Newton-type method. In Section 3, we give a detailed description of two applications, which can be reformulated as large, sparse, nonsmooth, nonlinear systems of equations. We provide some numerical results in Section 4 and end the paper with some concluding remarks in Section 5.

2. Nonlinear preconditioning technique for Newton-type method

This section first reviews a general framework for the right nonlinear preconditioned system. Then, we summarize the desired properties of the right preconditioned system and the corresponding nonlinear preconditioner, analogous to the left version of the nonlinear preconditioned system [8]. Such properties provide a guideline for designing a nonlinear preconditioner. Finally, we describe a solution algorithm for solving the right preconditioned system with an inexact Newton method, including deriving the linesearch technique based on the Armijo condition for the preconditioned systems.

2.1. Nonlinear preconditioner and its properties

Consider a right nonlinear preconditioned system

$$W(\mathbf{y}) \equiv F\left(G(\mathbf{y})\right) = 0, \tag{7}$$

where

$$\mathbf{x} = G(\mathbf{y})$$

and the operator G is referred to as a *right* preconditioner for the original system,

$$F(\mathbf{x}) = 0,$$

where G: $\mathbb{R}^n \to \mathbb{R}^n$ and W: $\mathbb{R}^n \to \mathbb{R}^n$ are two continuously differentiable nonlinear mappings. We hope the nonlinear preconditioner, G, and the preconditioned system (7) have the following properties.

(1) G is a fixed-point mapping, where its fixed-point $\mathbf{x}^*$ is the solution of $F(\mathbf{x}) = 0$. This property guarantees that the nonlinearly preconditioned problem has the same solution as the original problem.
(2) $W(\mathbf{y})$ is more nonlinearly balanced than $F(\mathbf{x})$ in some sense.
(3) The cost for performing the nonlinear preconditioning operation, $\mathbf{x} = G(\mathbf{y})$, for a given $\mathbf{y} \in \mathbb{R}^n$, is much lower than the one for evaluating the original function $F(\mathbf{x})$.

Let $\mathbf{y}^{(k)}$ be the current approximate solution, an inexact Newton with backtracking (INB) method for the preconditioned system (7) provides a new approximation

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha \Delta \mathbf{y}^{(k)}, \tag{8}$$

where $\Delta y^{(k)}$ is obtained by solving the following Jacobian system

$$\frac{\partial W\left(\mathbf{y}^{(k)}\right)}{\partial \mathbf{y}} \Delta \mathbf{y}^{(k)} = -W(\mathbf{y}^{(k)}) \tag{9}$$

and $\alpha \in [0, 1)$ is a damping scalar. An extension to the nonsmooth case will be discussed later.

2.2. Right nonlinearly preconditioned Newton algorithm

Solving the preconditioned system (7) by Steps (9) and (8) of INB is not practical and has the following potential drawbacks. First, the operator $G(\mathbf{y})$ is implicitly defined so that the Jacobian of $W(\mathbf{y})$ is hard to obtain by using the chain rule, $\frac{\partial W}{\partial \mathbf{y}} = \frac{\partial F}{\partial \mathbf{x}} \frac{\partial G}{\partial \mathbf{y}}$. Moreover, designing an efficient preconditioner for the Jacobian system in the composite form in (9) is quite challenging. Hence, in this paper, we suggest the right nonlinearly preconditioned INB as

$$\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \Delta \mathbf{x}^{(k)}, \tag{10}$$

where

$$\frac{\partial F(\mathbf{x}^{(k)})}{\partial \mathbf{x}} \Delta \mathbf{x}^{(k)} = -F(\mathbf{x}^{(k)}), \quad \text{with } \mathbf{x}^{(k)} = G(\mathbf{y}^{(k)}). \tag{11}$$

Equation (11) is derived from Equation (9) as

$$\frac{\partial F(G(\mathbf{y}^{(k)}))}{\partial \mathbf{x}} \frac{\partial G(\mathbf{y}^{(k)})}{\partial \mathbf{y}} \Delta \mathbf{y} = -F(G(\mathbf{y}^{(k)})), \tag{12}$$

by letting $\Delta \mathbf{x}^{(k)} = \frac{\partial G(\mathbf{y}^{(k)})}{\partial \mathbf{y}} \Delta \mathbf{y}^{(k)}$ and $\mathbf{x}^{(k)} = G(\mathbf{y}^{(k)})$. On the other hand, the right-hand side of (10) is obtained by using the first-order Taylor's expansion of $G(\mathbf{y}^{(k)} + \alpha \Delta \mathbf{y})$ around $\mathbf{y}^{(k)}$ as

$$\begin{aligned} x^{(k)} + \alpha \Delta x^{(k)} &= G(\mathbf{y}^{(k)}) + \alpha \frac{\partial G(\mathbf{y}^{(k)})}{\partial \mathbf{y}} \Delta \mathbf{y}^{(k)} \\ &\approx G(\mathbf{y}^{(k)} + \alpha \Delta \mathbf{y}^{(k)}) \end{aligned}$$

Note that this approximation introduces a truncation error expected to be $O((\Delta \mathbf{y})^2)$, where $O()$ is denoted by a big O notation.

We can guarantee that the limit point of $\{\mathbf{y}^{(k)}\}$ denoted by $\mathbf{y}^*$ such that $F(G(\mathbf{y}^*)) = 0$ is also the root of $F(\mathbf{x}) = 0$ as follows. Taking the limit for both sides of (10), we have

$$\mathbf{y}^* = G(\mathbf{y}^*) - \alpha \left(\frac{\partial F}{\partial \mathbf{x}} (G(\mathbf{y}^*)) \right)^{-1} F(G(\mathbf{y}^*)).$$

Since $\mathbf{y}^* = G(\mathbf{y}^*)$ by design, we have $F(G(\mathbf{y}^*)) = 0$, which implies $F(\mathbf{y}^*) = 0$.

Next, we introduce a merit function for the nonlinearly preconditioned function, which is needed for the globalization technique, such as linesearch or trust region approaches. Let $w(\mathbf{y}) = \frac{1}{2}\|W(\mathbf{y})\|^2 = \frac{1}{2}\|F(G(\mathbf{y}))\|^2 = \frac{1}{2}\|F(\mathbf{x})\|^2 = f(\mathbf{x})$. Note that $\nabla w(\mathbf{y}) = J_w^T(\mathbf{y}) W(\mathbf{y})$, then the Armijo condition for w takes the form

$$w(\mathbf{y}^{(k)} + \alpha \Delta \mathbf{y}^{(k)}) \leq w(\mathbf{y}^{(k)}) + \sigma \alpha (\nabla w(\mathbf{y}^{(k)}))^T \Delta \mathbf{y}^{(k)}$$

$$
\begin{aligned}
&= f(\mathbf{x}^{(k)}) + \sigma\alpha\mathcal{W}^T(\mathbf{y}^{(k)})J_w(\mathbf{y}^{(k)})\Delta\mathbf{y}^{(k)} \\
&= f(\mathbf{x}^{(k)}) + \sigma\alpha\mathcal{W}^T(\mathbf{y}^{(k)})\frac{\partial F}{\partial G}\frac{\partial G}{\partial \mathbf{y}}\Delta\mathbf{y}^{(k)} \\
&= f(\mathbf{x}^{(k)}) + \sigma\alpha(\nabla f(\mathbf{x}^{(k)}))^T\Delta\mathbf{x}^{(k)}.
\end{aligned}
$$

The left-hand side of the above inequality is

$$
\begin{aligned}
w(\mathbf{y}^{(k)} + \alpha\Delta\mathbf{y}^{(k)}) &= \frac{1}{2}\|\mathcal{W}(\mathbf{y}^{(k)} + \alpha\Delta\mathbf{y}^{(k)})\|^2 = \frac{1}{2}\|F(G(\mathbf{y}^{(k)} + \alpha\Delta\mathbf{y}^{(k)}))\|^2 \\
&\approx \frac{1}{2}\|F(G(\mathbf{y}^{(k)}) + \alpha\frac{\partial G}{\partial \mathbf{y}}\Delta\mathbf{y}^{(k)}))\|^2 \\
&= \frac{1}{2}\|F(\mathbf{x}^{(k)} + \alpha\Delta\mathbf{x}^{(k)})\|^2 \\
&= f(\mathbf{x}^{(k)} + \alpha\Delta\mathbf{x}^{(k)}).
\end{aligned}
$$

This is true if α is sufficiently small. As a result, the Armijo condition can be realized in any of the following formats.

(1) Find $\alpha \in (0,1]$ for $\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \alpha\Delta\mathbf{x}^{(k)}$ such that $f(\mathbf{x}^{(k)} + \alpha\Delta\mathbf{x}^{(k)}) \leqslant (1-\sigma\alpha)f(\mathbf{x}^{(k)})$ [27].
(2) Find $\alpha \in (0,1]$ for $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha\Delta\mathbf{y}^{(k)}$ such that $w(\mathbf{y}^{(k)} + \alpha\Delta\mathbf{y}^{(k)}) \leqslant (1-\sigma\alpha)w(\mathbf{x}^{(k)})$ [26].

3. Nonlinearly preconditioned semismooth Newton algorithm for nonlinear complementarity problems

In this section, we apply NPSN to two important classes of problems, namely, an NCP and the reformulation of an optimization problem with equality-inequality constraints as NCP and a series of illustrative examples in $\mathbb{R}^2$. We first discuss the details of the nonlinear preconditioner in conjunction with SN for solving (1), which has n functions $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_n)^T$ and n unknowns, $\mathbf{x} = (x_1, \ldots, x_n)^T$.

3.1. Nonlinear complementarity problem

The NCP [17,21] is stated as follows. Find a solution $\mathbf{x} \in \mathbb{R}^n$ such that

$$
\mathbf{x} \geq \Phi, \quad F(\mathbf{x}) \geq 0, \quad \text{and} \quad (\mathbf{x} - \Phi)^T F(\mathbf{x}) = 0, \tag{13}
$$

where $\Phi = (\phi_1, \ldots, \phi_n)^T$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, and $F = (F_1, F_2, \ldots, F_n)^T$. Here, $F_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq n$, is assumed to be continuously differentiable. To solve (13), we first reformulate the problem as the following nonlinear system of equations

$$
\mathcal{F}(\mathbf{x}) = \begin{pmatrix} \varphi\big(x_1 - \phi_1, F_1(\mathbf{x})\big) \\ \vdots \\ \varphi\big(x_i - \phi_i, F_i(\mathbf{x})\big) \\ \vdots \\ \varphi\big(x_n - \phi_n, F_n(\mathbf{x})\big) \end{pmatrix} = 0, \tag{14}
$$

with a NCP function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ that satisfies

$$
\varphi(a,b) = 0 \Longleftrightarrow a \geq 0, \quad b \geq 0, \quad ab = 0.
$$

There are several choices for this function, but we focus on the Fischer-Burmeister NCP function [17,43], defined as

$$\varphi(a,b) := a + b - \sqrt{a^2 + b^2}. \tag{15}$$

Note that φ is differentiable everywhere except at the point $(a,b) = (0,0)$. Hence, we construct the generalized Jacobian matrix by using the B-subdifferential in (3), i.e.

$$\partial_B \mathcal{F}(\mathbf{x}) \subseteq \{D_a + D_b(\mathbf{x})\nabla F(\mathbf{x})\},$$

where nonnegative diagonal matrices

$$\begin{cases} D_a = \operatorname{diag}(d_{a_1}, \ldots, d_{a_n}), \\ D_b = \operatorname{diag}(d_{b_1}, \ldots, d_{b_n}), \end{cases} \tag{16}$$

consist of the partial derivatives of the mapping φ with respect to the first variable $a_i = x_i - \phi$ and the second variable $b_i = F_i(\mathbf{x})$, respectively, or a suitable approximation to these partial derivatives at those points where φ is not differentiable. More precisely, the elements of the matrices D_a and D_b in (16) take the following form

$$\begin{cases} (d_{a_i}, d_{b_i}) = \left(1 - \dfrac{a_i}{\sqrt{a_i^2 + b_i^2}}, 1 - \dfrac{b_i}{\sqrt{a_i^2 + b_i^2}}\right) & \text{if } a_i^2 + b_i^2 \neq 0, \\ (d_{a_i}, d_{b_i}) \in \left\{(1-\alpha_1, 1-\alpha_2) \,|\, \alpha_1^2 + \alpha_2^2 \leq 1\right\} & \text{otherwise,} \end{cases}$$

with $0 \leq \alpha_1 \leq 1$ and $0 \leq \alpha_2 \leq 1$.

3.2. Constrained optimization problem

Consider a general constrained optimization problem with both equality and inequality constraints [19,39] formulated as follows: Find the states $\mathbf{s} \in \mathbb{R}^{n_s}$ and $\mathbf{u} \in \mathbb{R}^{n_u}$ such that

$$\begin{array}{ll} \text{minimize} & \mathcal{J}(\mathbf{s},\mathbf{u}) \\ \text{subject to} & \mathbf{C}(\mathbf{s},\mathbf{u}) = \mathbf{0}, \\ & \mathbf{u}_l \leq \mathbf{u} \leq \mathbf{u}_r, \end{array} \tag{17}$$

where the state variable is classified as two types: $\mathbf{s}$ is the *free* variable and $\mathbf{u}$ is the variable with some restrictions imposed. $\mathcal{J}(\mathbf{s},\mathbf{u}) : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is an objective function and $\mathbf{C} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_c}$ satisfying $\mathbf{C}(\mathbf{s},\mathbf{u}) = 0$, which is a nonlinear system of equations derived from the discretization of some partial differential equation. By introducing the Lagrange multipliers $\lambda \in \mathbb{R}^{n_c}$, $\mu_l \in \mathbb{R}^{n_u}$, and $\mu_r \in \mathbb{R}^{n_u}$, we define the following Lagrangian functional

$$\mathcal{L}(\mathbf{s},\mathbf{u},\lambda,\mu_l,\mu_r) \equiv \mathcal{J}(\mathbf{s},\mathbf{u}) + (\lambda, \mathbf{C}(\mathbf{s},\mathbf{u})) + (\mathbf{u}_l - \mathbf{u}, \mu_l) + (\mathbf{u} - \mathbf{u}_r, \mu_r). \tag{18}$$

Then, the corresponding KKT system is obtained by differentiating (18), with respect to λ, $\mathbf{s}$, $\mathbf{u}$, μ_l, μ_r, respectively.

$$\begin{cases} \mathcal{L}_\lambda(\mathbf{s},\mathbf{u},\lambda,\mu_l,\mu_r) = \mathbf{C}(\mathbf{s},\mathbf{u}) = 0, \\ \mathcal{L}_\mathbf{s}(\mathbf{s},\mathbf{u},\lambda,\mu_l,\mu_r) = \mathcal{J}_\mathbf{s}(\mathbf{s},\mathbf{u}) + (\lambda, \mathbf{C}_\mathbf{s}(\mathbf{s},\mathbf{u})) = 0, \\ \mathcal{L}_\mathbf{u}(\mathbf{s},\mathbf{u},\lambda,\mu_l,\mu_r) = \mathcal{J}_\mathbf{u}(\mathbf{s},\mathbf{u}) + (\lambda, \mathbf{C}_\mathbf{u}(\mathbf{s},\mathbf{u})) - \mu_l + \mu_r = 0, \\ \mathbf{u}_l - \mathbf{u} \leq 0,\ \mu_l \geq 0,\ (\mathbf{u}_l - \mathbf{u}, \mu_l) = 0, \\ \mathbf{u} - \mathbf{u}_r \leq 0,\ \mu_r \geq 0,\ (\mathbf{u} - \mathbf{u}_r, \mu_r) = 0, \end{cases} \tag{19}$$

where $\mathcal{J}_\mathbf{s}(\mathbf{s},\mathbf{u})$ denotes the gradient of $\mathcal{J}$ with respect to the variable $\mathbf{s}$, and the others are defined in a similar way. The last three equations in (19) form NCP, which is equivalent to the following system

by eliminating the Lagrange multipliers μ_l and μ_r

$$\begin{cases} \mathbf{u} = \mathbf{u}_l & \& \quad \mathcal{L}_{\mathbf{u}}(\mathbf{s},\mathbf{u},\lambda) = \mathcal{J}_{\mathbf{u}}(\mathbf{s},\mathbf{u}) + (\lambda, \mathbf{C}_{\mathbf{u}}(\mathbf{s},\mathbf{u})) \geq 0, \\ \mathbf{u} = \mathbf{u}_r & \& \quad \mathcal{L}_{\mathbf{u}}(\mathbf{s},\mathbf{u},\lambda) = \mathcal{J}_{\mathbf{u}}(\mathbf{s},\mathbf{u}) + (\lambda, \mathbf{C}_{\mathbf{u}}(\mathbf{s},\mathbf{u})) \leq 0, \\ \mathbf{u} \in (\mathbf{u}_l, \mathbf{u}_r) & \& \quad \mathcal{L}_{\mathbf{u}}(\mathbf{s},\mathbf{u},\lambda) = \mathcal{J}_{\mathbf{u}}(\mathbf{s},\mathbf{u}) + (\lambda, \mathbf{C}_{\mathbf{u}}(\mathbf{s},\mathbf{u})) = 0, \end{cases} \tag{20}$$

with only one of these three cases holding at a time.

Let the solution vector $\mathbf{x}$ be organized by $\mathbf{x} = (\mathbf{s},\mathbf{u},\lambda) = (x_1,\ldots,x_n) \in \mathbb{R}^n$ and the corresponding function be defined by

$$F(\mathbf{x}) \equiv (\mathcal{L}_{\mathbf{s}}(\mathbf{x}), \mathcal{L}_{\mathbf{u}}(\mathbf{x}), \mathcal{L}_{\lambda}(\mathbf{x}))^T \in \mathbb{R}^n.$$

Then, the variational inequality problem for the KKT system of (17) is defined as follows: find $\mathbf{x} \in \mathbb{R}^n$ such that one of the following conditions holds for all $i \in \{1,2,\ldots n\}$,

$$\begin{cases} x_i = \phi_i & \& \quad F_i(\mathbf{x}) \geq 0, \\ x_i = \psi_i & \& \quad F_i(\mathbf{x}) \leq 0, \\ x_i \in (\phi_i, \psi_i) & \& \quad F_i(\mathbf{x}) = 0, \end{cases} \tag{21}$$

where the lower- and upper-bound vectors for the solution $\mathbf{x}$ are defined by

$$\begin{cases} \Phi = (-\hat{\mathbf{s}}, \mathbf{u}_l, -\hat{\lambda}) = (\phi_1, \phi_2, \ldots, \phi_n) \in \mathbb{R}^n, \\ \Psi = (\hat{\mathbf{s}}, \mathbf{u}_r, \hat{\lambda}) = (\psi_1, \psi_2, \ldots, \psi_n) \in \mathbb{R}^n, \end{cases}$$

where $\hat{\mathbf{s}}$ and $\hat{\lambda}$ are the prescribed vectors whose components are chosen to be some sufficiently large numbers. We remark that the inequality holds componentwisely by ϕ and ψ, i.e. only one of these three equations holds at a time. Moreover, if the upper bound vector $\psi = -\infty$, (21) is reduced to the complementarity problem (13).

Similar to the complementarity problem (13), by using the Fischer-Burmeister function (15), we reformulate the variational inequality problem (21) as $\mathcal{F}(\mathbf{x}) = 0$, where $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$ is expressed by

$$\mathcal{F}(\mathbf{x}) := \begin{pmatrix} \varphi\big(x_1 - \phi_1, -\varphi\left(\psi_1 - x_1, -F_1(\mathbf{x})\right)\big) \\ \vdots \\ \varphi\big(x_i - \phi_i, -\varphi\left(\psi_i - x_i, -F_i(\mathbf{x})\right)\big) \\ \vdots \\ \varphi\big(x_n - \phi_n, -\varphi\left(\psi_n - x_n, -F_n(\mathbf{x})\right)\big) \end{pmatrix}. \tag{22}$$

And we can define the corresponding generalized Jacobian matrices by using the B-subdifferential in (3) with respect to (22) similarly. The optimality problem is a large, nonlinear, coupled, and muti-component system with an indefinite and often ill-conditioned Jacobian matrix. Hence, we use NPSN to deal with the difficulties.

3.3. Nonlinear elimination preconditioner

To construct the nonlinear elimination preconditioner, we assume the unknown variables and the corresponding nonlinear equations are partitioned into two subsets denoted as $[(\mathcal{F}_b(\mathbf{x}_b,\mathbf{x}_g), \mathcal{F}_g(\mathbf{x}_b,\mathbf{x}_g)]^T$. Usually, the dimension of $\mathbf{x}_b$ is much smaller than the dimension of $\mathbf{x}_g$. The set of $\mathbf{x}_b$, referred to as the 'bad' component, represents the components with locally strong nonlinearity, such as shock, discontinuity, and singularity. The way to obtain the partition is often problem-dependent, and several partitioning strategies are available in the literature, such as the

field-by-field approach for multiphysics problems [48] and the pointwise approach for scalar problems [26,27]. Next, we introduce a globally preconditioned nonlinear function, defined implicitly as

$$\mathbf{y} = \mathcal{G}(\tilde{\mathcal{F}}; \mathbf{x}) = \begin{bmatrix} \mathcal{T}_b \\ \mathbf{x}_g \end{bmatrix}.$$

Here, the evaluation of $\mathcal{G}(\tilde{\mathcal{F}}; \mathbf{x})$ for a given vector $\mathbf{x} = [\mathbf{x}_b, \mathbf{x}_g]^T$ requires the numerical solution of $\mathbf{z_b} = \mathcal{T}_b$, from the following subspace problem,

$$\tilde{\mathcal{F}}(\mathbf{z}_b, \mathbf{z}_g) = \begin{bmatrix} \mathcal{F}_b(\mathcal{T}_b, \mathbf{z}_g) \\ \mathbf{z}_g - \mathbf{x}_g \end{bmatrix} = 0$$

by using some nonlinear iterative method. Note that $\mathbf{z}_g$ is simply a copy of $\mathbf{x}_g$. The nonlinearly preconditioned semismooth Newton (NPSN) algorithm can be described as in Algorithm 1.

Algorithm 1 Nonlinearly Preconditioned Semismooth Newton (NPSN) algorithm.

Given an initial guess $\mathbf{y}^{(0)} \in \mathbb{R}^n$.

1: **for** $k = 0, 1, 2, 3, \ldots$ until convergence **do**
2: Nonlinear preconditioning (NP) step:

- Evaluate $\mathbf{x}^{(k)} = \mathcal{G}(\tilde{\mathcal{F}}; \mathbf{y}^{(k)})$

3: Semismooth Newton (SN) step:

- Inexactly solve $\mathcal{F}(\mathbf{x}^{(k)}) + \mathcal{J}_k d^{(k)} = 0$.
- Update $\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \alpha d^{(k)}$, where $\alpha \in (0, 1]$.

4: **end for**

Remark 3.1: Since the generalized Jacobian matrix is large, sparse, and ill-conditioned, a Krylov subspace method, such as the general minimal residual method (GMRES) with accelerated domain decomposition-based preconditioning, e.g. overlapping Schwarz methods, is more favorable than direct methods for solving the generalized Jacobian systems in Step 3 of Algorithm 1 in the parallel computing environment [3].

Remark 3.2: While the numerical results of right nonlinear elimination preconditioned Newton methods show promise for various applications, the theoretical analysis regarding their local or global convergence analysis remains open. As presented by Nguyen et al. [38], the mathematical tool developed for ordinal ordinary differential equations could help establish the global convergence of nonlinear preconditioned Newton-type methods or offer a heuristic explanation for their success.

3.4. Some illustrative examples in $\mathbb{R}^2$

Let us consider some simple problems, which can be easily visualized as a solution procedure to find the minimum of the objective function in $\mathbf{x} = (x_1, x_2)$

$$f(\mathbf{x}) := 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad \text{(O)}$$

with or without the constraint

$$h(\mathbf{x}) := x_1 + x_2 + 0.5 = 0 \quad \text{(E)},$$

and

$$g(\mathbf{x}) := x_1 \geq 0 \quad \text{(I)}.$$

Three cases are tested.

Case (I) Unconstrained optimization, (O): An optimal point is located at $(1, 1)^T$.
Case (II) Equality-constrained optimization, (O)+(E): An optimal point is located approximately at $(-0.4707, -0.0293)^T$.
Case (III) Inequality-equality-constrained optimization, (O)+(E)+(I): An optimal point is located at $(0, -0.5)^T$.

For all cases, $(x_1, x_2)^T = (0, 0)^T$ (and $\lambda = 0$ if needed) is used as an initial guess vector. An inexact Newton, Lagrange-Newton, and SN are used to solve these problems in conjunction with some nonlinear preconditioners, which will be described below in detail. We also include the results obtained using these methods without nonlinear preconditioning for comparison. To solve the equality-inequality-constrained optimization problem by SN, we begin by defining the Lagrangian functional as

$$\mathcal{L}(\mathbf{x}) := f(\mathbf{x}) + \lambda h(\mathbf{x}) + \mu g(\mathbf{x}),$$

and the associated KKT system is obtained by differentiating $\mathcal{L}$,

$$\begin{cases} F_3 := \mathcal{L}_\lambda(\mathbf{x}, \lambda, \mu) = h(\mathbf{x}) = 0, \\ F_2 := \mathcal{L}_{x_2}(\mathbf{x}, \lambda, \mu) = f_{x_2}(\mathbf{x}) + \lambda h_{x_2}(\mathbf{x}) = 0, \\ \quad\ \mathcal{L}_{x_1}(\mathbf{x}, \lambda, \mu) = f_{x_1}(\mathbf{x}) + \lambda h_{x_1}(\mathbf{x}) - \mu = 0, \\ \quad\ x_1 \geq 0,\ \mu \geq 0,\ \mu x_1 = 0. \end{cases}$$

The last two equations of the above system form the so-called complementarity problem, which is equivalent to the following system by eliminating μ.

$$\begin{cases} x_1 = 0 & \& \quad F_1 := \mathcal{L}_{x_1}(\mathbf{x}, \lambda, \mu) = f_{x_1}(\mathbf{x}) + \lambda h_{x_1}(\mathbf{x}) \geq 0, \\ x_1 \in (0, \infty) & \& \quad \mathcal{L}_{x_1}(\mathbf{x}, \lambda, \mu) = f_{x_1}(\mathbf{x}) + \lambda h_{x_1}(\mathbf{x}) = 0. \end{cases}$$

Note that only one of these two equations holds at a time. It can be further simplified as

$$\mathcal{F}_1 = \varphi(x_1, F_1(\mathbf{x})) = 0 \Longleftrightarrow x_1 \geq 0, \quad F_1(\mathbf{x}) \geq 0, x_1 F_1(\mathbf{x}) = 0,$$

where $\varphi(a, b)$ is the Fischer-Burmeister NCP function defined as (15). The corresponding nonlinear system of equations can be defined as

$$\begin{aligned} \mathcal{F}(\mathbf{x}) &= \begin{pmatrix} \varphi(g(\mathbf{x}), F_1(\mathbf{x})) \\ F_2(\mathbf{x}) \\ F_3(\mathbf{x}) \end{pmatrix} \\ &= \begin{pmatrix} \varphi(x_1, -400x_1(x_2 - x_1^2) + 2(x_1 - 1) + \lambda) \\ 200(x_2 - x_1^2) + \lambda \\ x_1 + x_2 + 0.5 \end{pmatrix} = 0. \end{aligned} \tag{23}$$

Note that

$$F(\mathbf{x}) := \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} \mathcal{L}_{x_1} \\ \mathcal{L}_{x_2} \\ \mathcal{L}_{\lambda} \end{pmatrix} = \begin{pmatrix} -400x_1(x_2 - x_1^2) + 2(x_1 - 1) + \lambda \\ 200(x_2 - x_1^2) + \lambda \\ x_1 + x_2 + 0.5 \end{pmatrix}.$$

Its generalized Jacobian $\mathcal{J}$ is defined as $\mathcal{J} = D_a + D_b J$, where

$$J = \begin{pmatrix} 400(3x_1^2 - x_2) + 2 & -400x_1 & 1 \\ -400x_1 & 200 + 2\lambda & 1 \\ 1 & 1 & 0 \end{pmatrix},$$

and the diagonal matrices

$$\begin{aligned} D_a &= \operatorname{diag}(d_a, 0, 0) \\ D_b &= \operatorname{diag}(d_b, 1, 1). \end{aligned}$$

Here, the coefficients are defined as

$$\begin{cases} (d_a, d_b) = \left(1 - \dfrac{a}{\sqrt{a^2 + b^2}}, 1 - \dfrac{b}{\sqrt{a^2 + b^2}}\right) & \text{if } a^2 + b^2 \neq 0, \\ (d_a, d_b) = (0, 0) & \text{otherwise,} \end{cases}$$

with $a = x_1$ and $b = F_1(\mathbf{x})$.

Next, we discuss the design of a nonlinear preconditioner for SN for this optimization problem. Assume the term $x_2 - x_1^2$ in the first two equations is a troublemaker. Hence, we eliminate $x_2 - x_1^2$ in the first equation by using the second equation to obtain

$$\varphi(x_1, 2\lambda x_1 + 2(x_1 - 1) + \lambda) = 0.$$

We construct two different preconditioners depending on whether the inequality constraint is satisfied. When $x_1 \geq 0$, by using the property of the NCP function, we have $(2\lambda + 2)x_1 + \lambda - 2 = 0$. Solving this equation for x_1 gives $x_1 = \frac{2-\lambda}{2\lambda+2}$, and we obtain $x_2 = -\frac{\lambda}{200} + x_1^2 = -\frac{\lambda}{200} + (\frac{2-\lambda}{2\lambda+2})^2$ from the second equation of (23). Substituting x_1 and x_2 in terms of λ into the third equation of (23) to derive a cubic polynomial of λ as $h(\lambda) = -4\lambda^3 + 192\lambda^2 + 396\lambda + 2000 = 0$. Then we define a nonlinear preconditioner by updating λ, followed by x_1 and x_2.

(1) Update $\lambda^{(k+1)} = \mathcal{G}_1(x_1^{(k)}, x_2^{(k)}, \lambda^{(k)}) \equiv \lambda^{(k)} - \frac{h(\lambda^{(k+1)})}{h'(\lambda^{(k)})}$ by one-step Newton iteration, where $h'(\lambda) = -12\lambda^2 + 384\lambda + 396$.
(2) Update $x_1^{(k+1)} = \mathcal{G}_2(x_1^{(k)}, x_2^{(k)}, \lambda^{(k+1)}) \equiv \frac{2-\lambda^{(k+1)}}{2\lambda^{(k+1)}+2}$.
(3) Update $x_2^{(k+1)} = \mathcal{G}_3(x_1^{(k+1)}, x_2^{(k)}, \lambda^{(k+1)}) = -\frac{\lambda^{(k+1)}}{200} + x_1^{(k+1)}$.

On the other hand, when the inequality condition is not satisfied, i.e. $x_1 < 0$, we construct the preconditioner by trying to make the intermediate solution to be in the active set as soon as possible, that is

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{2} x_1^{(k)} \\ x_2^{(k+1)} &= -0.5 - x_1^{(k)} \\ \lambda^{(k+1)} &= -200(x_2^{(k)} - (x_1^{(k)})^2) \end{aligned}$$

Table 1. A summary of three nonlinear functions and the corresponding nonlinear preconditioners for each test case.

Case	$\mathcal{F}(x)$	$\mathcal{G}(x)$
(I)	$\mathcal{F}_1 \equiv -400x_1(x_2 - x_1^2) + 2(x_1 - 1)$ $\mathcal{F}_2 \equiv 200(x_2 - x_1^2)$	$x_1 = \mathcal{G}_1 \equiv x_1 + \frac{1}{2}(x_1 - 1)$ $x_2 = \mathcal{G}_2 \equiv x_1^2$
(II)	$\mathcal{F}_1 \equiv -400x_1(x_2 - x_1^2) + 2(x_1 - 1) + \lambda$ $\mathcal{F}_2 \equiv 200(x_2 - x_1^2) + \lambda$ $\mathcal{F}_3 \equiv x_1 + x_2 + 0.5$	$\lambda = \mathcal{G}_1 \equiv \lambda - \frac{h(\lambda)}{h'(\lambda)}$ $x_1 = \mathcal{G}_2 \equiv \frac{2 - \lambda}{2\lambda + 2}$ $x_2 = \mathcal{G}_3 \equiv -\frac{\lambda}{200} + x_1^2$ if $x_1 \geq 0$, same as Case (II) else
(III)	$\mathcal{F}_1 \equiv \varphi(x_1, -400x_1(x_2 - x_1^2) + 2(x_1 - 1) + \lambda)$ $\mathcal{F}_2 \equiv 200(x_2 - x_1^2) + \lambda$ $\mathcal{F}_3 \equiv x_1 + x_2 + 0.5$	$x_1 = \mathcal{G}_1 \equiv \frac{1}{2}x_1$ $x_2 = \mathcal{G}_2 \equiv -0.5 - x_1$ $\lambda = \mathcal{G}_3 \equiv -200(x_2 - x_1^2)$

Table 2. A summary of the total number of Newton iterations for three test cases. 'without npc': SN is used; 'with NPC': NPSN is used; 'NPC only': the nonlinear preconditioners designed for each problem are used as iterative methods.

Method	*Case(I)*	*Case(II)*	*Case(III)*
without NPC	62	63	130
with NPC	8	3	9
npc only	21	916	54

Table 1 lists three nonlinear functions and their nonlinear preconditioners for each test case. The following findings are observed from Table 2, which summarizes the number of iterations required for convergence.

(1) Although the three test cases look simple, they are not easy for Newton-type methods, and many iterations are required for convergence. In particular, the inequality constraint increases the level of difficulty. As a result, the SN method takes more than 100 iterations to find the optimal point.
(2) With the help of a nonlinear preconditioner, the number of Newton iterations required for convergence can be significantly reduced; see the first and second rows in Table 2. On the other hand, the nonlinear preconditioners defined in Table 1 as nonlinear iterative methods are less efficient, especially for Case II. However, they can serve as effective preconditioners for accelerating the convergence of Newton-type methods.

To gain more insight into the convergence behavior of Newton-type iterations in each test case, in Figure 1, we plot the trajectories of the intermediate solutions produced by the Newton method with and without the nonlinear preconditioning. The contour curves of the modified objective function, $\log(f(x) + 1)$, are superimposed, and its corresponding level set numbers are also displayed.

In the unconstrained optimization problem (left of Figure 1), the un-preconditioned Newton solution gradually converges to the optimal point along the valley bottom, requiring many iterations due to small damping. Surprisingly, Newton's directions are nearly parallel to the gradient, indicating reasons other than weak descent direction for small step lengths. As noted by Cai and Hwang [7], also see Figure 2, the merit function shows significant changes between predicted and current points, indicating its highly 'ill-conditioned' nature. Consequently, the quadratic model's minimal point for linesearch could be very lose to the current approximation, leading to small damping. In contrast, the preconditioned Newton method's search direction slightly differs from the original, accelerating convergence by bypassing unnecessary intermediate steps in the valley.

On the other hand, for constrained optimization, the initial objective function value exceeds the optimal one, requiring Newton to work hard to ascend toward the optimal point along the shortest

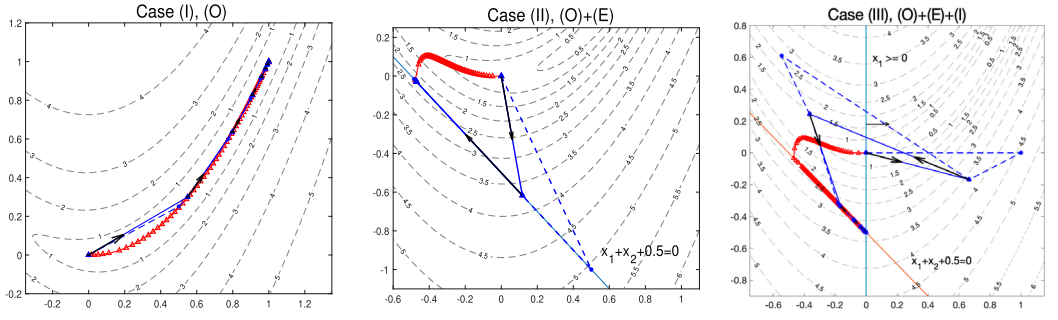


Figure 1. A comparison of the trajectory histories of Newton methods with and without nonlinear preconditioning. Case (I): unconstrained optimization problem (left); Case (II): equality-constrained optimization problem (middle); Case (III): equality-inequality optimization problem (right).

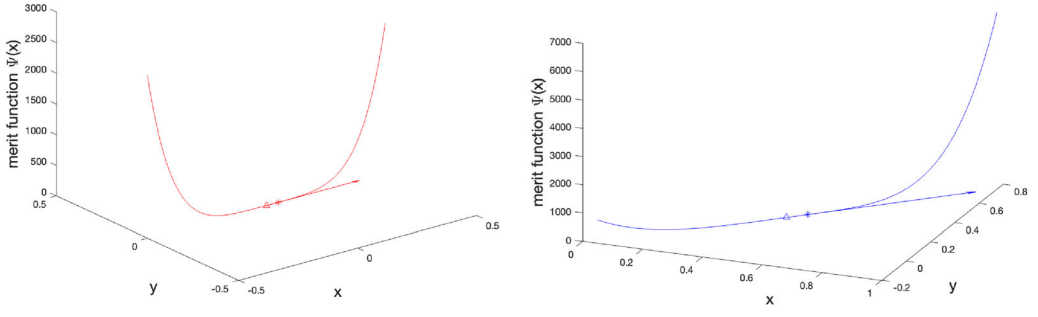


Figure 2. The merit function value along the Newton direction for INB without (top figure) and with (bottom figure) nonlinear preconditioning, respectively, at the first nonlinear iteration. The step length after backtracking in the un-preconditioned case is 0.0707, much smaller than the one in the preconditioned case (0.3050).

path between these two points. However, the problem is a minimization one. In this case, the Newton search direction is a weak descent direction in the (x_1, x_2, λ) space, differing from the unconstrained optimization case. When the additional inequality constraint is imposed, the convergence history of SN can be separated into two phases: the first phase is the same as the equality-constraint case. SN first tries to force the intermediate solution to satisfy the equality constraint, then moves it gradually along the line for the equality constraint to meet the inequality constraint in the second phase. The zig-zag dashed blue line (middle and right of Figure 1) shows the intermediate steps after the nonlinear preconditioning is applied and the global update is performed within each Newton iteration. The objective function in an optimization problem can be viewed as a measure of the potential energy of the system. When the difference in potential energy between the current and predicted steps is minor, convergence slows. Nonlinear preconditioning seeks a better initial guess, shifting the step to a higher potential energy position. This improved guess is globally updated with momentum, converting system potential energy to kinetic energy. This process accelerates convergence to the optimal solution. However, its specifics vary depending on the problem, possibly requiring further elaboration for complete understanding.

4. Numerical results and discussion

In this section, we report a series of numerical experiments to evaluate the performance of the proposed algorithm. The numerical experiments consist of an obstacle problem [32,45] and a high-pressure chemical vapor deposition (CVD) reactor problem [20,28,46,48]. The algorithm is

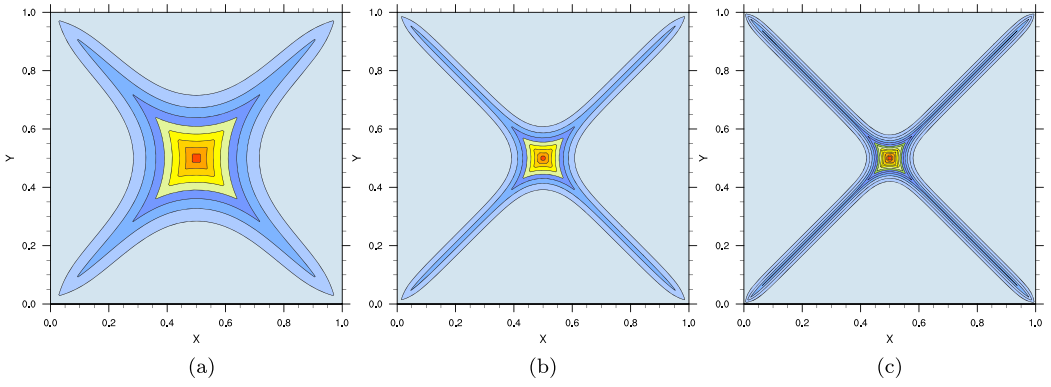


Figure 3. The active (multi-colored) and inactive (grey) sets with different C. (a) $C = 5$ (b) $C = 10$ (c) $C = 15$.

implemented using the Portable, Extensible Toolkit for Scientific computation (PETSc) library [3]. The numerical experiments are carried out on a parallel supercomputer with 512 processor cores.

4.1. An obstacle problem

In this experiment, we consider an obstacle problem [29,32,45] defined in the domain $\Omega = [0, 1] \times [0, 1]$. This problem is used to model the elastoplastic torsion of a cylindrical bar with a quadratic cross-section: find $u(x)$ such that

$$\begin{cases} -\Delta u(x) + C \geq 0, & x \in \Omega, \\ u(x) \geq -\mathrm{d}(x, \partial\Omega), & x \in \Omega, \\ (u(x) + \mathrm{d}(x, \partial\Omega))\,(-\Delta u(x) + C) = 0, & x \in \Omega, \\ u(x) = 0, & x \in \partial\Omega, \end{cases} \tag{24}$$

where the $\mathrm{d}(x, \partial\Omega)$-operator measures the distance from a point x to the domain boundary $\partial\Omega$. In this problem, the level curves of the solution consist of the region of 'inactive' points, where $-\Delta u(x) + C = 0$ is valid, and 'active' points, where the obstacle $u(x) + \mathrm{d}(x, \partial\Omega) = 0$ is achieved. The inactive region, whose size depends on parameter C, represents the plastic region, while the active points, where the second constraint with equality sign is valid, correspond to the elastic region. Figure 3 compares the active/inactive sets with the different values of C. In this figure, the inactive sets are the multi-colored regions, and the active sets are the grey regions. The active region increases as the value of C decreases.

Using the standard second-order central difference approximation for the term $\Delta u(x)$, the discretization of (24) on a uniform mesh leads to the complementarity problem in the form of (13). After reformulation, the resulting nonlinear algebraic system of Equation (14) is solved using SN or NPSN. We employ the standard cubic backtracking algorithm [13,15] with $\sigma = 10^{-4}$ in (6) to pick the step length for SN. While our focus is on the analysis of the SN method without or with the SN technique, we did not perform the sensitivity analysis for two parameters, α_1 and α_2, in the definition of the generalized Jacobian matrix. We chose the same parameters for both algorithms for a fair comparison. Although it is a naive but good choice, setting them to zero for the obstacle problem and the high-pressure CVD reactor problem can ensure numerical stability in the computation of the generalized Jacobian matrix, a crucial ingredient of our method to determine the search direction and update the solution iteratively.

For SN, the choice of an initial guess plays a crucial role in affecting the convergence and efficiency of the algorithm. Specifically, SN fails to converge when a naive zero vector is used as an initial guess. Even with a better initial guess, such as $u^0 = -\mathrm{d}(x, \partial\Omega)$, where x is a collection of corresponding

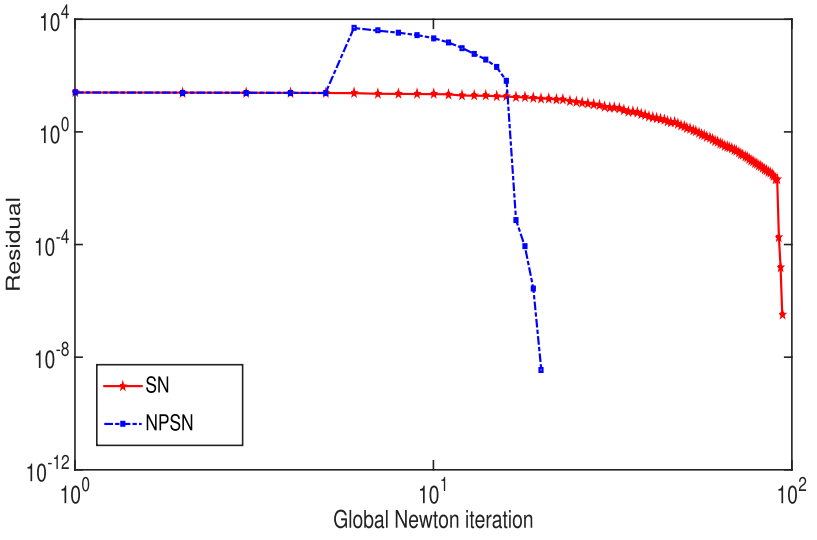


Figure 4. Nonlinear residual histories for the obstacle problem on a 128 × 128 mesh and $C = 15$.

grid points, where x represents a collection of corresponding grid points, as shown in Figure 4, SN still experiences a long stagnation period before recovering from quadratic convergence in the final iterations.

Within NPSN, SN is used for both global and local nonlinear solvers with an absolute (relative) tolerance of 10^{-6} (10^{-8}), and the Jacobian system is solved by the overlapping restricted additive Schwarz preconditioned GMRES method. For this example, we use the absolute value of the point-wise nonlinear residual and the nonlinear residual norm to judge whether the nonlinearity of the system is balanced and to identify the bad components needed for the construction of a nonlinear elimination preconditioner. The index set for the bad components, S_b, is defined as for all $i \in S$ such that

$$|\mathcal{F}_i(\mathbf{x})| > \rho \, \|\mathcal{F}(\mathbf{x})\|_\infty ,$$

where is a pre-chosen constant. The number of bad components corresponding to the dimension of the subspace problems depends on the value of ρ during the Newton iterations. After preliminary investigation, we found $\rho = 0.3$ to be an appropriate choice for all experiment runs to balance costs between global and subspace solutions to minimize overall computation time and ensure NPSN convergence. Generally, bad components gradually decrease as the semismooth Newton algorithm progresses. The percentage of bad components relative to the total number of points is reduced from 1 to a certain percentage controlled by the level curves of the solution to (24).

In NPSN, we can include mechanisms to turn the nonlinear preconditioning step on or off to avoid unnecessary overhead. From our numerical experience, we find that the small residual does imply a small error for some cases. As a result, we may identify the to-be-eliminated components inaccurately with the residual-based strategy when a zero initial guess is used. Therefore, we perform a few steps of classical SN, say N_{switch}, before turning on the nonlinear preconditioning step so that the large component-wise residuals can reveal the region of local high nonlinearity of the system. We set N_{switch} to be 5 for simplicity. Table 3 shows the number of Newton iterations and the computing time for the different values of C on a 128 × 128 mesh with SN and NPSN. The tests are carried out on a computer with four processors. As shown in the table, as C becomes smaller, the number of iterations and the computing time for SN increase. In contrast to SN, for NPSN, the number of nonlinear iterations is reduced with the nonlinear preconditioner. However, the number of iterations depends slightly on C.

Table 3. Performance of SN and NPSN for the obstacle problem with respect to the parameter C. In the table, 'Newton' denotes the number of inexact Newton iterations, and 'Time' denotes the total computing time in seconds.

	$C = 15$		$C = 10$		$C = 5$	
Method	Newton	*Time*	*Newton*	*Time*	*Newton*	*Time*
SN	94	2.5	103	2.8	116	3.2
NPSN	20	0.87	33	0.92	43	1.27

Figure 4 compares the histories of the nonlinear residual norms of NPSN and SN for $C = 15$ case on a 128×128 mesh. Since the nonlinear system is highly nonlinear, it is difficult for SN to progress after many Newton iterations, and the norm of nonlinear residual decreases gradually to around 10. On the other hand, the nonlinear preconditioning technique improves convergence by balancing the nonlinearities. As a result, NPSN converges well for the test case. It is worth mentioning that the performance of NPSN and SN for the other mesh sizes and values of C are similar to that for this case, and we do not include these plots for brevity.

4.2. A high pressure chemical vapor deposition reactor problem

We consider the boundary control of thermally convective flows and focus on minimizing the vorticity in incompressible fluid flow by controlling the temperature of the surrounding medium on the wall [1,5,6,19]. One benchmark problem of this type is the so-called high-pressure chemical vapor deposition (CVD) reactor problem that deposits layer by layer of a substance on a thin film [20,28,46,48], mathematically formulated as follows.

$$\min \mathcal{J} = \frac{1}{2}\int_{\Omega} |\omega|^2 \, \mathrm{d}\Omega + \frac{\gamma}{2}\int_{\{0,1\}\times(0,1)} |g|^2 \, \mathrm{d}\Gamma$$

subject to the constraints of stationary Navier–Stokes equations in the velocity-vorticity formulation on the domain $\Omega = (0,1) \times (0,1)$:

$$\begin{cases} -\Delta \mathbf{u} - \nabla \times \omega = 0, \\ -\Delta \omega + \nabla \cdot (\mathbf{u}\omega) - Gr \nabla_x T = 0, \\ -\Delta T + Pr \nabla \cdot \mathbf{u} = 0, \end{cases} \tag{25}$$

where Ω is the computational domain in $\mathbb{R}^2$, $\mathbf{u} = (u, v)$ is the velocity field, $\omega = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$ is the fluid vorticity, and T is the temperature, Gr is the Grashof number, and Pr is the Prandtl number. The corresponding boundary conditions are given as follows:

$$\begin{cases} \mathbf{u} = (0,0) \text{ and } T = 1, & \text{on } [0,1] \times \{0\}, \\ \mathbf{u} = (0,0) \text{ and } \dfrac{\partial T}{\partial n} = g - T, & \text{on } \{0,1\} \times (0,1), \\ \mathbf{u} = (0, -4(x - 1/3)(2/3 - x)) \text{ and } T = 0, & \text{on } (1/3, 2/3) \times \{1\}, \\ \mathbf{u} = (0, 2x(1/3 - x)) \text{ and } \dfrac{\partial T}{\partial n} = 0, & \text{on } [0, 1/3] \times \{1\}, \\ \mathbf{v} = (0, 2(x - 2/3)(1 - x)) \text{ and } \dfrac{\partial T}{\partial n} = 0, & \text{on } [2/3, 1] \times \{1\}, \end{cases}$$

where g is a temperature control to be computed, the regularization parameter γ equals 10^{-2}. In the study, we set $Re = 1$ and $Pr = 0.72$, considering different values of Gr. This work requires the interior temperature to be $0 \le T \le 1$. This problem can be mathematically formulated as a constrained optimization problem with inequality constraints, as introduced in Subsection 3.2.

We employ the discretize-then-optimize approach, where the objective functional and the PDE constraints are approximated numerically before applying an optimization method to the finite-dimensional nonlinear optimization problem; see [41,42,46] for more details. In particular, the

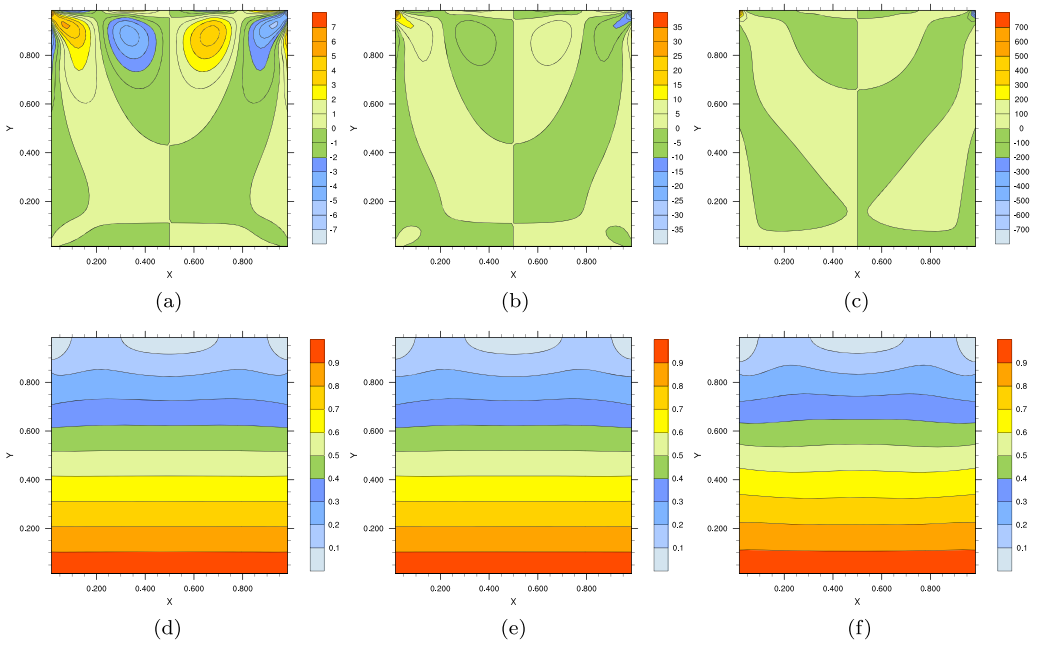


Figure 5. The magnitude of the vorticity (first row) and the temperature contours (second row) for different values of *Gr*. (a) $Gr = 5 \times 10^3$ (b) $Gr = 10^4$ (c) $Gr = 5 \times 10^4$ (d) $Gr = 5 \times 10^3$ (e) $Gr = 10^4$ (f) $Gr = 5 \times 10^4$.

corresponding nonlinear algebraic problem (22) is solved using either SN or NPSN with a zero initial vector. For the NE preconditioner, the subspace correction system is defined as

$$\tilde{\mathcal{F}}(\mathbf{x}) = \begin{pmatrix} \mathbf{x}_g - \mathbf{x}_g^{(k)} \\ \mathcal{F}_b(\mathbf{x}) \end{pmatrix} = 0, \tag{26}$$

where $\mathbf{x}_g$ is the subvector for all the components except for the variable T and its Lagrangian variable λ_T. These two field variables are selected with the convergence analysis as in [48]. An absolute (relative) tolerance of 10^{-10} (10^{-6}) is utilized for the Newton iteration, and the linear systems are solved by the Schwarz preconditioned GMRES method with relative and absolute tolerances of 10^{-10} and 10^{-6}, respectively. Figure 5 shows the magnitude of the vorticity and the temperature contours for different values of *Gr* with a fixed mesh, 64 × 64. It can be observed that the computed temperatures in the interior domain are in the range of 0 and 1, which is forced by the inequality constraint imposed.

4.2.1. Influence of nonlinear preconditioning

In the subsection, we conduct numerical experiments to investigate the effectiveness of SN with and without nonlinear preconditioning. The interpretation of the right nonlinear preconditioning step is to obtain a better initial guess for each Newton iteration, which can enhance the robustness and reduce the overall cost of the algorithm. Table 4 shows the nonlinear iteration counts and timing results obtained using the standard SN method and its nonlinearly preconditioned version. We keep the ratio of the mesh size and *Np* fixed and vary the mesh size from 64 × 64 to 256 × 256. The range of *Gr* from 10^3 to 10^5 is selected to test. The switch parameter, N_{switch} is set to be 3.

From this table, we find that for the small (mesh size 64 × 64) and medium-size (mesh size 128 × 128) cases, SN converges up to $Gr = 2 \times 10^4$, while for the finest mesh case, it only converges when *Gr* is around a few thousand. Moreover, for the convergent instances, the number of SN iterations is usually large, but it is not correlated to *Gr*, making the convergence behavior not predictable

Table 4. Performance of SN and NPSN with respect to different Grashof numbers and mesh sizes. $N_{switch} = 3$. $N_p = 4$ for a 64×64 mesh, $N_p = 16$ for a 128×128 mesh and $N_p = 64$ for a 256×256 mesh.

		64×64		128×128		256×256	
Method	Gr	Newton	Time	Newton	Time	Newton	Time
SN	10^3	35	82.8	98	148.7	92	307.2
	5×10^3	19	29.1	97	154.0	184	618.3
	10^4	24	32.4	85	137.5	*	*
	2×10^4	48	100.7	161	280.5	*	*
	5×10^4	*	*	*	*	*	*
	10^5	*	*	*	*	*	*
NPSN	10^3	8	29.1	12	54.5	13	100.8
	5×10^3	11	32.8	11	57.9	14	130.6
	10^4	11	33.1	11	63.9	14	131.2
	2×10^4	11	33.7	17	80.2	17	141.8
	5×10^4	12	33.9	14	76.3	17	165.6
	10^5	14	35.8	16	81.4	20	202.5

Note: In the table, 'Newton' denotes the number of inexact Newton iterations, and 'Time' is the total computing time in seconds. The symbol '$*$' denotes divergence of the Newton iteration caused by the failure of linesearch.

for different Gr. On the other hand, with the help of nonlinear preconditioning, NPSN converges for all test cases with a wide range of mesh sizes and Gr. Generally, in contrast to SN, the number of NPSN iterations is more predictable; it grows mildly as Gr increases, about 33% to 75% when we increase Gr by two orders of magnitude. Next, we consider the cases in which both SN and NPSN converge. The benefit of using a nonlinear preconditioning method in conjunction with SN is not apparent for smaller Gr, e.g. $Gr = 5 \times 10^3$ or 10^4, on a coarse mesh. Although the number of nonlinear iterations is reduced, the overhead due to the construction and application of nonlinear preconditioning contributes mainly to the total computing time. As a result, the time savings for NPSN are marginal or even slower than for SN. However, on finer meshes, the usefulness of nonlinear preconditioning is more profound. The reduction of the nonlinear iterations is significant, and about 3.0 and 4.7 speedups can be achieved for $Gr = 10^3$ and 5×10^3, respectively.

In addition, Figure 6 compares the nonlinear residual histories of SN and NPSN for the case of $Gr = 10^4$ on a 128×128 mesh. We can observe from the figure that the nonlinear residual norm of SN stagnates around 10^{-1} without progress after many iterations due to the bad quality of the initial guess. For NPSN, there is a nondecreasing residual norm at the third iteration after activating the nonlinear preconditioning step, and then it monotonically decreases toward the solution.

4.2.2. Parametric tuning

In this subsection, we focus on the parallel performance of the proposed solver. We study several performance-related parameters, including a threshold N_{switch} for the nonlinear solver, the overlapping size δ for the Schwarz preconditioner, and the strong scalability with different numbers of processors N_p. In NPSN, N_{switch} is used to determine when to turn on the nonlinear preconditioning step to balance the costs and accuracy of solving the global and subspace nonlinear problems. In Table 5, we show the performance of NPSN with different N_{switch}. Based on the results from this table, we find that the threshold N_{switch} around three is appropriate in terms of the total computing time and the robustness and stability of the solver.

In NPSN, an important feature is the selection of linear preconditioners for solving global and subspace Jacobian systems. Here, we use the overlapping restricted additive Schwarz preconditioned GMRES method to solve these linear systems, and this section focuses on the performance of Schwarz preconditioners. In the Schwarz preconditioner, we set a fixed overlapping size for the subspace Jacobian systems to be six and investigate the effect of the overlapping size, δ for the global Jacobian systems. As shown in Table 6, the number of GMRES iterations decreases as the overlap increases.

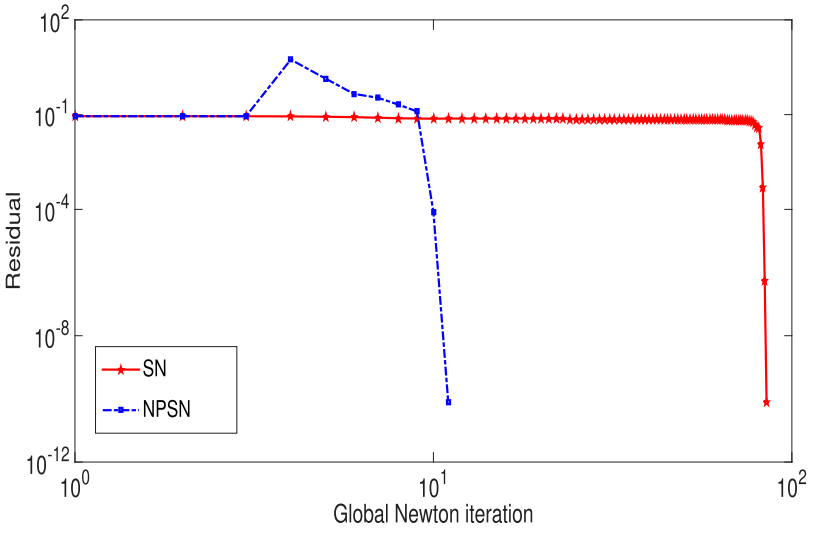


Figure 6. Nonlinear residual histories for the flow problem on a 128 × 128 mesh and $Gr = 10^4$.

Table 5. A comparison of the threshold N_{switch} with $Gr = 5 \times 10^4$. $N_p = 16$ for a 128 × 128 mesh and $N_p = 64$ for a 256 × 256 mesh.

		SN step			NP step		
Mesh	N_{switch}	Newton	GMRES	Time	Newton	GMRES	Time
128 × 128	1	14	71.8	53.4	81	17.2	73.5
	2	14	70.7	52.8	47	15.9	41.7
	3	15	69.2	52.4	34	15.6	29.9
	4	15	68.4	55.9	29	15.9	26.7
	5	17	68.9	57.4	27	15.8	23.9
256 × 256	1	10	191.4	68.3	100	29.7	125.7
	2	16	160.6	85.0	78	28.8	96.4
	3	17	157.2	87.8	64	27.8	77.8
	4	18	166.9	96.9	57	27.5	69.1
	5	19	173.4	105.4	51	27.6	62.1

Note: In the table, the 'SN step' denotes the number of NPSN in the semismooth Newton step, and the 'NP step' denotes the number of NPSN in the nonlinear preconditioning step. 'Time' is the computing time in seconds.

However, solving subdomain problems with a larger overlap requires more computing time. We conclude that a moderate overlap, e.g. $\delta = 6$ provides a good compromise between the computing time per iteration and the total number of linear iterations. Note that the Schwarz preconditioner is reduced to a block-Jacobi method for the case of $\delta = 0$, and we find that the number of GMRES iterations is two times or more when the block-Jacobi preconditioner is used compared to the Schwarz-type method as a preconditioner.

4.2.3. Parallel performance study

To study the parallel performance of NPSN, we consider the case of $Gr = 5 \times 10^3$ on a 512 × 512 grid. Two metrics are used for evaluating the parallel scalability: $S_p = T_1/T_2$, where T_1 and T_2 are the execution times obtained by running the parallel code with $N_{p,1}$ and $N_{p,2}$ processors ($N_{p,1} \leq N_{p,2}$), respectively. We also report the parallel efficiency of the method, defined as $E_f = (N_{p,1} \times T_1)/(N_{p,2} \times T_2)$. Table 7 summarizes the total computing time, parallel efficiency, and speed up with respect to the different number of processors. This table also reports the total number of nonlinear and nonlinear iterations for the NP and SN steps in Algorithm 1 and the computing time spent on these two components. From the table, we find that both SN and NP steps are nonlinear scalable and nearly linear

Table 6. Performance study for different overlapping size, δ, for $Gr = 5 \times 10^3$. 'Time' is the total computing time in seconds.

			SN step		
Mesh	N_p	δ	Newton	GMRES	Time
256 × 256	64	0	13	254.1	54.7
		2	13	116.3	44.1
		4	13	85.0	42.1
		6	13	69.6	43.6
		8	13	63.5	47.3
512 × 512	128	0	24	631.3	587.5
		2	24	236.9	258.6
		4	23	185.1	218.1
		6	23	154.7	210.9
		8	24	127.2	211.2

Table 7. Scalability analysis with N_p for $Gr = 5 \times 10^3$ on a 512 ×512 mesh.

	SN step			NP step			Overall performance		
N_p	Newton	GMRES	Time	Newton	GMRES	Time	Time	E_f(%)	S_p
64	23	108.0	401.3	128	37.2	840.2	1241.5	100	1.0
128	23	154.7	210.9	128	44.9	458.2	660.1	94.0	1.9
256	24	168.3	124.4	128	49.8	238.4	362.8	85.6	3.4
512	24	219.5	89.6	128	60.1	147.2	236.8	65.6	5.2

Note: 'SN step' denotes the number of NPSN in the semismooth Newton step, and 'NP step' indicates the number of NPSN in the nonlinearly preconditioned step. This table also includes the total computing time, parallel efficiency (E_f), and speed up (S_p).

scalable, i.e. the number of nonlinear iterations is independent of NP, and the number of linear iterations grows mildly as N_p increases. In addition, the overhead spent on the NP step is not neglectable, taking about 2/3 of the total computing time. Furthermore, NPSN achieves an efficiency of 65.6% or a good speedup of 5.2 with up to $N_p = 512$.

5. Concluding remarks

This paper developed a family of nonlinearly preconditioned semismooth Newton algorithms for large, sparse, nonsmooth nonlinear systems using physics-based field-split or domain decomposition-based nonlinear elimination preconditioning methods. Two key ingredients of the proposed method were subspace correction and global updates. Numerical results illustrate that the proposed nonlinear preconditioners effectively improved the performance of the global semismooth Newton iterations by balancing the distribution of nonlinearities in the system. Using obstacle and flow control problems with inequality constraints as numerical examples, we showed that the new approach is more robust and efficient than the standard semismooth method. Parallel scalability is also studied with a modest number of processors. Multilevel algorithm versions may need to be developed when the number of processors is large. In addition, we considered a series of illustrative 2D examples, which are easy to visualize. These examples help one understand how the equality and inequality constraint conditions, or both, for optimization problems affect the convergence behavior of the proposed methods. They also provide an idea of the role the preconditioner plays in the whole iterative process. On the other hand, we only considered the semismooth Newton algorithm with the classical linesearch technique based on the Armijo condition in the global update or subspace correction phases. Recent developments for some nonmonotone linesearch for unconstrained optimization problems with composite objective functions [23] in conjunction with a nonlinear preconditioning technique are worth investigating.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The first author was supported in part by the National Natural Science Foundation of China (No. 12371366 and 12131002), and the Basic and Advanced Research Project of CQ CSTC (No. cstc2021jcyj-msxmX0227) and the third author was supported in part by the National Science and Technology Council, Taiwan, NSTC 110-2115-M-008-005-MY2.

References

[1] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders, *Parallel algorithms for PDE-constrained optimization*, in *Parallel Processing for Scientific Computing*, M.A. Heroux, P. Raghavan, and H.D. Simon, eds., SIAM, Philadelphia, 2006.
[2] H.B. An, *On convergence of the additive Schwarz preconditioned inexact Newton method*, SIAM J. Numer. Anal. 43 (2005), pp. 1850–1871.
[3] S. Balay, S. Abhyankar, F.A. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, and H. Zhang, *PETSc Web page*, 2022. Available at http://www.mcs.anl.gov/petsc.
[4] S. Benson and T. Munson, *Flexible complementarity solvers for large-scale applications*, Optim. Method. Softw. 21 (2006), pp. 155–168.
[5] T.R. Bewley, *Flow control: new challenges for a new renaissance*, Prog. Aerosp. Sci. 37 (2001), pp. 21–58.
[6] G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver*, SIAM J. Sci. Comput. 27 (2005), pp. 687–713.
[7] S.R. Cai and F.N. Hwang, *A hybrid-line-and-curve search globalization technique for inexact Newton methods*, Appl. Numer. Math. 173 (2022), pp. 79–93.
[8] X.C. Cai and D.E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput. 24 (2002), pp. 183–200.
[9] X.C. Cai, D. Keyes, and L. Marcinkowski, *Non-linear additive Schwarz preconditioners and application in computational fluid dynamics*, Int. J. Numer. Methods Fluids 40 (2002), pp. 1463–1470.
[10] X.C. Cai, D.E. Keyes, and D.P. Young, *A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow*, in *Proceedings of the 13th International Conference on Domain Decomposition Methods*, Lyon, France, 2001. pp. 343–350.
[11] X.C. Cai and X. Li, *Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity*, SIAM J. Sci. Comput. 33 (2011), pp. 746–762.
[12] T. De Luca, F. Facchinei, and C. Kanzow, *A semismooth equation approach to the solution of nonlinear complementarity problems*, Math. Program. 75 (1996), pp. 407–439.
[13] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.
[14] V. Dolean, M.J. Gander, W. Kheriji, F. Kwok, and R. Massin, *Nonlinear preconditioning: How to use a nonlinear Schwarz to precondition Newton's method*, SIAM J. Sci. Comput. 38 (2016), pp. 3357–3380.
[15] S.C. Eisenstat and H.F. Walker, *Globally convergent inexact Newton method*, SIAM J. Optim. 4 (1994), pp. 393–422.
[16] F. Facchinei and J. Soares, *A new merit function for nonlinear complementarity problems and a related algorithm*, SIAM J. Optim. 7 (1997), pp. 225–247.
[17] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, 2008.
[18] S. Gong and X.C. Cai, *A nonlinear elimination preconditioned inexact Newton method for heterogeneous hyperelasticity*, SIAM J. Sci. Comput. 41 (2019), pp. S390–S408.
[19] M. Gunzburger, *Perspectives in Flow Control and Optimization*, SIAM, Philadelphia, 2003.
[20] M. Gunzburger and S. Manservisi, *The velocity tracking problem for Navier-Stokes flow with boundary control*, SIAM J. Control Optim. 39 (2000), pp. 594–634.
[21] P. Harker and J.S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications*, Math. Program. 48 (1990), pp. 161–220.
[22] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim. 13 (2003), pp. 865–888.
[23] Y. Huang and H. Liu, *A barzilai-borwein type method for minimizing composite functions*, Numer. Algor. 69 (2015), pp. 819–838.
[24] F.N. Hwang and X.C. Cai, *A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations*, J. Comput. Phys. 204 (2005), pp. 666–691.

[25] F.N. Hwang and X.C. Cai, *A class of parallel two-level nonlinear schwarz preconditioned inexact Newton algorithms*, Comput. Methods Appl. Mech. Eng. 196 (2007), pp. 1603–1611.
[26] F.N. Hwang, H.L. Lin, and X.C. Cai, *Two-level nonlinear elimination based preconditioners for inexact Newton methods with application in shocked duct flow calculation*, Electron. Trans. Numer. Anal. 37 (2010), pp. 239–251.
[27] F.N. Hwang, Y.C. Su, and X.C. Cai, *A parallel adaptive nonlinear elimination preconditioned inexact Newton method for transonic full potential equation*, Comput. Fluids 110 (2015), pp. 96–107.
[28] K. Ito and S.S. Ravindran, *Optimal control of thermally convected flows*, SIAM J. Sci. Comput. 19 (1998), pp. 1847–1869.
[29] T. Kärkkäinen, K. Kunisch, and P. Tarvainen, *Augmented Lagrangian active set methods for obstacle problems*, J. Optimi. Theory Appl. 119 (2003), pp. 499–533.
[30] C. Kanzow, *An unconstrained optimization technique for large-scale linearly constrained convex minimization problems*, Computing 53 (1994), pp. 101–117.
[31] A. Klawonn, M. Lanser, and O. Rheinbach, *Nonlinear FETI-DP and BDDC methods*, SIAM J. Sci. Comput. 36 (2014), pp. A737–A765.
[32] R. Kornhuber, *Monotone multigrid methods for elliptic variational inequalities I*, Numer. Math. 69 (1994), pp. 167–167.
[33] L. Liu, F.N. Hwang, L. Luo, X.C. Cai, and D. Keyes, *Nonlinear eliminiation preconditioned inexact Newton algorithm*, SIAM J. Sci. Comput. 44 (2022), pp. A1579–A1605.
[34] L. Liu and D. Keyes, *Convergence analysis for the multiplicative schwarz preconditioned inexact Newton algorithm*, SIAM J. Numer. Anal. 54 (2016), pp. 3145–3166.
[35] L. Liu, D. Keyes, and R. Krause, *A note on adaptive nonlinear preconditioning techniques*, SIAM J. Sci. Comput. 40 (2018), pp. A1171–A1186.
[36] L. Luo, W.S. Shiu, R. Chen, and X.C. Cai, *A nonlinear elimination preconditioned inexact Newton method for blood flow problems in human artery with stenosis*, J. Comput. Phys. 399 (2019), p. 108926.
[37] T. Munson, F. Facchinei, M.C. Ferris, A. Fischer, and C. Kanzow, *The semismooth algorithm for large scale complementarity problems*, INFORMS J. Comput. 13 (2001), pp. 294–311.
[38] N.C. Nguyen, P. Fernandez, R.M. Freund, and J. Peraire, *Accelerated residual methods for the iterative solution of systems of equations*, SIAM J. Sci. Comput. 40 (2018), pp. A3157–A3179.
[39] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer-Verlag, New York, 2006.
[40] J.S. Pang and L. Qi, *Nonsmooth equations: motivation and algorithms*, SIAM J. Optim. 3 (1993), pp. 443–465.
[41] E. Prudencio, R. Byrd, and X.C. Cai, *Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems*, SIAM J. Sci. Comput. 27 (2006), pp. 1305–1328.
[42] E. Prudencio and X.C. Cai, *Parallel multilevel restricted Schwarz preconditioners with pollution removing for PDE-constrained optimization*, SIAM J. Sci. Comput. 29 (2007), pp. 964–985.
[43] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res.18 (1993), pp. 227–244.
[44] M. Ulbrich, *Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems*, SIAM J. Optim. 11 (2001), pp. 889–917.
[45] H. Yang and X.C. Cai, *Parallel two-grid semismooth Newton-Krylov-Schwarz method for nonlinear complementarity problems*, J. Sci. Comput. 47 (2011), pp. 258–280.
[46] H. Yang and X.C. Cai, *Scalable parallel algorithms for boundary control of thermally convective flows*, in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, Shanghai, China, 2012, pp. 1387–1396.
[47] H. Yang and F.N. Hwang, *An adaptive nonlinear elimination preconditioned inexact Newton algorithm for highly local nonlinear multicomponent PDE systems*, Appl. Numer. Math. 133 (2018), pp. 100–115.
[48] H. Yang, F.N. Hwang, and X.C. Cai, *Nonlinear preconditioning techniques for full-space Lagrange-Newton solution of PDE-constrained optimization problems*, SIAM J. Sci. Comput. 5 (2016), pp. A2756–A2778.
[49] H. Yang, S. Sun, Y. Li, and C. Yang, *Parallel reservoir simulators for fully implicit complementarity formulation of multicomponent compressible flows*, Comput. Phys. Commun. 244 (2019), pp. 2–12.
[50] H. Yang, S. Sun, and C. Yang, *Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media*, J. Comput. Phys. 332 (2017), pp. 1–20.