PIN$^{\mathcal{L}}$: PRECONDITIONED INEXACT NEWTON WITH LEARNING CAPABILITY FOR NONLINEAR SYSTEM OF EQUATIONS*

LI LUO† AND XIAO-CHUAN CAI†

Abstract. Nonlinearly preconditioned inexact Newton methods have been applied successfully for some difficult nonlinear systems of algebraic equations arising from the discretization of partial differential equations. The preconditioning step involves identifying and balancing of the nonlinearities in the system. One of the challenging tasks when applying the methods is to accurately and efficiently identify the unbalanced nonlinearities. In this work, we propose an unsupervised learning strategy based on the classical principal component analysis that learns the bad behavior of a Newton solver in the nonlinear residual subspace of a training problem. A new initial guess is produced by the nonlinear preconditioner where a projected low dimensional Jacobian system corresponding to the slow subspace of the current residuals is solved for the Newton correction vector. Numerical experiments for high Reynolds number incompressible flow problems show that the proposed method is more robust and efficient compared with existing nonlinear solvers.

Key words. Inexact Newton, learning-based nonlinear preconditioning, principal component analysis, nonlinear system of algebraic equations, incompressible Navier-Stokes equations

AMS subject classifications. 49M15, 65Y05, 65M55, 76D05

1. Introduction.

Nonlinear preconditioning is a technique to enhance the robustness and efficiency of Newton-type methods for solving nonlinearly difficult system of algebraic equations arising from the discretization of nonlinear partial differential equations [7, 8, 22, 23]. The technique aims to balance the nonlinearities of the system by changing the function or the variable of the system without changing the solution, similar to linear preconditioning of linear systems [12]. Left nonlinear preconditioner changes the function of the original system and then solves the new system by a Jacobian-free Newton method [7, 10, 14, 18, 25, 30, 31, 32]. On the other hand, right nonlinear preconditioner changes the unknown variables of the original system [8, 17, 19, 24, 34, 35, 36, 43, 44, 45]. For most applications considered so far, the right preconditioner is easier to implement than the left version since it is less invasive to the standard software for inexact Newton methods. The key assumption needed in the design of a right preconditioner is that the components of the nonlinear system can be decomposed into two subspaces: a good subspace to be kept for further Newton iterations, and a bad subspace to be eliminated approximately using inner subspace Newton iterations. The method is often regarded as a nonlinear extension of Gaussian elimination, therefore, in the rest of the paper we refer to the method as Nonlinear Elimination (NE) preconditioning. The ability to identify the components that slow down the convergence is essential to the success of the NE preconditioner. Though the NE preconditioned inexact Newton method (PIN-NE) has been quite successful in many applications, there are challenges when using the method in practice:

1. The existing strategies identify the slow components by using knowledge of the physics or feedback from the intermediate solution, which generally require extra analysis of the numerical results. For example, for the transonic flow problems, the physics-based approach requires to detect the region where the shock occurs [19]. The field-based approach requires to determine which field variable is responsible for the dominant part of the residual norm

*Submitted to the editors DATE.

†Department of Mathematics, University of Macau, Macao SAR, China (liluo@um.edu.mo, xccai@um.edu.mo).

[36, 43, 44].

2. In some of the existing approaches such as the pointwise approach [17, 35, 45], the region-based approach [34], and the subdomain-based approach [8], the number of slow components to be eliminated depends sensitively on the preselected parameters, which has a significant impact on the effectiveness and efficiency of the preconditioner.
3. For domain-based approaches [8, 24, 34, 35, 45], new jumps may be produced in the residual across the interface between the good and bad regions or the subdomains, and this may lead to the relocation of unbalanced nonlinearities. Such interfacial jumps are often not easy to remove.

In this paper, we propose and study a novel nonlinear preconditioning method based on unsupervised learning to circumvent these obstacles.

Recent advances in machine learning and data analysis have shed light on devising new numerical methods with learning capability. With the explosive growth of available data and computing resources, a series of learning-based approaches emerged in the past decade for various scientific applications, i.e., image recognition [28], weather prediction [5], fluid mechanics [6], and particularly, the solve of general partial differential equations [20, 40]. The goal of this work is to develop a new paradigm in integrating learning capability into the class of preconditioned inexact Newton methods for nonlinear system of equations. We consider an unsupervised learning algorithm based on the classical principal component analysis (PCA), which is also known as the proper orthogonal decomposition (POD) method [11]. PCA was designed to find a low dimensional subspace of the given (high dimensional) data that keeps its most statistically descriptive factors, which has been successfully used in a variety of fields including data compression [33], computational fluid dynamics [26], structural mechanics [21], and reservoir simulation [37]. For the purpose of reduced order modeling, the algorithm has been applied to improve the convergence of linear solvers. In [9], the authors proposed a class of POD-augmented Krylov-subspace recycling methods. In [38], a reduced order model based preconditioner was introduced for the solution of transient diffusion equations. The preconditioners in both references [9, 38] are obtained by nesting appropriate POD projection into the classical conjugate gradient method. In [3], the authors combined POD with a two-stage constrained pressure residual solver for the solution of a two-phase reservoir model.

In this work, we associate the bad behavior of a Newton solver with the principal components of the nonlinear system, and apply PCA to find a reduced order approximation of the residuals with the projection operators learned from a training problem. Such an approximation is regarded as the low frequency components of the nonlinearity and is then reduced by a nonlinear preconditioning step. In the nonlinear preconditioner, a subspace Newton iteration with a projected low dimensional Jacobian system corresponding to the slow subspace of the residuals is introduced to obtain a new initial guess for the global Newton iteration. In contrast to common reduced order models, the training problem may differ from the original problem in size and complexity. Moreover, the proposed nonlinear preconditioner features a low computational cost since the projected Jacobian system generally has a very small size. We test the proposed method with two high Reynolds number incompressible flow problems including the lid-driven cavity flows and the backward-facing step flows. For such problems, the classical inexact Newton method often suffers from slow convergence or not converge at all, even with a good initial guess provided by some continuation techniques, such as parameter continuation [1] and mesh sequencing [27]. Numerical results show that the proposed method outperforms the classical

inexact Newton method and other preconditioned inexact Newton methods in terms of robustness and efficiency.

The paper is organized as follows. In Section 2, the proposed preconditioned inexact Newton method with learning capability is presented. The algorithm of PCA and the process of the proposed method are described in detail. In Section 3, numerical experiments for high Reynolds number incompressible flow problems are provided, including the validation with benchmark results, the study of robustness and efficiency of the algorithm, and the comparison with other nonlinear solvers. Some concluding remarks are given in Section 4.

2. Preconditioned inexact Newton methods with learning capability. Consider a nonlinear system of algebraic equations F: $R^n \to R^n$. We seek $X^* \in R^n$, such that

$$F(X^*) = 0, \tag{2.1}$$

starting from an initial guess $X^0 \in R^n$, where $F = (F_1, \ldots, F_n)^T$, $F_i = F_i(X)$, and $X = (X_1, \ldots, X_n)^T$. We first recall the inexact Newton algorithm with backtracking (IN) [42]. Assume X^k is the current approximate solution, a new X^{k+1} can be computed via

$$X^{k+1} = X^k + \lambda^k S^k, \tag{2.2}$$

where the inexact Newton direction S^k satisfies

$$\|F'(X^k)S^k + F(X^k)\| \le \eta^k \|F(X^k)\|. \tag{2.3}$$

Here, $\eta^k \in [0, 1)$ is a forcing term that determines how accurately the Jacobian system needs to be solved. The step length $\lambda^k \in [0, 1]$ is obtained from a standard backtracking line search technique [13]. It determines a step size along the inexact Newton direction S^k such that

$$f(X^k + \lambda^k S^k) \le f(X^k) + \alpha \lambda^k \nabla f(X^k)^T S^k, \tag{2.4}$$

where the merit function $f = \|F\|^2/2$, and the parameter α is used to assure that f is reduced sufficiently (herein $\alpha = 10^{-4}$). The nonlinear iteration is stopped if

$$\|F(X^k)\| \le \max\left\{\gamma_a, \gamma_r \|F(X^0)\|\right\}, \tag{2.5}$$

where γ_a and γ_r are prescribed absolute and relative tolerances, respectively.

We remark that λ^k is a critically important parameter in IN. IN converges slowly when the value of λ^k is too small. In practice, the value of λ^k is often determined by a small number of components in the system that contribute a large percentage of the nonlinear residual norm.

The idea of nonlinear preconditioning is to increase the value of λ^k by balancing the overall nonlinearities of the system so that a single search direction S^k benefits all components of the system. Inspired by the recent advances in unsupervised learning techniques, we present a novel nonlinear preconditioning method with learning capability in this paper.

For the classical IN, the residual vectors computed during the Newton iterations offer useful information that is currently not sufficiently utilized. For example, there are often dominant coherent structures in the residual profile obtained at different

Newton steps, which are associated with the slow components of F. Using the language of multigrid methods, such structures characterize the low frequency components of the residual space that are difficult to be removed effectively by using global Newton iterations. In this work, we propose a nonlinear preconditioning algorithm to smooth these dominant structures by learning their patterns from the residual data. In the rest of the paper, we refer to this method as PIN$^{\mathcal{L}}$: preconditioned inexact Newton method with learning capability.

2.1. Unsupervised learning based on principal component analysis. In this section, we consider the widely used PCA to characterize a low dimensional approximation to the residuals produced by inexact Newton iterations. PCA first centers the dataset by a mean subtraction, then represents the dataset with a new coordinate system determined by the principal components that are uncorrelated (orthogonal) to each other, but have maximal correlation.

Suppose a dataset of s residual vectors $\left\{F(X^k) \in \mathbb{R}^n,\ k = 0, \ldots, s-1\right\}$ is generated by the Newton iterations of a training problem, which can be assembled as a residual matrix

$$\mathrm{F} = \left[F(X^0), F(X^1), \ldots, F(X^{s-1})\right] \in \mathbb{R}^{n\times s}. \tag{2.6}$$

PCA is to find an orthonormal matrix $\mathrm{P} \in \mathbb{R}^{n\times d}$, where d is an integer much smaller than n such that $\left\{y^k = \mathrm{P}^T F(X^k) \in \mathbb{R}^d,\ k = 0, \ldots, s-1\right\}$ forms a reduced dimensional subspace that keeps important features of F and the variance of the projected vectors is maximized. We define the space

$$\mathrm{H}_{n\times d} = \left\{\mathrm{P} \mid \mathrm{P} \in \mathbb{R}^{n\times d},\ \mathrm{P}^T\mathrm{P} = \mathrm{I}_{d\times d}\right\},$$

where $\mathrm{I}_{d\times d}$ is a $d \times d$ identity matrix, and the variance

$$\mathcal{V}(\mathrm{P}) = \sum_{k=0}^{s-1} \left\| y^k - \frac{1}{s}\sum_{l=0}^{s-1} y^l \right\|^2 = \sum_{k=0}^{s-1} \left\| \mathrm{P}^T \left(F(X^k) - \frac{1}{s}\sum_{l=0}^{s-1} F(X^l) \right) \right\|^2, \tag{2.7}$$

then P is obtained by solving the optimization problem

$$\max_{\mathrm{P}\in\mathrm{H}_{n\times d}} \mathcal{V}(\mathrm{P}). \tag{2.8}$$

Let the mean of the residual vectors be $\bar{F} = \dfrac{1}{s}\displaystyle\sum_{l=0}^{s-1} F(X^l) \in \mathbb{R}^n$. We denote the centered residual vector as $\hat{F}^k = F(X^k) - \bar{F}$, and the centered residual matrix $\hat{\mathrm{F}} = \left[\hat{F}^0, \hat{F}^1, \ldots, \hat{F}^{s-1}\right]$. To obtain the residual subspace projector P, we perform the singular value decomposition (SVD) of $\hat{\mathrm{F}}$ as follows:

$$\hat{\mathrm{F}} = \hat{\mathrm{U}}_F \hat{\Sigma}_F \hat{\mathrm{V}}_F^T, \tag{2.9}$$

where $\hat{\mathrm{U}}_F$ is an $n \times n$ orthogonal matrix, $\hat{\Sigma}_F$ is an $n \times s$ diagonal matrix of singular values $\sigma_F^0, \sigma_F^1, \ldots, \sigma_F^{s-1}$ arranged in a decreasing order, and $\hat{\mathrm{V}}_F$ is an $s \times s$ orthogonal matrix. The solution to the optimization problem (2.8) is given as $\mathrm{P} = \hat{\mathrm{U}}_F^d$, consisting of the first d columns of $\hat{\mathrm{U}}_F$ that form a new coordinate system of F, which is regarded

as the slow subspace of the nonlinear residuals. Therefore, P can be used to construct a PCA-projection of $F(X)$, i.e.,

$$\mathcal{F}(X) = \mathrm{P}\mathrm{P}^T(F(X) - \bar{F}) + \bar{F}. \tag{2.10}$$

Using this approximation, we can define an approximate nonlinear system

$$\mathcal{F}(Y) = 0, \tag{2.11}$$

whose solution Y will play the main role in the preconditioning algorithm to be introduced later.

Corresponding to the residual matrix (2.6), we define the following approximate solution matrix

$$\mathrm{X} = \left[X^0, X^1, \ldots, X^{s-1}\right] \in \mathbb{R}^{n\times s}. \tag{2.12}$$

Similar to the residual subspace projector, we also introduce a solution subspace projector Q such that

$$\max_{\mathrm{Q}\in\mathrm{H}_{n\times d}} \mathcal{J}(\mathrm{Q}), \tag{2.13}$$

where

$$\mathcal{J}(\mathrm{Q}) = \sum_{k=0}^{s-1} \left\| \mathrm{Q}^T \left(X^k - \frac{1}{s}\sum_{l=0}^{s-1} X^l \right) \right\|^2. \tag{2.14}$$

Let the mean of the solution vectors be $\bar{X} = \dfrac{1}{s}\displaystyle\sum_{l=0}^{s-1} X^l \in \mathbb{R}^n$. We denote the centered solution vector as $\hat{X}^k = X^k - \bar{X}$, and the centered solution matrix $\hat{\mathrm{X}} = \left[\hat{X}^0, \hat{X}^1, \ldots, \hat{X}^{s-1}\right]$. To obtain Q, we perform the SVD

$$\hat{\mathrm{X}} = \hat{\mathrm{U}}_X \hat{\Sigma}_X \hat{\mathrm{V}}_X^T, \tag{2.15}$$

then Q can be formed by the first d columns of $\hat{\mathrm{U}}_X$. Note that for standard problems in, for example, image processing [29], a single PCA is performed, but here we need a pair of PCA projections. Because d is often a small value, the cost of calculating the SVDs is usually small.

2.2. The PIN$^{\mathcal{L}}$ algorithm. In this section, we describe the main steps of the proposed PIN$^{\mathcal{L}}$ algorithm:

Step 1. (The training step) Choose a suitable training problem, and run the classical IN to generate the training dataset. Compute P and Q by PCA based on the training dataset.

Step 2. (The nonlinear preconditioning step) Solve the approximated nonlinear system $\mathcal{F}(Y) = 0$ by a subspace Newton method to be discussed below with the initial guess $Y^0 = X^0$. The intermediate solution Y^* is accepted as an output when $\|\mathcal{F}(Y^*)\|$ is sufficiently small.

Step 3. (The global IN step) Solve the original nonlinear system $F(X) = 0$ by using IN with a corrected initial solution $X^{(0)} = Y^*$.

Steps 1 and 3 have been discussed in the previous section, here we focus on Step 2. The approximate nonlinear system $\mathcal{F}(Y) = 0$ is intended to capture the low frequency components of the original nonlinear system but its dimension is still n, and moreover its definition involves an $n \times n$ matrix $\mathrm{P}\mathrm{P}^T$ which is generally dense. It is often computationally intensive to solve the resulting algebraic system directly using a Newton-Krylov method. We hereby introduce a subspace Newton iteration with a projected low dimensional Jacobian system corresponding to the slow subspace of the residuals to correct the Newton solution. Start from the initial guess $Y^0 = X^0$, we proceed the following steps for $j = 0, 1, \ldots$

1. Compute the dimension-reduced PCA-projection

$$\mathcal{F}_p = \mathrm{P}^T \mathcal{F}(Y^j) = \mathrm{P}^T F(Y^j) \in \mathbb{R}^d. \tag{2.16}$$

2. Compute the low dimensional Newton correction $S_p^j \in \mathbb{R}^d$ by solving

$$\mathrm{J}_p S_p^j = -\mathcal{F}_p, \tag{2.17}$$

where

$$\mathrm{J}_p = \mathrm{P}^T F'(Y^j)\mathrm{Q} \tag{2.18}$$

is the projected Jacobian of size $d \times d$.

3. Compute the new approximate solution

$$Y^{j+1} = Y^j + \mathrm{Q} S_p^j. \tag{2.19}$$

The resulting Y^* is accepted as a corrected solution if the stopping condition

$$\|\mathcal{F}(Y^*)\| \le \gamma_r^s \|\mathcal{F}(Y^0)\|$$

is satisfied, where γ_r^s is a relative tolerance.

In the subspace Newton iteration, we use an exact Newton method without line search because the system is small. From an algebraic point of view, this process can be regarded as restricting the space $\mathbb{R}^n$ to a subspace of dimension d, finding the exact solution in $\mathbb{R}^d$ and prolongating the reduced solution back to $\mathbb{R}^n$. This is similar to a two-level multigrid method used to correct the low frequency components in the residual space. A detailed description of the overall method is presented in Algorithm 2.1.

Remark 2.1. In Step 2 of Algorithem 2.1, the solve of $\mathcal{F}(Y) = 0$ is considered as a nonlinear preconditioner of F, that is, $Y = G(X)$. Hence, the nonlinear system can be written as

$$F(G(X)) = 0 \tag{2.20}$$

and is called a right-preconditioned nonlinear system.

Remark 2.2. The dimension of the subspace Jacobian system is determined by the number of principal components d which is often small, thus the solve of the subspace Jacobian system is almost trivial. This is one key advantage of the proposed algorithm.

Remark 2.3. The learning-based preconditioner identifies the slow components by using an algebraic method for the residual space, by nature, it does not require extra

Algorithm 2.1 PIN$^{\mathcal{L}}$: Preconditioned Inexact Newton methods with Learning capability.

Step 1. The training step:

(1) Collect s nonlinear residual vectors $F(X^k)$ and s approximate solution vectors X^k from a training problem solved by IN, $k = 0, \ldots, s-1$.

(2) Form the centered residual matrix $\hat{\mathrm{F}} = \left[\hat{F}^0, \hat{F}^1, \ldots, \hat{F}^{s-1}\right]$ and the centered solution matrix $\hat{\mathrm{X}} = [\hat{X}^0, \hat{X}^1, \ldots, \hat{X}^{s-1}]$ by a mean subtraction.

(3) Compute the SVD for the centered residual matrix $\hat{\mathrm{F}} = \hat{\mathrm{U}}_F \hat{\Sigma}_F \hat{\mathrm{V}}_F^T$ and for the centered solution matrix $\hat{\mathrm{X}} = \hat{\mathrm{U}}_X \hat{\Sigma}_X \hat{\mathrm{V}}_X^T$.

(4) Form the residual subspace projector $\mathrm{P} = \hat{\mathrm{U}}_F^d$ and the solution subspace projector $\mathrm{Q} = \hat{\mathrm{U}}_X^d$.

Step 2. The nonlinear preconditioning step:

Start from the initial guess $Y^0 = X^0$, for $j = 0, 1, \ldots$

(1) Compute the approximated residual vector $\mathcal{F}(Y^j) = \mathrm{P}\mathrm{P}^T(F(Y^j) - \bar{F}) + \bar{F}$.

(2) If the stopping condition $\|\mathcal{F}(Y^j)\| \le \gamma_r^s \|\mathcal{F}(Y^0)\|$ is satisfied, set $Y^* = Y^j$, go to **Step 3**.

(3) Project $\mathcal{F}(Y^j)$ to a dimension-reduced vector $\mathcal{F}_p = \mathrm{P}^T \mathcal{F}(Y^j)$.

(4) Compute the projected Jacobian $\mathrm{J}_p = \mathrm{P}^T F'(Y^j)\mathrm{Q}$.

(5) Exactly solve $\mathrm{J}_p S_p^j = -\mathcal{F}_p$.

(6) Update $Y^{j+1} = Y^j + \mathrm{Q}S_p^j$.

Step 3. The global IN step:

Start from the initial guess $X^{(0)} = Y^*$, for $i = 0, 1, \ldots$

(1) Form the nonlinear residual $F(X^{(i)})$.

(2) If the global stopping condition $\|F(X^{(i)})\| \le \max\left\{\gamma_a, \gamma_r \|F(X^0)\|\right\}$ is satisfied, set $X^* = X^{(i)}$, stop.

(3) Form the Jacobian $\mathrm{J} = F'(X^{(i)})$.

(4) Inexactly solve $\mathrm{J}S^{(i)} = -F(X^{(i)})$.

(5) Compute $\lambda^{(i)}$ using the cubic backtracking line search.

(6) Update $X^{(i+1)} = X^{(i)} + \lambda^{(i)} S^{(i)}$.

analysis of the physics behind the partial differential equations. For the incompressible flow problems to be studied in Section 3, we will not separate the field variables or apply any *prior* knowledge of the solution or the intermediate solutions when performing the training step and the nonlinear preconditioning step.

Remark 2.4. Compared to the domain-based NE preconditioners [8, 24, 34, 35, 45], the proposed method does not partition the domain into different parts and treat them differently, thus avoiding the potential interfacial jumps.

Remark 2.5. In contrast to the adaptive nonlinear elimination preconditioner [17, 19, 32, 34, 35, 45], the learning-based nonlinear preconditioner is applied only once before the global Newton iteration, saving considerable compute time.

2.3. Other training methods. Since the training step of Algorithm 2.1 is for the construction of a preconditioner which does not need to be very precise, in this section, we propose several possible approximations of the training step without going into details:

- Training with a different problem. In the previous section we assume the training datasets (2.6) and (2.12) are from the original problem (2.1). In practical applications, the training problem is not necessary to be identical to the original problem. Similar to the idea of transfer learning, one may choose a training problem with certain parameters so that it is easier to solve than the original problem.
- Training on a different mesh. The algorithm introduced in the previous section is for nonlinear algebraic systems without requiring any mesh information. For problems defined on a mesh, the robustness of the nonlinear solver often degrades when the mesh is fine because more delicate physics are resolved, such as the small eddies of a driven cavity flow considered in the numerical experiments of this paper. When applying PIN$^{\mathcal{L}}$ directly on a fine mesh, the computational cost of the training step and the preconditioning step could be high. In order to reduce the computational cost, one possible strategy is to move the training step and the preconditioning step to a coarser mesh, and interpolate the solution to the fine mesh.
- Training data generated by a different method. Besides the classical IN, a variety of nonlinear solvers can be used to generate the training dataset, such as PIN-NE and other nonlinear preconditioned Newton methods. In particular, one can use PIN$^{\mathcal{L}}$ to generate a new dataset for further training and preconditioning by another PIN$^{\mathcal{L}}$ applied to even more difficult problems. The idea of retraining is similar to the continuation approaches utilizing results of prior problems [1, 27], but applied in a learning procedure. We will show in numerical tests that the proposed method is more powerful than the continuation approaches for solving highly nonlinear problems.

3. Numerical experiments. To evaluate the performance of the proposed algorithm, we consider two steady-state incompressible flow problems with high Reynolds numbers: the lid-driven cavity flows and the backward-facing step flows. Let $\Omega = (a,b)\times(c,d)$ be a bounded domain in $\mathbb{R}^2$. These flow problems can be modeled by the Navier-Stokes equations in the velocity-vorticity formulation:

$$\begin{cases} -\Delta u - \dfrac{\partial \omega}{\partial y} = 0, & \text{in } \Omega, \\ -\Delta v + \dfrac{\partial \omega}{\partial x} = 0, & \text{in } \Omega, \\ -\dfrac{1}{Re}\Delta\omega + u\dfrac{\partial \omega}{\partial x} + v\dfrac{\partial \omega}{\partial y} = 0, & \text{in } \Omega, \end{cases} \tag{3.1}$$

where u and v are the velocity fields in the x- and y-directions, respectively, and

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{3.2}$$

is the vorticity normal to the xy-plane. The Reynolds number Re quantifies the relative importance of inertial forces to viscous forces. Suitable boundary conditions are needed to close the system, which will be given later in the two problems respectively.

A standard central second-order finite difference scheme is used for the discretization of both the Laplacian operators and the first order partial derivatives in (3.1). Let Ω be covered by a $M\times N$ mesh, then each point $p_{ij} = (x_i, y_j)$ is located at the position $x_i = a+(i-1)h_x$ with $i = 1,\ldots,M$ and $y_j = c+(j-1)h_y$ with $j = 1,\ldots,N$, $h_x = (b-a)/(M-1)$, and $h_y = (d-c)/(N-1)$. In this work, we consider the point-block ordering to build up the large sparse nonlinear system of algebraic equations

(2.1), in which the unknown variables u_{ij}, v_{ij}, ω_{ij} associated with a mesh point p_{ij} are always together in a 3×3 block, i.e.,

$$X = (u_{11}, v_{11}, \omega_{11}, u_{21}, v_{21}, \omega_{21}, \ldots, u_{MN}, v_{MN}, \omega_{MN})^T,$$

and the corresponding functions are in the order of

$$F = (F^u_{11}, F^v_{11}, F^\omega_{11}, F^u_{21}, F^v_{21}, F^\omega_{21}, \ldots, F^u_{MN}, F^v_{MN}, F^\omega_{MN})^T,$$

where F^u_{ij}, F^v_{ij}, F^ω_{ij} are the components of F corresponding to the variables u, v, ω, respectively.

The numerical experiments are carried out on a computer with an Intel Xeon 6248 2.50GHz CPU. A zero vector is used as the initial guess, i.e., $X^0 = \mathbf{0}$. GMRES [41] is used for solving the Jacobian systems both in the global and subspace Newton iterations where the Jacobian matrices are computed analytically. The nonlinear solver is implemented using PETSc [4] and the SVD is calculated using the LAPACK dgesvd routine [2]. We use the following parameters in our solvers if they are not specifically stated. The restart value of GMRES is fixed at 50. A point-block ILU factorization with 3 fill-in levels is used for preconditioning the GMRES solver. The relative and absolute tolerance of the global nonlinear solver are $\gamma_r = 10^{-12}$ and $\gamma_a = 10^{-8}$, respectively. To enhance the robustness of inexact Newton, the forcing term $\eta^{(i)}$ is computed based on norms that are by-products of the iteration. For $i = 1, 2, \ldots$, we choose

$$\eta^{(i)} = \begin{cases} \eta_0, & \|F(X^{(i)})\| \geq \beta, \\ \frac{\left| \|F(X^{(i)})\| - \|F'(X^{(i-1)})S^{(i-1)} + F(X^{(i-1)}))\| \right|}{\|F(X^{(i-1)})\|}, & \|F(X^{(i)})\| < \beta, \end{cases} \tag{3.3}$$

where $\eta_0 \in [0, 1)$ and β are given constants. By default we use $\beta = \infty$ which corresponds to the Eisenstat-Walker method [15].

In the rest of this paper, "NI_g" denotes the number of global Newton iterations; "LI_g" denotes the averaged number of GMRES iterations per global Newton iteration; "NI_s" refers to the averaged number of subspace Newton iterations in the nonlinear preconditioning step; "LI_s" is the averaged number of GMRES iterations per subspace Newton; "T_{total}(s)" is the total compute time in seconds for the overall algorithm; "T_{precon}(s)" is the compute time in seconds for the nonlinear preconditioning step; "T_{train}(s)" is the compute time in seconds for PCA in the training step, in which the time needed for solving with IN to collect the datasets for PCA is not included.

3.1. The lid-driven cavity flow problem. In this section, we consider flows confined in the unit domain $\Omega = (0, 1) \times (0, 1)$, as depicted in Figure 1. The top boundary Γ_{lid} represents a lid moving with velocity $u = 1$ in the positive x-direction. On all walls we impose a no-slip and no-penetration boundary condition, specifically,

$$\begin{cases} u = 1, & \text{on} \quad \Gamma_{lid}, \\ u = 0, & \text{on} \quad \partial\Omega / \Gamma_{lid}, \\ v = 0, & \text{on} \quad \partial\Omega, \\ \omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, & \text{on} \quad \partial\Omega. \end{cases} \tag{3.4}$$

The boundary condition for the vorticity is discretized with a second-order approximation using mesh points adjacent to the boundary [39].

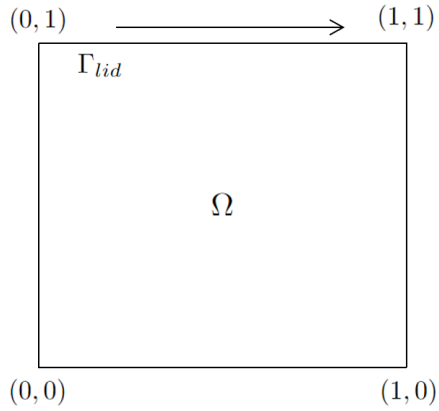
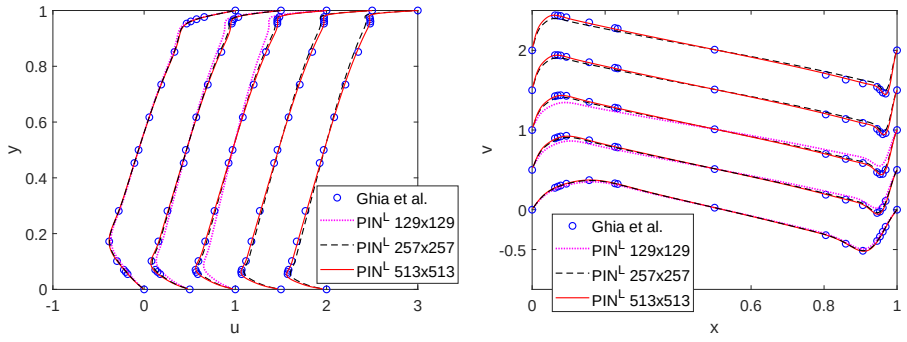


FIG. 1. *The computational domain for the lid-driven cavity flow problem.*

3.1.1. Validation of the proposed numerical method. We first validate the finite difference discretization and the proposed algorithm by comparing the velocity profiles of the cavity flow with benchmark results. A sequence of refined meshes ranging from 129×129 to 513×513 are used for the tests. Figure 2 shows the two velocity components u and v along the vertical and horizontal centerlines of the cavity for cases $Re = 10^3$, 3.2×10^3, 5×10^3, 7.5×10^3, and 10^4. The computed velocity profiles converge as the mesh is refined, and show good agreement with the published benchmark solutions in [16].

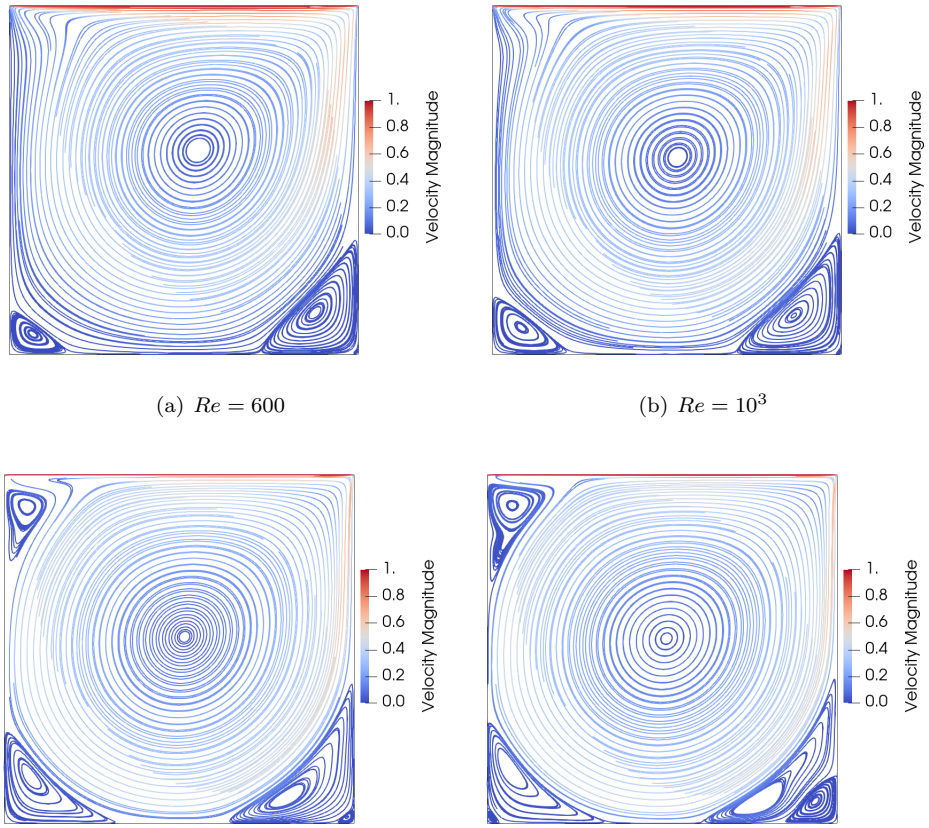


(a) $u(1/2, y)$ (b) $v(x, 1/2)$

FIG. 2. *Velocity profiles of the cavity flow at different Reynolds numbers. Note that the profiles are shifted for visual comparison. (a) u, from left to right: $Re = 10^3$, 3.2×10^3, 5×10^3, 7.5×10^3, and 10^4. (b) v, from bottom to top: $Re = 10^3$, 3.2×10^3, 5×10^3, 7.5×10^3, and 10^4.*

Figure 3 and Figure 4 show the streamlines and vorticity contours for the cavity flow with $Re = 600$, 10^3, 5×10^3, and 10^4, respectively. The mesh size is 513×513. As Re increases, a sequence of eddies with diminishing size are observed at the corners of the cavity. The patterns of streamlines and the vorticity contours match well with the results in the earlier studies [16, 18, 45].

3.1.2. Comparison of IN and PIN$^{\mathcal{L}}$. In this section, we study how PIN$^{\mathcal{L}}$ improves the convergence of the classical IN. Figure 5 displays the history of nonlinear



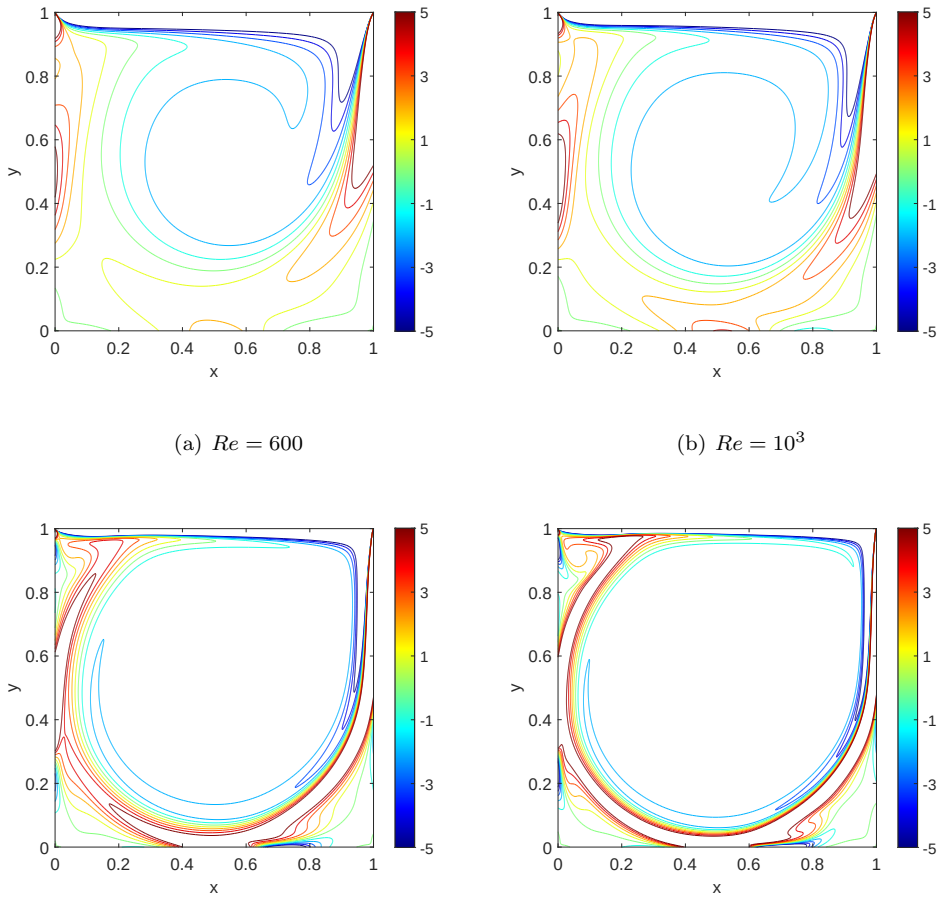
(a) $Re = 600$ (b) $Re = 10^3$

(c) $Re = 5 \times 10^3$ (d) $Re = 10^4$

FIG. 3. *Streamlines for the cavity flow with different Reynolds numbers. The mesh size is* 513×513.

residuals obtained using IN and PIN$^{\mathcal{L}}$ for the cavity flow with $Re = 600$ on a 257×257 mesh. For the classical IN, it is observed that the residual norm stagnates around 10^{-2} and the method requires 20 Newton steps to converge. We collect the resulting residuals to form the residual matrix F and learn the slow subspace by PCA. The singular values of $\hat{\mathrm{F}}$ and $\hat{\mathrm{X}}$ are plotted in Figure 6. Observed from Figure 6, $d = 5$ is a suitable choice for PCA to capture the principal components of the problem. The corresponding singular vectors that characterize the dominant patterns of nonlinearities are shown in Figure 7. In PIN$^{\mathcal{L}}$, the relative tolerance for the subspace Newton is set to be $\gamma_r^s = 10^{-3}$, and the parameters for the forcing term are given as $(\eta_0, \beta) = (0.1, 10^{-3})$. The numbers of iterations and compute time obtained using IN and PIN$^{\mathcal{L}}$ for this test are presented in Table 1. It can be seen in Figure 5 that with only 3 subspace Newton steps the residual norm reaches $O(10^{-3})$, providing a better initial guess for the global Newton iteration. Then, the global Newton converges quickly without any stagnation.

To see how the proposed preconditioner smoothes out the nonlinearities of the system, we show in Figure 8 the residual of components u and ω at different subspace



(a) $Re = 600$ (b) $Re = 10^3$

(c) $Re = 5 \times 10^3$ (d) $Re = 10^4$

FIG. 4. *Vorticity contours for the cavity flow with different Reynolds numbers. The mesh size is* 513×513.

Newton steps ($j = 0, 1, 2$). Note that the nonlinear function $\mathcal{F} = \mathrm{P}\mathrm{P}^T(F - \bar{F}) + \bar{F}$ is an approximation of F that characterizes its low frequency components. We can see from the figure that $\mathcal{F}$ captures the main features of F very well though a small value of d is used ($d = 5$), while the difference $F - \mathcal{F}$ shows the high frequency components. It is also seen that the subspace Newton effectively reduces the magnitude of the residuals, which leads to the fast convergence of PIN$^{\mathcal{L}}$.

As Re increases, the nonlinear system becomes harder to solve. On a 257×257 mesh, the classical IN fails to converge when Re is greater than 700, resulting in a series of residuals that can hardly be reduced. With such a dataset, PCA is not able to identify the slow subspace effectively. In this work, we perform the training step on the dataset obtained from a low Reynolds number problem, i.e., $Re = 600$, and use the resulting subspace projectors for preconditioning the nonlinear solver for a high

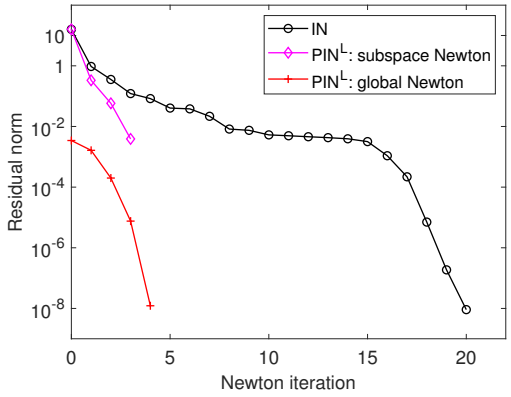
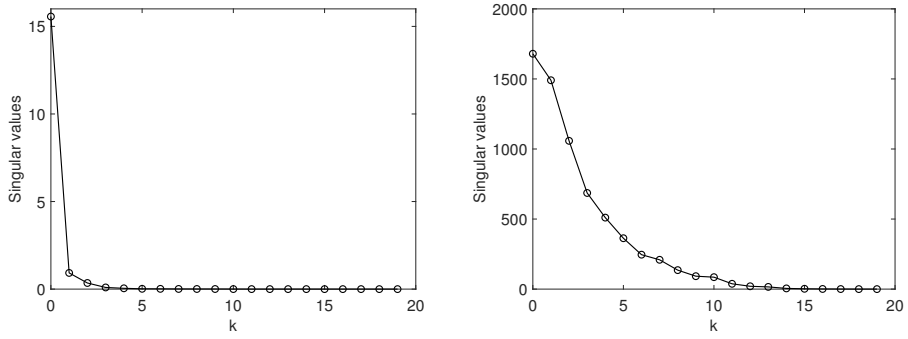


FIG. 5. *Nonlinear residual history obtained using IN and PIN$^{\mathcal{L}}$ for the cavity flow with $Re = 600$. The mesh size is 257×257. $d = 5$, $\gamma_r^s = 10^{-3}$.*

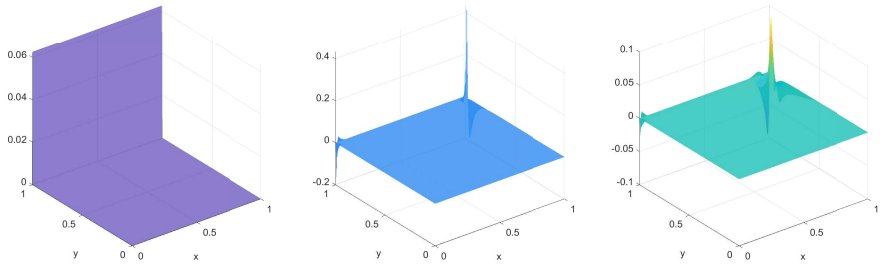


(a) σ_F^k (b) σ_X^k

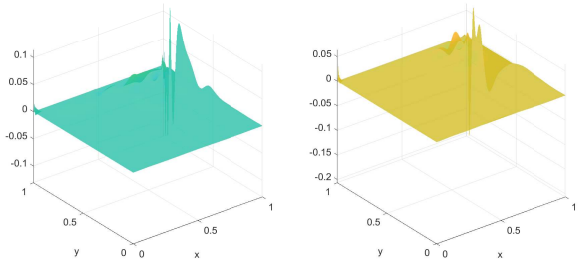
FIG. 6. *Singular values obtained using PCA for the residual dataset (a) and the solution dataset (b).*

Reynolds number problem, i.e., $Re \geq 10^3$. Figure 9(a) shows the nonlinear residual history obtained using IN and PIN$^{\mathcal{L}}$ for the cavity flow problem with different Re. For PIN$^{\mathcal{L}}$, we choose $d = 10$ and $\gamma_r^s = 10^{-4}$. For comparison, the results obtained using the Reynolds number continuation approach [1] are also presented in which the solution for case $Re = 600$ is used as the initial guess for cases with a larger Re. The continuation approach converges when $Re \leq 5 \times 10^3$ but fails for cases $Re \geq 7.5 \times 10^3$. In contrast, PIN$^{\mathcal{L}}$ converges well for all cases with $Re = 10^3 \sim 10^4$. Figure 9(b) shows the step length $\lambda^{(i)}$ with respect to the global Newton step for case $Re = 10^4$. PIN$^{\mathcal{L}}$ results in $\lambda^{(i)} = 1$ for almost every Newton step. The ability to restore the full step length along the Newton direction implies fast convergence of the Newton iteration.

A detailed comparison for the numbers of iterations and the total compute times between the two methods are summarized in Table 2. When $Re > 10^3$, PIN$^{\mathcal{L}}$ performs better than the Re continuation approach in terms of the numbers of global iterations and the total compute time. This shows that the proposed preconditioning technique is superior to the continuation approach provided with the same solution of the training problem. We also note that the compute time for the training step



(a) Component u of P$(:,1)$ (b) Component ω of P$(:,2)$ (c) Component ω of P$(:,3)$



(d) Component ω of P$(:,4)$ (e) Component ω of P$(:,5)$

FIG. 7. *Surface plot of the first five singular vectors of $\hat{F}$ (columns of the residual subspace projector P). $\hat{F}$ is obtained by using IN for the cavity flow with $Re = 600$ on a 257×257 mesh.*

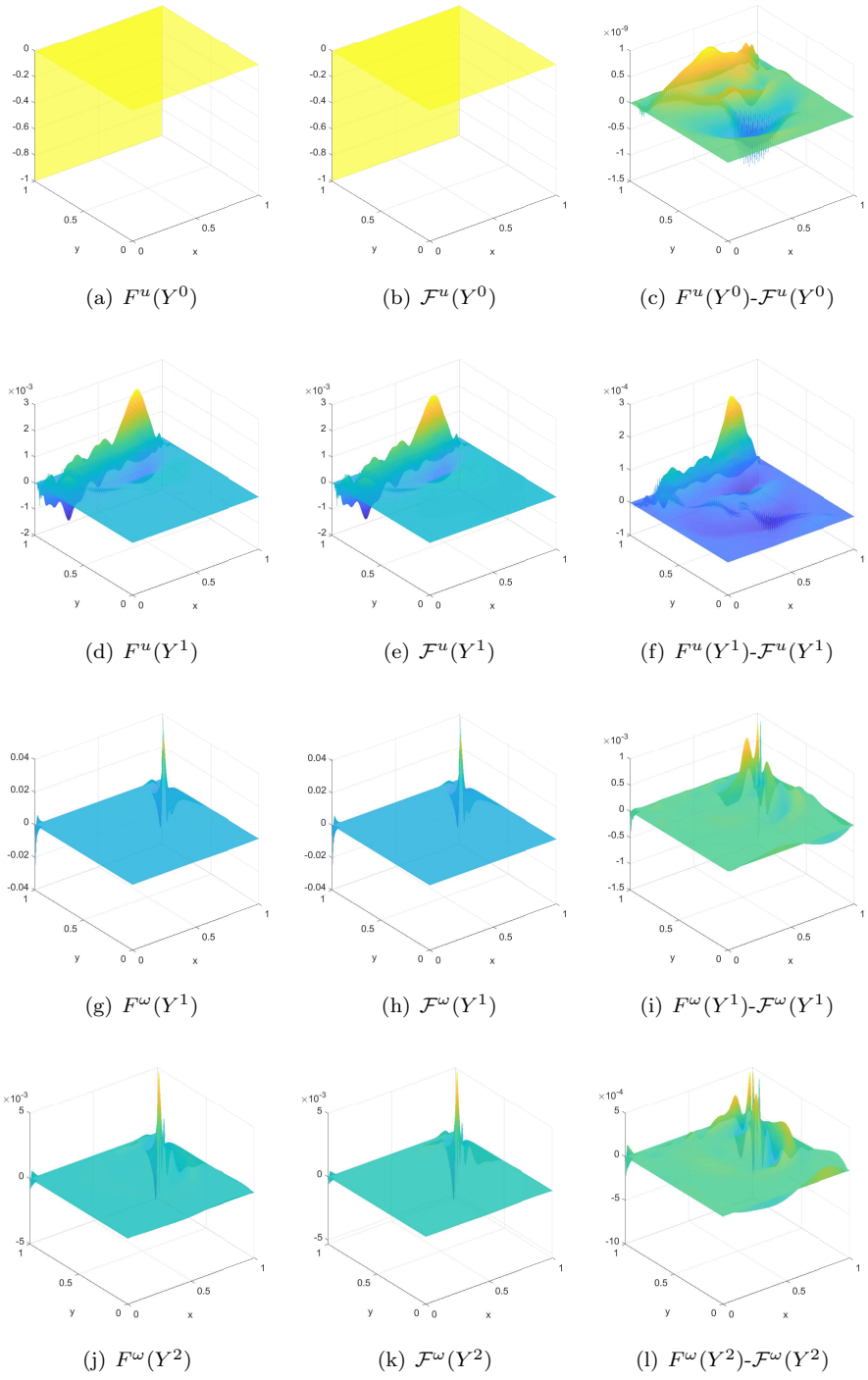
TABLE 1

The results obtained using IN and PIN$^{\mathcal{L}}$ for the cavity flow with $Re = 600$. The mesh size is 257×257, $d = 5$, and $\gamma_r^s = 10^{-3}$. "NI" denotes the number of Newton iterations, "LI" denotes the averaged number of GMRES iterations per Newton iteration, "T(s)" denotes the compute time in seconds.

	IN	PIN$^{\mathcal{L}}$		
		Training	Subspace Newton	Global IN
NI	20		3	4
LI	23.6		1	36.8
T(s)	45.7	0.7	1.2	19.3

and the preconditioning step take a small percentage of the total compute time of PIN$^{\mathcal{L}}$. On one hand, since the dataset F and X consist of a small number of residual and solution vectors obtained from a low Re problem, the application of PCA can be done efficiently. On the other hand, because the projected Jacobian system in the subspace Newton iteration has only d dimensions, one iteration is often sufficient for the linear solve.

3.1.3. The impact of preselected parameters and datasets. To understand the impact of the parameters on the performance of PIN$^{\mathcal{L}}$, we test the case $Re = 10^3$ using different values of d and γ_r^s. The mesh size is 257×257. The dataset collected for PCA is obtained by using IN for case $Re = 600$. The resulting numbers of Newton iterations and the compute times are shown in Table 3. The relative tolerance γ_r^s is used to determine how accurately the subspace nonlinear problem is to be solved. We find from the table that the method is robust with respect to γ_r^s



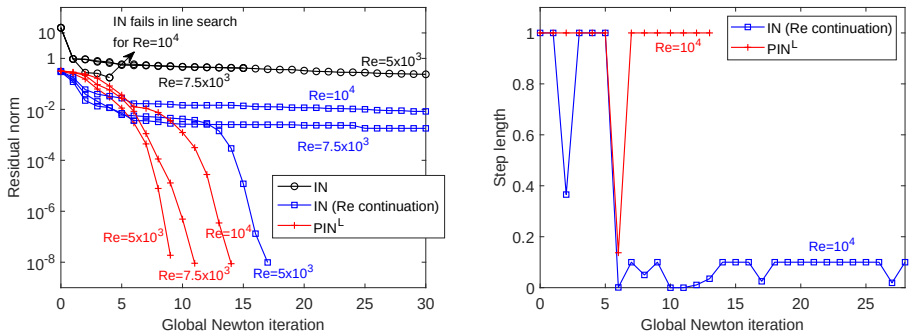
(a) $F^u(Y^0)$ (b) $\mathcal{F}^u(Y^0)$ (c) $F^u(Y^0)$-$\mathcal{F}^u(Y^0)$

(d) $F^u(Y^1)$ (e) $\mathcal{F}^u(Y^1)$ (f) $F^u(Y^1)$-$\mathcal{F}^u(Y^1)$

(g) $F^\omega(Y^1)$ (h) $\mathcal{F}^\omega(Y^1)$ (i) $F^\omega(Y^1)$-$\mathcal{F}^\omega(Y^1)$

(j) $F^\omega(Y^2)$ (k) $\mathcal{F}^\omega(Y^2)$ (l) $F^\omega(Y^2)$-$\mathcal{F}^\omega(Y^2)$

FIG. 8. *The residual of components u and ω at different subspace Newton steps ($j = 0, 1, 2$) obtained using PIN$^{\mathcal{L}}$ for the cavity flow with $Re = 600$. The mesh size is 257×257, $d = 5$, $\gamma_r^s = 10^{-3}$. (a),(d),(g),(j) are the residuals computed using the original nonlinear function F. (b),(e),(h),(k) are the residuals computed using the approximated function $\mathcal{F}$. (c),(f),(i),(l) are the difference between $F(Y^j)$ and $\mathcal{F}(Y^j)$.*



(a) $\|F(X^{(i)})\|$ (b) $\lambda^{(i)}$

FIG. 9. *(a) Nonlinear residual history obtained using IN, the Reynolds number continuation approach, and PIN$^{\mathcal{L}}$ for the cavity flow problem. (b) The step length $\lambda^{(i)}$ in the global Newton iteration for the case with $Re = 10^4$. The mesh size is 257×257, $d = 10$, and $\gamma_r^s = 10^{-4}$.*

TABLE 2

The numbers of iterations and compute times obtained using IN with the Reynolds number continuation approach and PIN$^{\mathcal{L}}$ for the cavity flow problem. The mesh size is 257×257, $d = 10$, and $\gamma_r^s = 10^{-4}$.

Re	T_{train}(s)	NI_s	LI_s	T_{precon}(s)	NI_g	LI_g	T_{total}(s)
			IN (Re continuation)				
10^3					7	39.6	20.1
3.2×10^3					12	24.2	24.8
5×10^3					17	31.4	43.7
			PIN$^{\mathcal{L}}$				
10^3	0.8	5	1	2.4	9	23.1	23.7
3.2×10^3	0.8	3	1	1.5	10	24.8	24.4
5×10^3	0.8	3	1	1.4	10	25.7	26.5
7.5×10^3	0.8	3	1	1.4	11	21.3	22.8
10^4	0.8	3	1	1.4	14	22.1	30.5

in terms of the number of global Newton iterations. Since the output projector of PCA is used for the purpose of nonlinear preconditioning, the selection of d should be within a suitable range. On one hand, when d is too small, the principal components selected may not be sufficient to figure out the slow subspace of the residuals, and the subspace Newton may not converge when a small γ_r^s is used. On the other hand, when d is too large, the residual subspace may have no distinction from the original space so that solving the system in nonlinear preconditioning is as difficult as the original problem, which violates the purpose of preconditioning. In terms of the total compute time, the best choice for this case is $d = 10$, which is half the size of the dataset.

The dataset collected for PCA is another important factor that affects the performance of PIN$^{\mathcal{L}}$. With different datasets obtained using IN for cases $Re = 300 \sim 600$, we compare the results in Table 4. For each dataset we choose a suitable d to obtain the optimal performance. When the training problem is far from the original problem, i.e., the one obtained from Case $Re = 300$, the subspace projectors P and Q learned from this dataset are considered not good enough for preconditioning. With such preconditioning PIN$^{\mathcal{L}}$ needs more global Newton iterations and more compute

TABLE 3

The impact of parameters d and γ_r^s on the performance of PIN$^{\mathcal{L}}$ for the cavity flow problem with $Re = 10^3$. The mesh size is 257×257. "" means the subspace Newton does not converge and returns the intermediate solution at this step.*

	$\gamma_r^s = 10^{-3}$			$\gamma_r^s = 10^{-4}$			$\gamma_r^s = 10^{-5}$		
d	NI_s	NI_g	T_{total}(s)	NI_s	NI_g	T_{total}(s)	NI_s	NI_g	T_{total}(s)
5	3	19	40.8	6*	19	45.6	6*	19	45.1
7	6	11	28.5	9*	10	29.7	9*	10	28.1
10	4	9	21.2	5	9	23.7	7*	9	26.6
12	5	11	29.6	7	10	26.6	7	10	27.0

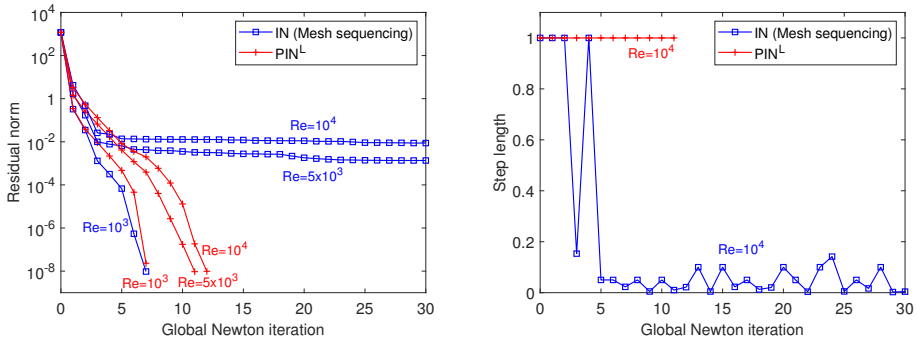
time to converge, or not converge at all for the difficult case $Re = 10^4$. We remark that the classical IN fails for $Re > 700$ and the resulting dataset does not work well. One way to provide useful dataset for preconditioning high Re problems is by using PIN$^{\mathcal{L}}$ instead of IN. For example, with the dataset obtained using PIN$^{\mathcal{L}}$ for case $Re = 5 \times 10^3$, choosing only one principal component ($d = 1$) in the nonlinear preconditioning is sufficient for case $Re = 10^4$ to converge. With a suitable choice of d and γ_r^s, solution of more difficult cases with $Re = 10^4 \sim 2 \times 10^4$ can be obtained by the proposed PIN$^{\mathcal{L}}$, as shown in the table.

TABLE 4

The impact of dataset on the performance of PIN$^{\mathcal{L}}$ for the cavity flow problem. The mesh size is 257×257. "s" refers to the number of vectors (samples) in the dataset. "–" indicates that the case fails to converge.

Re	Data collection method	s	d	γ_r^s	T_{train}(s)	NI_g	T_{total}(s)
5×10^3	IN for $Re = 300$	18	5	10^{-4}	0.6	14	31.2
	IN for $Re = 400$	17	7	10^{-4}	0.6	11	27.6
	IN for $Re = 600$	20	10	10^{-4}	0.8	10	26.5
10^4	IN for $Re = 300$	18	5	10^{-4}	0.6	–	–
	IN for $Re = 400$	17	7	10^{-4}	0.6	19	49.9
	IN for $Re = 600$	20	10	10^{-4}	0.8	14	30.5
10^4	PIN$^{\mathcal{L}}$ for $Re = 5 \times 10^3$	10	1	10^{-1}	0.2	9	21.7
1.5×10^4	PIN$^{\mathcal{L}}$ for $Re = 5 \times 10^3$	10	2	10^{-2}	0.3	10	24.8
2×10^4	PIN$^{\mathcal{L}}$ for $Re = 5 \times 10^3$	10	4	10^{-2}	0.3	14	32.3

3.1.4. Performance of training and preconditioning on a coarser mesh. We next study the convergence of the proposed method using a fine mesh 513×513 for the cavity flow problem. As discussed in Section 2.3, we perform the training step and the nonlinear preconditioning step on a coarser mesh 257×257, and project the corrected initial guess to the fine mesh using a standard linear interpolation. We use the same dataset and parameters as in the convergence test on the coarse mesh (Table 2), except for selecting a larger restart value 100 for GMRES and $(\eta_0, \beta) = (0.25, 10^{-3})$ for the forcing term. We compare the convergence of PIN$^{\mathcal{L}}$ with the mesh sequencing approach [27] in which the solution obtained for case $Re = 600$ on the coarse mesh is interpolated to the fine mesh as an initial guess. Figure 10 shows the nonlinear residual history and the step length. It is observed that PIN$^{\mathcal{L}}$ with the proposed coarse mesh preconditioning converges well for $Re = 10^3 \sim 10^4$ within 12 global Newton steps. In contrast, the mesh sequencing approach fails to converge for almost all cases except using $\beta = \infty$ for case $Re = 10^3$. The results show that PIN$^{\mathcal{L}}$ yields better convergence and robustness compared with the mesh sequencing approach for problems with a large Re defined on a fine mesh.



(a) $\|F(X^{(i)})\|$ (b) $\lambda^{(i)}$

FIG. 10. *(a) Nonlinear residual history and (b) step length for the cavity flow problem obtained using the mesh sequencing approach and PIN$^{\mathcal{L}}$ with training and preconditioning on a coarser mesh. The size of the fine mesh is* 513×513*, the size of the coarse mesh is* 257×257*.* $d = 10$*, and* $\gamma_r^s = 10^{-4}$*.*

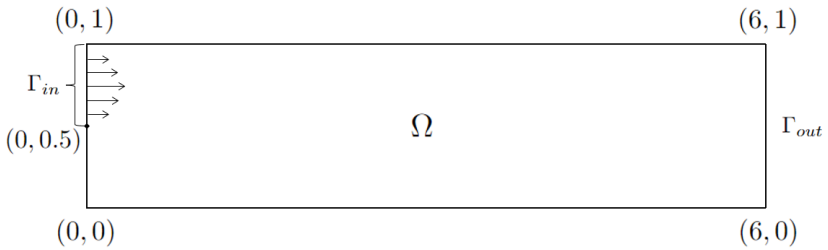


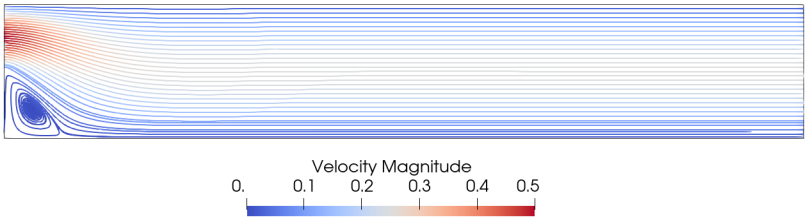
FIG. 11. *The computational domain for the backward-facing step flow problem.*

3.2. The backward-facing step flow problem. In this section, we consider the backward-facing step flow problem defined on a channel $\Omega = (0, 6) \times (0, 1)$ as shown in Figure 11. A fully developed parabolic velocity profile is specified at the inlet boundary $\Gamma_{in} : x = 0, 0.5 \le y \le 1$; an outflow boundary condition is given on the right boundary $\Gamma_{out} : x = 6$; on the other boundaries $\partial\Omega \setminus (\Gamma_{in} \cup \Gamma_{out})$ we impose no-slip and no-penetration conditions, specifically,

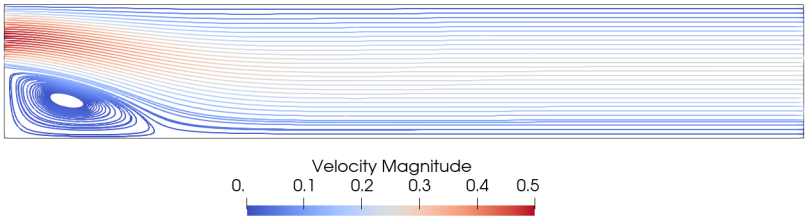
(3.5)
$$\begin{cases} u = 8(0.5 - y)(y - 1), \quad v = 0, \quad \omega = \frac{\partial v}{\partial x} + 16y - 12, & \text{on} \quad \Gamma_{in}, \\ u = -y(y - 1), \quad v = 0, \quad \omega = \frac{\partial v}{\partial x} + 2y - 1, & \text{on} \quad \Gamma_{out}, \\ u = 0, \quad v = 0, \quad \omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, & \text{on} \quad \partial\Omega \setminus (\Gamma_{in} \cup \Gamma_{out}). \end{cases}$$

The boundary condition for the vorticity on Γ_{in} and Γ_{out} is discretized with a second-order central finite difference method. For the boundary condition on $\partial\Omega\setminus(\Gamma_{in}\cup\Gamma_{out})$ we use the same discretization as in the driven cavity flow problem. The mesh size used for this case is 481×81. Figure 12 shows the streamlines for the backward-facing step flow with $Re = 50$, 200, and 1.2×10^3, respectively. A vortex appears at the bottom left region caused by the flow separation, and its size develops with the increase of Re.

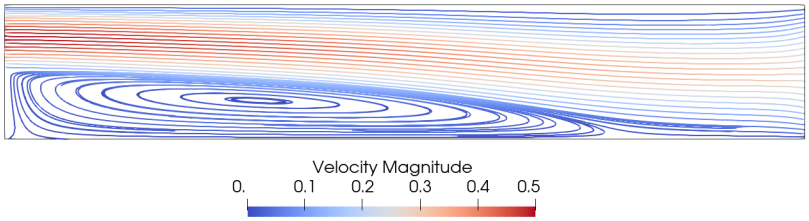
For this problem, the classical IN fails to converge when $Re \ge 800$. We compare the performance of the proposed method with a multilayer pointwise PIN-NE



(a) $Re = 50$



(b) $Re = 200$



(c) $Re = 1.2 \times 10^3$

FIG. 12. *Streamlines for the backward-facing step flows with different Reynolds numbers. The mesh size is* 481×81.

approach that is efficient for solving incompressible flows with high Reynolds numbers [35]. At the kth global Newton step, the components corresponding to a mesh point (i, j) are eliminated if $\|F(X^k)\|/\|F(X^{k-1})\| \geq 0.9$ and

$$\max\{|F^u_{ij}(X^k)|, |F^v_{ij}(X^k)|, |F^\omega_{ij}(X^k)|\} > \rho_l \|F(X^k)\|_\infty, \tag{3.6}$$

where ρ_l is a preselected parameter used for determining the number of the to-be-eliminated components on the lth layer. We refer to [35] for more details of this approach. In the test, we consider a single-layer approach with $\rho_1 = 10^{-2}$ and a two-layer approach with $(\rho_1, \rho_2) = (10^{-2}, 10^{-3})$. For PIN$^{\mathcal{L}}$, the dataset collected for PCA is obtained by using IN for case $Re = 200$, consisting of 7 vectors in F and X. The number of principal components used is $d = 4$. We use the same relative tolerance $\gamma_r^s = 10^{-3}$ for all methods in the test. Figure 13 shows the nonlinear residual history obtained using IN, the single-layer PIN-NE, the two-layer PIN-NE, and PIN$^{\mathcal{L}}$ for cases with $Re = 800$, 10^3, and 1.2×10^3. A detailed comparison for the numbers of iterations and the total compute times is shown in Table 5. As Re increases, both the single-layer PIN-NE and the two-layer PIN-NE result in more global Newton iterations. Note that the single-layer PIN-NE fails in line search for case $Re = 1.2 \times 10^3$. The two-layer approach significantly improves the convergence of the single-layer approach. Compared to PIN-NE, PIN$^{\mathcal{L}}$ saves more than half of the global

Newton steps and half of the total compute time for the difficult case $Re = 1.2 \times 10^3$. We summarize the observations as follows: (1) In PIN$^{\mathcal{L}}$, the subspace Newton is performed only once before the global Newton is called, in contrast to the PIN-NE methods that usually perform subspace Newton multiple times when NE is activated adaptively. (2) The compute time spent for the subspace Newton iteration in PIN$^{\mathcal{L}}$ is much smaller than the NE approaches since the dimension of the subspace Jacobian problem (d-dimensions) is rather smaller compared to the dimension controlled by ρ_l in PIN-NE. (3) PIN$^{\mathcal{L}}$ results in a fixed number of global Newton iteration that is independent of Re for this problem, which shows the robustness of the proposed method for nonlinearly difficult problems.

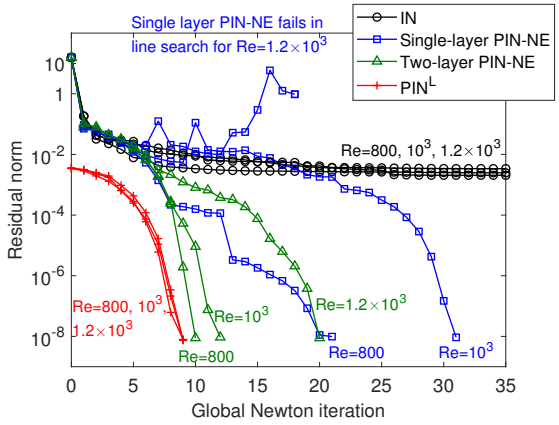


FIG. 13. *Nonlinear residual history obtained using IN, single-layer PIN-NE, two-layer PIN-NE, and PIN$^{\mathcal{L}}$ for the backward-facing step flow problem.*

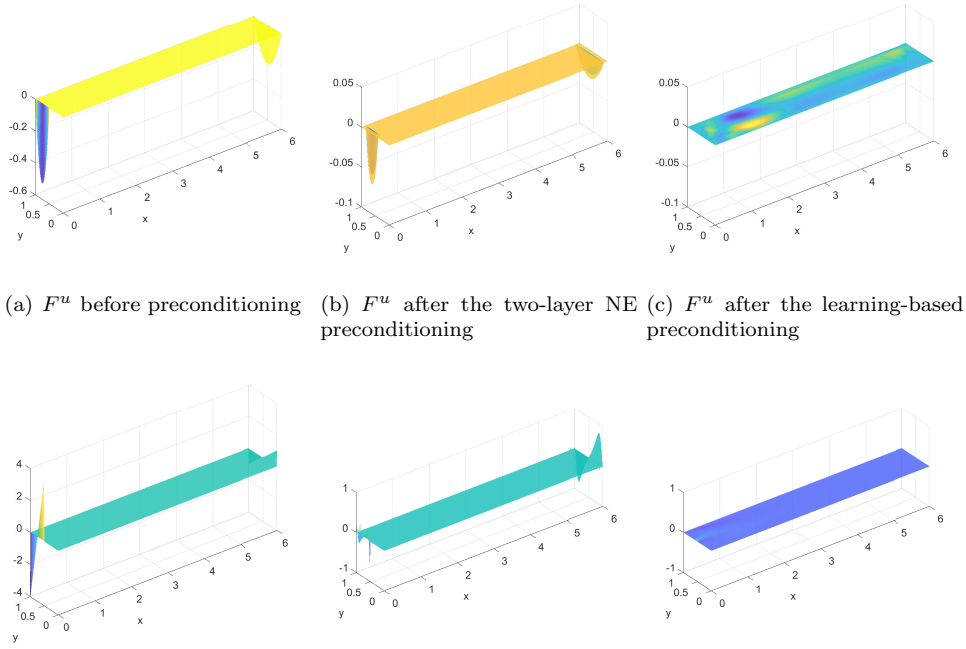
TABLE 5

The numbers of iterations and compute times obtained using the single-layer PIN-NE, the two-layer PIN-NE, and PIN$^{\mathcal{L}}$ for the backward-facing step flow problem. "N_{ne}" is the number of NE applications in PIN-NE.

Re	T_{train}(s)	N_{ne}	NI_s	LI_s	T_{precon}(s)	NI_g	LI_g	T_{total}(s)
				Single-layer PIN-NE				
800		2	3	19.2	6.3	21	12.0	25.3
10^3		6	8	22.7	50.8	31	9.9	80.6
				Two-layer PIN-NE				
800		1	2.5	1.2	1.1	10	16.1	10.5
10^3		1	2.5	1.2	1.1	12	17.1	12.3
1.2×10^3		1	2.5	1.2	1.1	20	17.6	20.7
				PIN$^{\mathcal{L}}$				
800	0.1		1	1	0.2	9	14.7	8.7
10^3	0.1		1	1	0.2	9	16.2	9.1
1.2×10^3	0.1		1	1	0.2	9	16.6	9.5

To explore how PIN$^{\mathcal{L}}$ improves the convergence, we show in Figure 14 the residual of components u and ω before and after the nonlinear preconditioning for the backward-facing step flow with $Re = 1.2 \times 10^3$. From Figure 14 (a) and (d) we observe that the local high nonlinearities cluster around the inlet and outlet boundaries. After the two-layer NE preconditioning, such nonlinearities are reduced by a factor of 10. In comparison, the learning-based preconditioning reduces the nonlinearities

by a factor of 10^4 and returns a better initial guess for the global Newton iteration. The comparison results indicate that the learning-based approach is more powerful to identify and balance the nonlinearities of the system compared to the NE approach.



(a) F^u before preconditioning (b) F^u after the two-layer NE preconditioning (c) F^u after the learning-based preconditioning

(d) F^ω before preconditioning (e) F^ω after the two-layer NE preconditioning (f) F^ω after the learning-based preconditioning

FIG. 14. *The residual of components u and ω before and after the two-layer NE preconditioning and the learning-based preconditioning for the backward-facing step flow problem with $Re = 1.2\times10^3$.*

4. Concluding remarks. We propose and study a novel nonlinearly preconditioned inexact Newton method with learning capability for solving nonlinear system of algebraic equations. The preconditioner is constructed by a decomposition of the nonlinear residual space into two subspaces; one corresponds to the low frequency subspace and the other corresponds to the high frequency subspace. Such a decomposition is obtained by a PCA based unsupervised learning method from a training problem. The nonlinear preconditioner is applied to produce a better initial guess for the global Newton iteration, within which a projected low dimensional Jacobian system is constructed and solved at each subspace Newton iteration. The new method features a low computational cost and is capable of balancing the overall nonlinearity effectively. We test the algorithm with extensive numerical experiments for high Reynolds number incompressible flow problems. Results show that the proposed method is more robust and faster than other popular nonlinear solvers, such as PIN-NE and the classical IN with globalization techniques such as parameter continuation and mesh sequencing. The focus of the paper was on the incompressible flow problems, but the algorithm is algebraic, and is expected to work for other highly nonlinear problems.

REFERENCES

[1] V. F. de Almeida and J. J. Derby, *Construction of solution curves for large two-dimensional problems of steady-state flows of incompressible fluids*, SIAM J. Sci. Comput., 22 (2000), pp. 285–311.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al., *LAPACK Users' Guide*, Volume 9. SIAM, 1999.

[3] P. Astrid, S.Weiland, K.Willcox, and T. Backx, *Missing point estimation in models described by proper orthogonal decomposition*, IEEE Trans. Automat. Contr., 53 (2008), pp. 2237–2250.

[4] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc Users Manual*, Argonne National Laboratory, 2022.

[5] B. Bochenek and Z. Ustrnul, *Machine learning in weather prediction and climate analyses–applications and perspectives*, Atmosphere, 13 (2022), 180.

[6] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, *Machine learning for fluid mechanics*, Annu. Rev. Fluid Mech., 52 (2020), pp. 477–508.

[7] X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200.

[8] 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.

[9] K. Carlberg, V. Forstall, and R. Tuminaro, *Krylov-subspace recycling via the POD-augmented conjugate-gradient method*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1304–1336.

[10] F. Chaouqui, M. J Gander, P. M. Kumbhar, and T. Vanzan, *Linear and nonlinear substructured restricted additive Schwarz iterations and preconditioning*, Numer Algor., 2022.

[11] A. Chatterjee, *An introduction to the proper orthogonal decomposition*, Curr. Sci., 78 (2000), pp. 808–817.

[12] G. Ciaramella and M. J. Gander, *Iterative Methods and Preconditioners for Systems of Linear Equations*, SIAM, 2022.

[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, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[16] U. Ghia, K. N. Ghia, and C. T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys., 48 (1982), pp. 387–411.

[17] J. Huang, C. Yang, and X.-C. Cai, *A nonlinearly preconditioned inexact Newton algorithm for steady lattice Boltzmann equations*, SIAM J. Sci. Comput., 38 (2016), pp. 1701–1724.

[18] 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.

[19] 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.

[20] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, *Physics-informed machine learning*, Nat. Rev. Phys., 3 (2021), pp. 422–440.

[21] P. Kerfriden, P. Gosselet, S. Adhikari, and S. P. A. Bordas, *Bridging proper orthogonal decomposition methods and augmented Newton-Krylov algorithms: An adaptive model order reduction for highly nonlinear mechanical problems*, Comput. Methods Appl. Mech. Eng., 200 (2011), pp. 850–866.

[22] A. Klawonn, M. Lanser, and O. Rheinbach, *Nonlinear FETI-DP and BDDC methods*, SIAM J. Sci. Comput., 36 (2014), pp. A737–A765.

[23] A. Klawonn, M. Lanser, and O. Rheinbach, *Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations*, SIAM J. Sci. Comput., 37 (2015), pp. C667–C696.

[24] A. Klawonn, M. Lanser, and M. Uran, *Adaptive nonlinear elimination in nonlinear FETI-*

DP methods, Technical Report, Universität zu Köln, 2021.
[25] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
[26] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Sci. Comput., 40 (2002), pp. 492–515.
[27] W. Layton, H. K. Lee, and J. Peterson, *Numerical solution of the stationary Navier-Stokes equations using a multilevel finite element method*, SIAM J. Sci. Comput., 20 (1998), pp. 1–12.
[28] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature, 521 (2015), pp. 436–444.
[29] J. Li and X.-C. Cai, *Summation pollution of principal component analysis and an improved algorithm for location sensitive data*, Numer. Linear Algebr., 28 (2021), e2370.
[30] L. Liu and D. E. Keyes, *Field-split preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 37 (2015), pp. 1388–1409.
[31] L. Liu and D. E. Keyes, *Convergence analysis for the multiplicative Schwarz preconditioned inexact Newton algorithm*, SIAM J. Numer. Anal., 54 (2016), pp. 3145–3166.
[32] L. Liu, D. E. Keyes, and R. Krause, *A note on adaptive nonlinear preconditioning techniques*, SIAM J. Sci. Comput., 40 (2018), pp. 1171–1186.
[33] B. Liu, M. Mohandes, H. Nuha, M. Deriche, and F. Fekri, *A distributed principal component analysis compression for smart seismic acquisition networks*, IEEE Trans. Geosci. Remote. Sens., 56 (2018), pp. 3020–3029.
[34] 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), 108926.
[35] L. Luo, X.-C. Cai, Z. Yan, L. Xu, and D. E. Keyes, *A multilayer nonlinear elimination preconditioned inexact Newton method for steady-state incompressible flow problems in three dimensions*, SIAM J. Sci. Comput., 42 (2020), pp. B1404–B1428.
[36] L. Luo, X.-C. Cai,and D. E. Keyes, *Nonlinear preconditioning strategies for two-phase flows in porous media discretized by a fully implicit discontinuous Galerkin method*, SIAM J. Sci. Comput., 43 (2021), pp. S317–S344.
[37] R. Markovinović and J. D. Jansen, *Accelerating iterative solution methods using reduced-order models as solution predictors*, Int. J. Numer. Methods Eng., 68 (2006), pp. 525–541.
[38] D. Pasetto, M. Ferronato, and M. Putti, *A reduced order model-based preconditioner for the efficient solution of transient diffusion equations*, Int. J. Numer. Methods Eng., 109 (2017), pp. 1159–1179.
[39] E. 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.
[40] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., 378 (2019), pp. 686–707.
[41] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Second ed., 2003.
[42] J. N. Shadid, R. S. Tuminaro, and H. F. Walker, *An inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass transport*, J. Comput. Phys., 137 (1997), pp. 155–185.
[43] H. Yang, C. Yang, and S. Sun, *Active-set reduced-space methods with nonlinear elimination for two-phase flow problems in porous media*, SIAM J. Sci. Comput., 38 (2016), pp. 593–618.
[44] 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.
[45] 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.