A NOTE ON ADAPTIVE NONLINEAR PRECONDITIONING TECHNIQUES

LU LU LIU†, DAVID E. KEYES‡, AND ROLF KRAUSE§

Abstract. Nonlinear preconditioning is a globalization technique for Newton’s method applied to systems of equations with unbalanced nonlinearities, in which nonlinear residual norm reduction stagnates due to slowly evolving subsets of the degrees of freedom. Even though the Newton corrections may effectively be sparse, a standard Newton method still requires large ill-conditioned linear systems resulting from global linearizations of the nonlinear residual to be solved at each step. Nonlinear preconditioners may enable faster global convergence by shifting work to where it is most strategic, on subsets of the original system. They require additional computation per outer iteration while aiming for many fewer outer iterations and correspondingly fewer global synchronizations. In this work, we improve upon previous nonlinear preconditioning implementations by introducing parameters that allow turning off nonlinear preconditioning during outer Newton iterations where it is not needed. Numerical experiments show that the adaptive nonlinear preconditioning algorithm has performance similar to monolithically applied nonlinear preconditioning, preserving robustness for some challenging problems representative of several PDE-based applications while saving work on nonlinear subproblems.

Key words. nonlinear equations, Newton’s method, nonlinear preconditioning, domain decomposition, field splitting

AMS subject classifications. 65H10, 65H20, 65N22, 65N55

DOI. 10.1137/17M1128502

1. Introduction. Newton-like methods in their many variants, such as Newton–Krylov algorithms, are frequently employed in the implicit solution of large systems of algebraic equations arising from discretization of nonlinear partial differential equations (PDEs) in scientific and engineering applications. When a good initial guess is not available Newton-like methods often suffer from lack of robustness in the form of stagnation or slow convergence until a domain of superlinear or quadratic convergence is found. The majority of state variables may barely evolve over many iterations in situations characterized as having unbalanced nonlinearities [5, 7, 25]. Meanwhile, the solution of large linear systems with ill-conditioned Jacobians, often the phase of dominant cost in the overall simulation, is required at each Newton step [3].

Besides algebraic “blackbox” globalizations such as line search and trust region

†Department of Mathematics, School of Science, Nanjing University of Science and Technology, Nanjing, 210094, China, and Institute of Computational Science, Università della Svizzera italiana, Lugano, 6900, Switzerland (lulu.liu@usi.ch).
‡Program in Applied Mathematics and Computational Science and Center for Extreme Computing, King Abdullah University of Science and Technology, Thuwal, 23955–6900, Saudi Arabia (david.keyes@kaust.edu.sa).
§Institute of Computational Science, Università della Svizzera italiana, Lugano, 6900, Switzerland (rolf.krause@usi.ch).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
there are several PDE-specific globalization techniques for nonlinear systems on which Newton-like methods fail when applied directly, such as mesh continuation [35], parameter continuation [1], and pseudotransient continuation [11, 22]. Such continuation methods solve a sequence of problems related to the ultimate rootfinding problem, e.g., on coarsened meshes, at reduced physical loadings or initial and boundary forcings, or with a very small time steps, respectively. The intermediate problems in the continuation sequence have relatively rapid convergence rates and lead from a convenient initial iterate to an iterate that lies in the ball of convergence around the desired root of the ultimate problem. Nonlinear preconditioners offer a complementary technique to desensitize Newton to the initial guess and improve the nonlinear conditioning in a fundamental way. They may, if necessary, be further combined with the other globalization techniques mentioned above.

Nonlinear preconditioning can be applied on the left or right side of nonlinear functions just as linear preconditioning may be. When left nonlinear preconditioning is effective, we solve a modified system with better conditioning and better balanced nonlinearities than the original system using a Jacobian-free [25] inexact Newton method. Examples include the additive or multiplicative Schwarz preconditioned inexact Newton methods, referred to as ASPIN [5] or MSPIN [27], or a restricted nonlinear Schwarz preconditioner named RASPEN in [15]. For nonlinear programming problems, a novel, globally convergent preconditioned trust-region approach with the inherently parallel ASPIN strategy was recently presented in [17]. Left nonlinear preconditioners have been employed successfully to solve some problems that are challenging for Newton in their original coordinates, such as a transonic full potential flow [7], incompressible Navier–Stokes flow at high Reynolds numbers [20], and two-phase flow in porous media [28, 33, 34]. In contrast, right nonlinear preconditioners recast the basis for the solution space by a nonlinear transformation and find the solution in the transformed space rather than in the original space. Nonlinear elimination (NE) [26] handles nonlinear “imbalance” by removing variables deemed to cause trouble before applying a global Newton iteration. The NE algorithm can serve as a right nonlinear preconditioner. NE is employed to provide a new starting point for the outer Newton iteration in a subspace correction phase; then the solution is updated in the whole space. With different partitioning strategies, this leads to a domain-based form called NKS-RAS [8] and a field-based form called INB-NE or RS-NE [36, 37]. Nonlinear FETI-DP domain decomposition approaches have also attracted increasing attention; see [24].

Left and right nonlinear preconditioning techniques deal with nonlinearities by replacing nonlinear functions or changing unknowns, respectively, but they both require inner iterations working on subsets of the original system, which result in additional cost per step. However, nonlinear preconditioning need not be invoked on every outer Newton iteration [8], especially when the approximate solution lies in the domain of convergence. In this paper, we propose an adaptive algorithm based on parameters that are natural byproducts of the iteration, in which nonlinear preconditioning is activated at the current iteration only if parameters reflect low quality of the previous Newton iteration.

The rest of the paper is organized as follows. In section 2, we compare the performance of INB and nonlinear preconditioners for an illustrative example of just two scalar components, which motivates the work in this paper. In section 3, we describe an adaptive nonlinear preconditioning algorithm, which introduces two complementary tuning parameters. In section 4, the benefit of adaptive nonlinear preconditioning is illustrated on three challenging nonlinear model problems—flame sheet combustion,
two-phase reservoir flow, and lid-driven cavity flow—and we summarize.

2. A simple example. To motivate the main contribution of this paper in section 3, in this section we reference three nonlinear preconditioning techniques and compare them to the baseline inexact Newton algorithm on a simple two-component example.

The classical inexact Newton algorithm with backtracking (INB) [13, 14, 19, 21] is customarily equipped with Eisenstat–Walker evolution of the inner linear solver tolerance [16], as implemented, for example, in [2, 19, 30].

The first use of nonlinear preconditioning in the Newton context is believed to be the additive Schwarz preconditioning inexact Newton (ASPIN) algorithm [5]. It was introduced and is generally applied in a domain decomposition approach, to exploit high concurrency. The multiplicative Schwarz preconditioning inexact Newton (MSPIN) algorithm [27], a Gauss–Seidel-like version of the Jacobi-like ASPIN, is designed for a field-split approach, to achieve robustness and to allow construction of a full-system preconditioner from solvers for different physical subsystems. ASPIN and MSPIN can be nested within each other in either order. For example, each sequential stage of MSPIN can be domain-decomposed with ASPIN. Both ASPIN and MSPIN should be thought of as left preconditioners. Left nonlinear preconditioning changes the coordinates of the nonlinear residual, leading to a new system with the same root. The preconditioned system is generally solved by an outer Newton method in Jacobian-free form, so that no new code for function or Jacobian evaluation need be written. ASPIN and MSPIN may be regarded as generalizations of linear preconditioners, to which they reduce if the original system is linear. They create a modified global residual function from the union of subspace corrections of inner Newton iterations, then apply to it a global INB that accesses the original function within each subspace.

In contrast, nonlinear elimination (INB-NE) [36] is a right preconditioner that for each outer step of INB modifies the input iterate by applying an inner INB to a subspace of the global solution space. Hence, as in the understanding of “right” in linear preconditioning, it changes the coordinates of the solution; however, the switch from left to right is not as straightforward in the nonlinear case as specifying on which side to apply a linear transformation. The subspace of components to be eliminated at each step may often usefully be identified by an association of solution degrees of freedom with the components of large magnitude in the global residual. Alternatively, in the numerical experiments of this paper, we focus on the special class of NE in which the degrees of freedom to be eliminated at each stage are identified with different fields, independent of any measurement of the magnitudes of individual residual components.

In this note, INB, ASPIN, MSPIN, and INB-NE are taken directly from the algorithmic definitions in the references above. For a more general discussion of nonlinear preconditioning, see [4].

For the discrete nonlinear function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, given an initial iterate $x \in \mathbb{R}^n$, we want to find a solution vector $x^* \in \mathbb{R}^n$ such that

$$F(x^*) = 0,$$

where $F = [F_1, F_2, \ldots, F_n]^T$ and $x = [x_1, x_2, \ldots, x_n]^T$.

To illustrate the methods, we consider the system $F(x)$ of two nonlinear equations in two unknowns [36],

$$F(x_1, x_2) = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} (x_1 - x_2^3 + 1)^3 - x_2^3 \\ x_1 + 2x_2 - 3 \end{bmatrix} = 0$$
The number of nonlinear iterations starting from different points. The relative tolerance is $10^{-6}$ and $10^{-8}$ for nonlinear and linear solvers, respectively.

<table>
<thead>
<tr>
<th>Initial guess</th>
<th>INB</th>
<th>ASPIN</th>
<th>ASPIN+INB</th>
<th>MSPIN</th>
<th>MSPIN+INB</th>
<th>INB-NE</th>
<th>NE+INB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0 = [0,0]^T$</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$x_0 = [0,2]^T$</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$x_0 = [2,0]^T$</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$x_0 = [2,2]^T$</td>
<td>11</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

with root $x^* = [1,1]^T$.

We run tests starting from the four different initial estimates surrounding the root: $x_0 = [0,0]^T$, $x_0 = [0,2]^T$, $x_0 = [2,0]^T$, $x_0 = [2,2]^T$. In Table 1, we compare the number of nonlinear iterations using INB, ASPIN, MSPIN, INB-NE, and the following combination approaches of INB and nonlinear preconditioners:

- ASPIN+INB: ASPIN is applied for the first three Newton iterations and INB for the remaining iterations.
- MSPIN+INB: MSPIN is applied for the first three Newton iterations and INB for the remaining iterations.
- NE+INB: INB-NE is applied for the first three Newton iterations and INB for the remaining iterations.

Compared with the six nonlinear preconditioners and combination approaches, the number of nonlinear iterations using INB is more sensitive to the initial guess. All three nonlinear preconditionings and their corresponding combination approaches are effective in desensitizing Newton convergence to the initial guess, and they have similar performance in terms of numbers of nonlinear iterations. The cost per preconditioned iteration may, however, depending upon the implementation, be more expensive than a global step of INB. It seems unnecessary to carry out nonlinear preconditioning for each Newton iteration, which inspires adaptive nonlinear preconditioning, as described in the next section.

### 3. Adaptive nonlinear preconditioning techniques

The nonlinear preconditioning techniques defined above are inner-outer Newton solvers, which require solving nonlinear subproblems in inner iterations to define either the residual (left-hand form) or the initial iterate (right-hand form) of the respective outer iteration.

In this section, we propose parameter-based nonlinear preconditioning techniques to realize a trade-off between the amount of computation dedicated to inner solves and the convergence of global outer Newton iterations. The parameters determine whether nonlinear preconditioning is applied for each Newton iteration. The goal is to take advantage of the globalizing benefits of nonlinear preconditioning only where necessary and otherwise reduce the number of nonlinear subproblem instances.

#### 3.1. Convergence monitors $\eta_k$ and $\rho_k$

We introduce two parameters that are convenient byproducts of norms generally available in Newton-like methods, where $s_{k-1}$ is the previously accepted (perhaps modified) Newton correction:

$$
\eta_k = \frac{||F(x_k)|| - ||F(x_{k-1}) + F'(x_{k-1})s_{k-1}||}{||F(x_{k-1})||}, \quad k = 1, 2, \ldots,
$$

and

$$
\rho_k = \frac{\text{red}}{\text{pred}} = \frac{||F(x_{k-1})|| - ||F(x_k)||}{||F(x_{k-1})|| - ||F(x_{k-1}) + F'(x_{k-1})s_{k-1}||}, \quad k = 1, 2, \ldots.
$$
The parameter $\eta_k$ directly reflects the agreement between $F$ and its local linear model at the previous step, and the parameter $\rho_k$ is the ratio of the actual reduction to the predicted reduction at the previous step. We use these two parameters to determine whether a simple Newton iteration or a nonlinearly preconditioned Newton iteration is used for the current step.

We first show that $\rho_k + \eta_k \approx 1$ in common practice, so that the choice of which parameter to use is essentially unimportant. Using the definition of $\rho_k$ in (4), we can rewrite $\rho_k$ as

$$
\rho_k = 1 - \frac{\|F(x_k)\|}{\|F(x_{k-1})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|}.
$$

In this paper, we assume that an inexact Newton method is used to solve Jacobian systems and that it requires that $\|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\| \leq \epsilon$ $\|F(x_{k-1})\|$, where the relative tolerance $\epsilon$ is typically set to a small value, e.g., $10^{-4}$. Therefore, it holds that

$$
1 \leq \frac{\|F(x_{k-1})\|}{\|F(x_{k-1})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|} \leq \frac{1}{1 - \epsilon},
$$

from which we have

$$
\frac{\|F(x_{k-1})\|}{\|F(x_{k-1})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|} \approx 1.
$$

There are two cases for the numerator in (5):

- When it holds that
  $$
  \|F(x_k)\| > \epsilon\|F(x_{k-1})\| \geq \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|,
  $$
  following (5) and (7), we obtain
  $$
  \rho_k = 1 - \eta_k \frac{\|F(x_{k-1})\|}{\|F(x_{k-1})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|} \approx 1 - \eta_k,
  $$
  and it is easy to see $\rho_k + \eta_k \approx 1$.

- When $\|F(x_{k})\| \leq \epsilon\|F(x_{k-1})\|$, we have
  $$
  \|F(x_{k})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\| \leq 2\epsilon\|F(x_{k-1})\|,
  $$
  which shows that $\eta_k \leq 2\epsilon$. From (5) and (6), it holds that
  $$
  \rho_k \leq 1 + \eta_k \frac{\|F(x_{k-1})\|}{\|F(x_{k-1})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|} \leq 1 + \frac{2\epsilon}{1 - \epsilon}
  $$
  and
  $$
  \rho_k \geq 1 - \eta_k \frac{\|F(x_{k-1})\|}{\|F(x_{k-1})\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|} \geq 1 - \frac{2\epsilon}{1 - \epsilon}.
  $$

Therefore,

$$
1 - \frac{2\epsilon}{1 - \epsilon} \leq \rho_k + \eta_k \leq 1 + \frac{2\epsilon}{1 - \epsilon} + 2\epsilon.
$$

Recalling that $\epsilon$ is typically small, we again get $\rho_k + \eta_k \approx 1$. 

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
Algorithm 1. Adaptive nonlinear preconditioning strategy.

Initial guess $x^{(0)}$.
Set $\eta_0 = 1$ or $\rho_0 = 0$.

while $k = 0, 1, 2, \ldots$ until convergence do
    Step 1. Update $x^{(k+1)}$ starting from $x^{(k)}$ based on parameters $\eta_k$ or $\rho_k$:
    if $\eta_k < \epsilon_1$ or $\rho_k > \epsilon_2$ then
        Implement one step INB.
    else
        Implement one step of nonlinear preconditioners (ASPIN, MSPIN, or INB-NE).
    end if
    Step 2. Compute $\eta_{k+1}$ or $\rho_{k+1}$.
    Step 3. $k = k + 1$.
end while

3.2. The basic algorithm. In Algorithm 1, we describe the general framework of the adaptive nonlinear preconditioning algorithm based on the parameters $\eta_k$ or $\rho_k$ introduced above.

We set $x^{(0)}$ as the initial guess. It can be split into subvectors, possibly overlapping or nonoverlapping, corresponding to subdomain-based or field-based partitioning. At the $k$th iteration, the current approximate solution is denoted by $x^{(k)}$ and $x^{(k+1)}$ is the new approximate solution. Parameters $\eta_k$ and $\rho_k$ can be used to measure the quality of the previous Newton iteration based on global linearization. Nonlinear preconditioning will be employed instead of INB to update $x^{(k)}$ when the reduction of nonlinear residuals has barely evolved. Since $\eta_k + \rho_k \approx 1$, we can pick either one to control whether one-step INB or one-step nonlinear preconditioning is implemented for the current step.

Some practical remarks about Algorithm 1 follow:
1. The parameters $\epsilon_1$ and $\epsilon_2$ are preselected constants. We set $\epsilon_1 = 0.2$ and $\epsilon_2 = 0.8$ for all numerical tests in this paper.
2. When the initial guess is not good, nonlinear preconditioners often result in a superior reduction at the first iteration relative to INB. Therefore, the default option for the first iteration is one step of nonlinearly preconditioned INB, by setting $\eta_0 = 1$ and $\rho_0 = 0$. This is illustrated graphically in Figure 1 for the two-component example in section 2. We plot the contours of $\log(||F(x)|| + 1)$ and the path history of ASPIN, MSPIN, and INB-NE for various initial points. In comparison to INB, the nonlinear preconditioners all provide in their first step a better reduction in the merit function and a larger step length than INB. Figure 2 replots the same histories zoomed out on a consistent scale for all four starting guesses and all three preconditioning methods, wherein it can be seen that right preconditioning introduces some wide excursions.
3. In contrast with INB and right nonlinear preconditioning, left nonlinear preconditioning based on ASPIN or MSPIN solves a system based on a modified residual vector. In Algorithm 1, it is more convenient to stop the global Newton iterations by monitoring the residual of the original system:

$$||F(x^{(k)})|| \leq \max\{\epsilon_{\text{global-linear-rtol}}||F(x^{(0)})||, \epsilon_{\text{global-linear-atol}}\}.$$
4. The preconditioners have different implications for large-scale parallel implementation. The variable workloads of the local problems in the domain-based form of ASPIN, in which the nonlinear subproblems are solved concurrently, may lead to load imbalance among the processors. The load balancing issue essentially disappears for the field-split methods because the concurrent problems on the subdomains (in a parallel implementation) are linear and generally easy to load balance. Load balancing issues with right preconditioning must be considered case by case. In this paper, we merely note these issues and leave their practical resolution to future work that takes into account the cost of data movement in rebalancing load.

5. Even though the domain-based ASPIN is typically more robust than INB and is a good default strategy, local subproblems may still fail to converge due to unbalanced nonlinearities or the lack of a good initial guess. The field-split algorithms such as MSPIN or INB-NE provide additional options for local nonlinear solves of ASPIN—that is, the preconditioning strategies may be further nested.

![Contours of ||F(x)||](image)

**Fig. 1.** The path history using INB, ASPIN, MSPIN, and INB-NE for the example in section 2 from a variety of starting points. Note that the exact solution $x^* = [1, 1]^T$ is located within the valley. The red triangle, the blue diamond, the blue circle, and the blue square represent the intermediate solution $x^{(k+1)}$ corresponding to INB, ASPIN, MSPIN, and INB-NE, respectively.
Fig. 2. The path history using INB, ASPIN, MSPIN, and INB-NE for the example in section 2. The red triangle, the blue diamond, and the blue circle represent the intermediate solution $x^{(k+1)}$ corresponding to INB, ASPIN, and MSPIN, respectively. For INB-NE, the blue star represents the solution $\tilde{x}^{(k)}$ after the subspace correction phrase and the blue square represents the solution $x^{(k+1)}$ after the global updates.
4. Numerical results. In this section, we apply parameter-based nonlinear preconditioning techniques to three model boundary value problems. In the first, INB-NE in the field-split manner is considered for a flame sheet model coupling the conservation of mass and momentum with a Shvab-Zeldovich formulation for the energy and species fraction. In the second, ASPIN is applied to physical models for two-phase flow in porous media with a high-contrast permeability field. Considering the partitions of physical fields, in the third, we implement MSPIN for the lid-driven cavity flow problem. One partition contains all the vorticity unknowns in a Navier–Stokes problem, which come from a nonlinear PDE with the velocities as coefficients, and the other contains the velocity unknowns, which satisfy linear equations for a given vorticity distribution.

All of numerical experiments are done in the Portable, Extensible Toolkit for Scientific Computation (PETSc) [2].

4.1. One-dimensional flame sheet. We consider a flame sheet model for a diluted methane-air flame in a double-jet configuration [23, 31]. There are three unknowns: the similarity function $f'$, the mass flux in the transverse direction $V$, and the conserved scalar $S$.

\begin{align*}
\frac{dV}{dy} + 2a\rho f' &= 0, \\
\frac{d}{dy} \left( \mu \frac{df'}{dy} \right) - V \frac{df'}{dy} + a(\rho_\infty - \rho(f')^2) &= 0, \\
\frac{d}{dy} \left( \rho D \frac{dS}{dy} \right) - V \frac{dS}{dy} &= 0.
\end{align*}

(11) (12) (13)

We use a power law for the viscosity $\mu$ as a function of temperature and express mass diffusivities $\rho D$ in terms of viscosity through the Prandtl number:

\begin{align*}
\mu &= \mu_0 \left( \frac{T}{T_0} \right)^{0.7}, \\
\rho D &= \frac{\mu}{Pr_{ref}},
\end{align*}

(14)

where $Pr_{ref} = 0.75$, $T_0 = 298$ K, and $\mu_0 = 1.85 \times 10^{-4}$ gm/cm·s is a reference value for air. The strain rate used in the calculation is $a = 40s^{-1}$. The boundary conditions at the fuel jet $y = -0.6$ cm are given by

$V = 2.8 \times 10^{-2}$ gm/cm²·s, \\
$f' = 1.216$, \\
$S = 1$,

and at the oxidizer jet $y = 0.8876$ cm by

$f' = 1$, \\
$S = 0$.

For the given conserved scalar, we first solve for the location of the flame front, then recover temperature and species, which are used to evaluate viscosity and density by the ideal gas law. More details such as the mass fraction of species (CH₄, O₂, N₂, ...) can be found in [23]. Figure 3 shows the “double peak” normal velocity profile and temperature profile predicted by the flame sheet model.
Without a good initial guess, it is very hard to solve directly the system obtained by discretizing equations (11), (12), and (13). As a consequence, we instead employ a time-dependent approach and solve the pseudotransient problem

\begin{equation}
D \frac{x_k - x_{k-1}}{\Delta t_k} + F(x_k) = 0, \quad k = 0, 1, \ldots,
\end{equation}

which brings the approximate solution into steady-state domain of convergence, and then resume steady-state iterations (when $\Delta t_k > 10$). Here $D$ is a diagonal matrix with entries 0 or 1, and no pseudotime-stepping is used on the boundaries or for the algebraic constraint equation [11].

We discretize systems at 500 points and only report some results for the steady-state problem after successive time steps. For this example, NE methods are carried out in a field-split manner by eliminating the conserved scalar $S$. The initial guess is set as follows: $f'$ and $S$ are the linear interpolation using boundary values and $V$ is the constant for the whole domain.

Figure 4 compares a global inexact Newton method with backtracking (INB) against the nonlinear elimination preconditioned inexact Newton algorithm (INB-NE) and adaptive INB-NE version (INB-NE$^\text{adapt}$). In the first step, the right preconditioning results in an increase in the steady-state residual, but this hill-climbing allows the Newton method to find the domain of convergence more quickly. Compared with INB, the INB-NE and INB-NE$^\text{adapt}$ methods converge within five steps and dramatically reduce the number of nonlinear iterations. Table 2 monitors the history of $\rho_k$ and $\eta_k$ and shows that the right nonlinear preconditioner (INB-NE) is required for only the first two steps.

4.2. Two-phase flow in porous media. We consider the immiscible displacement of incompressible two-phase (water-oil) flow in a two-dimensional (2D) porous medium in the absence of capillary forces and gravity, which is described by the saturation equation and Darcy's law.

\begin{align}
\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot u_\alpha &= q_\alpha, \quad \alpha = w, o, \\
 u_\alpha &= -\lambda_\alpha K \nabla p, \quad \alpha = w, o,
\end{align}
ADAPTIVE NONLINEAR PRECONDITIONING TECHNIQUES

Fig. 4. Nonlinear residual history for 1D flame sheet model discretized at 500 points. We set the parameters $\epsilon_{\text{global-linear-rtol}} = 10^{-6}$, $\epsilon_{\text{global-linear-atol}} = 10^{-10}$, $\epsilon_{\text{global-linear-rtol}} = 10^{-5}$, and $\epsilon_{\text{global-nonlinear-rtol}} = 10^{-4}$ is the stopping condition for subproblems solved in subspace correction.

Table 2
The history of $\eta_k$ and $\rho_k$ in the adaptive INB-NE algorithm for 1D flame sheet model.

<table>
<thead>
<tr>
<th>Iter.</th>
<th>$\eta_k$</th>
<th>$\rho_k$</th>
<th>One step of INB-NE$_{\text{adapt}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>0.0</td>
<td>INB-NE</td>
</tr>
<tr>
<td>1</td>
<td>0.551347</td>
<td>0.448653</td>
<td>INB-NE</td>
</tr>
<tr>
<td>2</td>
<td>0.105162</td>
<td>0.894838</td>
<td>INB</td>
</tr>
<tr>
<td>3</td>
<td>0.013395</td>
<td>0.986589</td>
<td>INB</td>
</tr>
<tr>
<td>4</td>
<td>0.000015</td>
<td>0.999985</td>
<td>INB</td>
</tr>
</tbody>
</table>

where $p$, $S_\alpha$, $\rho_\alpha$, $u_\alpha$, and $q_\alpha$ are pressure, saturation, density, velocity, and the external volumetric flow rate of phase $\alpha$, respectively. The saturation constraint is given by

$S_w + S_o = 1.$

Here $\phi$ is the porosity and $\mathbf{K}$ is the absolute permeability tensor. We import porosity and permeability data from the 85th layer of Model 2 of the 10th SPE Comparative Solution Project (SPE10) [10], which is a classic benchmark problem for reservoir simulation. Figure 5 shows that the permeability field is characterized by variations of more than six orders of magnitude. The irreducible oil saturation $S_{or}$ and the connate water saturation $S_{wc}$ are both set to be 0.2. The relative permeability functions are given by the following simple analytical expressions:

$k_{rw}(S_w) = S_e^2$, $k_{ro}(S_w) = (1 - S_e)^2$,

where $S_e$ is the normalized saturation defined by

$S_e = \frac{S_w - S_{wc}}{1 - S_{or} - S_{wc}}.$

There is one injection well at the center and there are four production wells at the four corners. We use a rate-control (58.82 bbl/day) injection well model and bottom hole pressure-control (4000 psi) production well models in our simulation [9].
The reservoir domain is $[0, 1200 \text{ ft}] \times [0, 2200 \text{ ft}]$ divided into $60 \times 220$ cells. The initial water saturation is 0.2. The standard two-point flux-approximation finite volume method is applied to discretize the model equations for the spatial terms, and the implicit first-order scheme (backward Euler) is used for the time differencing approximation. The unknowns $p$ and $S_w$ are approximated by cellwise constants. The permeability and mobilities are not well-defined at the interface and hence are approximated separately by harmonic average values and standard upstream weighting. The upwind direction for each phase $\alpha$ is determined using its phase velocity $u_\alpha$.

Large Jacobian systems are typically solved by some Krylov subspace method, here GMRES [32]. Numerical tests are carried out on $60 \times 220$ mesh using eight processors in parallel. The ASPIN method is implemented with the partition of $2 \times 4$ subdomains and the overlap of four cells. The INB method is implemented with right overlapping additive Schwarz preconditioners that use the same partition as the ASPIN. The time step is halved if the global nonlinear solver fails and is doubled if two successive time steps are not reduced. The initial time step is 1 day, and the maximum time step is 20 days. Table 3 gives the history of $\rho_k$ and $\eta_k$ in the adaptive ASPIN method (ASPIN$^{\text{adapt}}$) for the initial time step and compares the performance of INB, ASPIN, and ASPIN$^{\text{adapt}}$ up to 270 days. For the initial time step, the left nonlinear preconditioner (ASPIN) is implemented only for the first two outer iterations and it returns adaptively again for the fourth. Table 3 shows that ASPIN and ASPIN$^{\text{adapt}}$ greatly reduce the number of nonlinear iterations compared with INB but require more GMRES iterations on average per nonlinear iteration. The adaptive ASPIN method requires 76 nonlinear iterations, including 49 INB iterations and 27 ASPIN iterations, and requires the least execution time.

4.3. Lid-driven cavity flow. We consider the 2D lid-driven cavity flow problem [5, 6] in the velocity-vorticity formulation with three unknowns: velocity components $u$, $v$ and vorticity $\omega$.

$$
\begin{align*}
-\Delta u - \frac{\partial \omega}{\partial y} &= 0, \\
-\Delta v + \frac{\partial \omega}{\partial x} &= 0, \\
-\frac{1}{Re} \Delta \omega + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} &= 0.
\end{align*}
$$
The problem is defined on a square cavity $\Omega = (0,1) \times (0,1)$ consisting of a lid moving with a tangential unit velocity and three rigid walls with no-slip conditions, i.e., $u = 1, v = 0$ on the top boundary and $u = 0, v = 0$ on the other boundaries. The boundary condition on $\omega$ is given by its definition:

\begin{equation}
\omega(x, y) = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}.
\end{equation}

We partition both the unknowns and the nonlinear system into two groups to apply the field-based nonlinear preconditioners. Taking as one the partition of the velocity degrees of freedom and the other the vorticity degrees of freedom, we split the system (21) into two submodels:

\begin{equation}
\hat{F}_1 : \begin{cases}
-\Delta u - \frac{\partial \omega}{\partial y} = 0, \\
-\Delta v + \frac{\partial \omega}{\partial x} = 0,
\end{cases}
\end{equation}

and

\begin{equation}
\hat{F}_2 : \quad -\frac{1}{Re} \Delta \omega + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = 0.
\end{equation}

We discretize the PDEs shown in (21) using a finite difference scheme with the standard five-point stencil. Upwinding is employed for the convection terms in the vorticity equation and the vorticity boundary condition is differenced inward with respect to the normal direction. (It is noted that this is only a first-order discretization but our emphasis is on the robustness of the nonlinear algebraic solution.) We first solve velocity equations for $u$ and $v$ and then solve vorticity equations for $\omega$ in the multiplicative nonlinear preconditioners.

Here, the subproblems obtained from the discretization of (23) and (24) are linear, which can be regarded as a profound benefit of field splitting for this example. In our tests, all subproblems and linear problems involved in the nonlinearly preconditioned Jacobian application are solved by GMRES with BoomerAMG preconditioning [18].
Table 4

The number of nonlinear iterations using INB, MSPIN, and MSPIN\textsuperscript{adapt}, and the history of $\eta_k$ and $\rho_k$ for the driven cavity flow problem on 128 × 128 mesh with different Reynolds numbers using MSPIN\textsuperscript{adapt}. We set the parameters $\epsilon_{\text{global-linear-rtol}} = 10^{-6}$, $\epsilon_{\text{global-linear-atol}} = 10^{-10}$, $\epsilon_{\text{global-linear-rtol}} = 10^{-6}$, and $\epsilon_{\text{Jacc-linear-rtol}} = 10^{-3}$ denote the relative tolerance for linear subproblems and linear problems involved in the nonlinearly preconditioned Jacobian application, respectively.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$Re = 100$</th>
<th>$Re = 1000$</th>
<th>$Re = 5000$</th>
<th>$Re = 10000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>INB</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MSPIN</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>MSPIN\textsuperscript{adapt}</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Summary for MSPIN\textsuperscript{adapt} at different $Re$

<table>
<thead>
<tr>
<th>$Re = 100$</th>
<th>Iter.</th>
<th>$\eta_k$</th>
<th>$\rho_k$</th>
<th>One step of MSPIN\textsuperscript{adapt}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1.0</td>
<td>0.0</td>
<td>MSPIN</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.580156</td>
<td>0.419454</td>
<td>MSPIN</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.132696</td>
<td>0.867245</td>
<td>INB</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.028887</td>
<td>0.971112</td>
<td>INB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Re = 1000$</th>
<th>Iter.</th>
<th>$\eta_k$</th>
<th>$\rho_k$</th>
<th>One step of MSPIN\textsuperscript{adapt}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1.0</td>
<td>0.0</td>
<td>MSPIN</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.2727271</td>
<td>0.727271</td>
<td>MSPIN</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.025135</td>
<td>0.974858</td>
<td>INB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Re = 10000$</th>
<th>Iter.</th>
<th>$\eta_k$</th>
<th>$\rho_k$</th>
<th>One step of MSPIN\textsuperscript{adapt}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1.0</td>
<td>0.0</td>
<td>MSPIN</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.061412</td>
<td>0.938557</td>
<td>INB</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.034303</td>
<td>0.965695</td>
<td>INB</td>
</tr>
</tbody>
</table>

We set the tolerance parameters as $\epsilon_{\text{global-linear-rtol}} = 10^{-10}$, $\epsilon_{\text{global-linear-atol}} = 10^{-6}$, $\epsilon_{\text{sub-linear-rtol}} = 10^{-3}$, and $\epsilon_{\text{Jacc-linear-rtol}} = 10^{-3}$. The Jacobian matrices are formed using a finite difference scheme. We run the test for the uniform meshes 128 × 128 using four processors.

In Table 4, we compare the number of nonlinear iterations using MSPIN and the adaptive MSPIN method (MSPIN\textsuperscript{adapt}) and monitor the history of $\eta_k$ and $\rho_k$ involved in MSPIN\textsuperscript{adapt} for the driven cavity flow problem at different Reynolds numbers. It shows that MSPIN\textsuperscript{adapt} requires nonlinear preconditioning techniques for the first few steps, but it converges within the same number of nonlinear iterations as MSPIN.

4.4. Summary. In this paper, we introduce an adaptive nonlinear preconditioning framework based on convergence monitors. Nonlinear preconditioning techniques usually reduce the number of outer Newton iterations by involving the additional computational cost of solving subproblems in inner iterations and the communication of global assembly in outer iterations. Compared with the traditional nonlinear preconditioners, the new adaptive algorithm has similar robustness and a fast convergence rate but is more flexible since nonlinear preconditioning can be turned on and off during some of the outer Newton iterations. This provides an algorithmic tuning knob that may be especially useful for large-scale simulations, in which polyalgorithms can help to extract performance.

Acknowledgment. The authors are grateful for the use of a Cray XC40 (“Shah-een”) operated by the Supercomputing Laboratory of the King Abdullah University of Science and Technology.
REFERENCES


