SIAM J. SCI. COMPUT.
Vol. 37, No. 3, pp. A1388–A1409

FIELD-SPLIT PRECONDITIONED INEXACT NEWTON ALGORITHMS*

LULU LIU† AND DAVID E. KEYES†

Abstract. The multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm is presented as a complement to additive Schwarz preconditioned inexact Newton (ASPIN). At an algebraic level, ASPIN and MSPIN are variants of the same strategy to improve the convergence of systems with unbalanced nonlinearities; however, they have natural complementarity in practice. MSPIN is naturally based on partitioning of degrees of freedom in a nonlinear PDE system by field type rather than by subdomain, where a modest factor of concurrency can be sacrificed for physically motivated convergence robustness. ASPIN, originally introduced for decompositions into subdomains, is natural for high concurrency and reduction of global synchronization. We consider both types of inexact Newton algorithms in the field-split context, and we augment the classical convergence theory of ASPIN for the multiplicative case. Numerical experiments show that MSPIN can be significantly more robust than Newton methods based on global linearizations, and that MSPIN can be more robust than ASPIN and maintain fast convergence even for challenging problems, such as high Reynolds number Navier–Stokes equations.

Key words. nonlinear equations, nonlinear preconditioning, field splitting, Newton method, Navier–Stokes equations

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

DOI. 10.1137/140970379

1. Introduction.

Newton-like methods in their many variants are often favored for the solution of nonlinear systems, especially large algebraic systems arising from discretized differential equations. However, the problem of "nonlinear stiffness" frequently arises, in which progress in updating the state variables with a global Newton step is retarded by an often small subset of the variable-equation pairs. Global Newton-like methods may waste considerable computational resources while the majority of state variables barely evolve for many iterations until some critical feature (e.g., boundary or interior layer, aerodynamic shock, reaction zone, contact discontinuity, phase transition) of the solution falls into place, following which the desired superlinear asymptotic convergence of Newton polishes the root. These unproductive iterations typically require the solution of large linear systems with ill-conditioned Jacobians. Worse, Newton may stagnate indefinitely.

To conquer unbalanced nonlinearities and improve global convergence properties, the additive Schwarz preconditioned inexact Newton (ASPIN) was devised in [5] as an inner-outer Newton solver, with most of the work performed on independent subproblems in inner iterations, plus relatively few outer iterations on a transformed system, on which Newton is supposed to converge quickly. The key idea of ASPIN is to transform the original system into a modified system with the same root, and to solve it using a Jacobian-free [19] inexact Newton method.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 27, 2014; accepted for publication (in revised form) March 2, 2015; published electronically June 2, 2015. This work was supported by KAUST's baseline funding and by the Extreme Computing Research Center.

http://www.siam.org/journals/sisc/37-3/97037.html

†Program in Applied Mathematics and Computational Science, and Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi Arabia (lulu.liu@kaust.edu.sa, david.keyes@kaust.edu.sa).

ASPIN has virtues beyond potentially more robust nonlinear convergence:

- The inner nonlinear subproblems possess smaller working sets and, being less nonlinearly stiff, may require in aggregate less work than the outer iterations they replace, resulting in a net reduction of execution time.
- The auxiliary inner iteration linear problems are smaller than the global system, so the sets of processors that need to synchronize frequently (e.g., because of inner products or recurrences) to carry out their solution are smaller. Furthermore, these sets may be asynchronous relative to each other.
- Since the outer iteration is Jacobian-free and the inner iterations work on subsets of the original system, ASPIN can be implemented with modest additions to an existing Newton code, and it is available in the popular PETSc framework [2].

Various enhancements of ASPIN have been proposed, including two-level versions [6, 17, 18, 22] to improve the conditioning of the outer Jacobian-free linear problem, and the incorporation of a rescaling step [16]. For a taxonomy of nonlinear preconditioning methods placing ASPIN in a broader context of scalable nonlinear solvers, see [3].

The classical ASPIN was presented in [5] as a nonlinear domain decomposition method. Its algebraic generalization to other types of decompositions was anticipated, and here we propose an algebraic variant of ASPIN based on field splitting, which typically generates subproblems involving nonintersecting subsets of the original unknowns, whereas the classical ASPIN in [5] typically generates subproblems whose unknowns overlap as induced by subdomain geometry. Under field splitting, a Gauss–Seidel-like sequential ordering of the subproblems is often natural, whereas domain decomposition naturally generates concurrent problems of Jacobi character. The Gauss–Seidel-like variant of ASPIN is referred to as the multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm. The field-split preconditioned inexact Newton method generates a number of subproblems proportional to the number of identifiable strongly coupled physical phenomena. In contrast, the granularity of subproblems is arbitrarily large in the classical ASPIN in [5], limited only by resolution considerations. In multiphysics problems of large scale, the classical ASPIN may naturally nest inside the field-split preconditioned inexact Newton methods, or vice versa.

We briefly review the classical inexact Newton algorithm with the backtracking technique (INB) [8, 10, 24], which serves as the basic block of both ASPIN and MSPIN. The INB method is used both as a nonlinear solver for the transformed global system and for subsystems in the original coordinates. For the discrete nonlinear function $F : R^n \to R^n$, we want to find a vector $x^* \in R^n$ such that

$$F(x^*) = 0, \tag{1.1}$$

where $F = [F_1, F_2, \ldots, F_n]^T$ and $x = [x_1, x_2, \ldots, x_n]^T$. The framework of INB is described as follows.

ALGORITHM 1. INB.

An initial iterate $x^{(0)}$ is given. For $k = 0, 1, 2, \ldots$ until convergence:

1. Find an inexact Newton direction $d^{(k)}$ such that

$$\|F(x^{(k)}) - F'(x^{(k)})d^{(k)}\| \le \eta_k \|F(x^{(k)})\|. \tag{1.2}$$

2. Determine a step size $\lambda^{(k)}$ using a backtracking linesearch technique based on the function $f(x) = \frac{1}{2}\|F(x)\|^2$.

3. Compute a new approximate solution

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} d^{(k)}. \tag{1.3}$$

Here η_k, called a "forcing term," determines how accurately the Jacobian linear system needs to be solved using some iterative method, such as GMRES [25]. Some choices of η_k based on norms or updates and residuals available as by-products of the main computation were suggested by Eisenstat and Walker [11].

The rest of the paper is organized as follows. In section 2, we introduce two types of the nonlinear field-split preconditioned inexact Newton algorithm and derive formulae for the preconditioned functions and the corresponding Jacobian matrices. In section 3, we present the details of the field-split preconditioned inexact Newton algorithm implementation as well as an illustrative example of two scalar components. Section 4 contains a proof of the equivalence of the original nonlinear system and the nonlinearly preconditioned system under some reasonable conditions for applications, the $N > 2$ extension of which is left for the appendix. Some numerical results are given in section 5. We conclude with comments on the niche for the field-split preconditioned inexact Newton algorithm variants.

2. Field-split preconditioned inexact Newton algorithm. Consider a nonlinear root-finding problem (1.1). In order to demonstrate the field-split preconditioned inexact Newton framework, the nonlinear function $F(x)$ is split conformally into two nonoverlapping components representing different physical aspects as

$$F(x) = F(u, v) = \begin{bmatrix} G(u, v) \\ H(u, v) \end{bmatrix} = 0, \quad x = [u, v]^T. \tag{2.1}$$

As with the domain-based ASPIN in [5], the field-split preconditioned inexact Newton algorithm solves a preconditioned problem,

$$\mathcal{F}(x) = \mathcal{F}(u, v) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} = 0, \tag{2.2}$$

which has the same solution with (2.1), where $g(u, v)$ and $h(u, v)$ are defined as the physical variable corrections of any given $x = [u, v]^T$, and the INB algorithm is used as the nonlinear solver for the global preconditioned system, as well as the two submodels. In this section, we focus on the form of nonlinearly preconditioned function $\mathcal{F}(x)$ and the Jacobian calculation corresponding to the field-split preconditioned inexact Newton algorithms. We introduce two types of field-split preconditioned inexact Newton algorithms: ASPIN and MSPIN.

2.1. ASPIN. We describe first the construction of the Jacobi-type nonlinear preconditioner. In the ASPIN algorithm, the submodels are solved independently for the physical variable corrections, and these corrections form the preconditioned system.

For any given $x = [u, v]^T \in R^n$, define $g = g(u, v)$ and $h = h(u, v)$ as the solutions of the submodels in the original nonlinear system (2.1),

$$G(u - g, v) = 0, \tag{2.3}$$

$$H(u, v - h) = 0. \tag{2.4}$$

The Jacobi-type nonlinearly preconditioned function $\mathcal{F}_J$ is then defined as

$$\mathcal{F}_J(x) = \begin{bmatrix} g(u,v) \\ h(u,v) \end{bmatrix}, \quad x = [u,v]^T. \tag{2.5}$$

The Jacobian of $\mathcal{F}_J(x)$ is not as explicitly available as $F'(x)$ due to the implicit definition of $\mathcal{F}_J(x)$ via (2.3) and (2.4); therefore, it is necessary to derive a formula for the application of the Jacobian matrix corresponding to the preconditioned function $\mathcal{F}_J$ in order to apply Newton's method or its variations. We next describe an approximate, readily computable Jacobian form of $\mathcal{F}_J(x)$.

Let $p = u - g(u,v)$ and $q = v - h(u,v)$. Taking the derivative of (2.3) with respect to u, we have

$$\frac{\partial G}{\partial p}\left(I_u - \frac{\partial g}{\partial u}\right) = 0, \tag{2.6}$$

where I_u is the identity matrix that has the same dimension as the u block. Assuming $\frac{\partial G}{\partial p}$ is nonsingular, (2.6) implies

$$\frac{\partial g}{\partial u} = I_u. \tag{2.7}$$

Next, we take the derivative of (2.3) with respect to v, yielding

$$\frac{\partial G}{\partial p}\left(-\frac{\partial g}{\partial v}\right) + \frac{\partial G}{\partial v} = 0, \tag{2.8}$$

which is equivalent to

$$\frac{\partial g}{\partial v} = \left(\frac{\partial G}{\partial p}\right)^{-1} \frac{\partial G}{\partial v}. \tag{2.9}$$

Similarly, taking the derivatives of (2.4) with respect to u and v, we obtain

$$\frac{\partial h}{\partial u} = \left(\frac{\partial H}{\partial q}\right)^{-1} \frac{\partial H}{\partial u} \tag{2.10}$$

and

$$\frac{\partial h}{\partial v} = I_v, \tag{2.11}$$

where I_v is again the identity matrix of appropriate size. From expressions (2.7), (2.9), (2.10), and (2.11), we can write the Jacobian of $\mathcal{F}_J(x)$ as follows:

$$\mathcal{J}(u,v) = \begin{bmatrix} g_u & g_v \\ h_u & h_v \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial G}{\partial p}\right)^{-1} & \\ & \left(\frac{\partial H}{\partial q}\right)^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial u} & \frac{\partial H}{\partial q} \end{bmatrix}. \tag{2.12}$$

Due to the continuity of $F(x)$, we know that $g(u,v) \to 0$ and $h(u,v) \to 0$, i.e., $p \to u$, $q \to v$, when $x = [u,v]^T$ is sufficiently close to the exact solution. In practice, it is more convenient to use the following approximate Jacobian:

$$\hat{\mathcal{J}}(u,v) = \begin{bmatrix} G_u^{-1} & \\ & H_v^{-1} \end{bmatrix} \begin{bmatrix} G_u & G_v \\ H_u & H_v \end{bmatrix} = \begin{bmatrix} G_u & \\ & H_v \end{bmatrix}^{-1} J(u,v). \tag{2.13}$$

The approximate Jacobian matrix $\hat{\mathcal{J}}$ will generally be dense and expensive to form explicitly. However, only the multiplication by $\hat{\mathcal{J}}$ with a given vector x, $y = \hat{\mathcal{J}}x$, is required for Krylov subspace methods when we solve the Jacobian system. In our implementation, the matrix-vector multiplication $\hat{\mathcal{J}}x = y$ is carried out as follows:

1. Perform the multiplication $w = Jx$, $w = [w_1, w_2]^T$.
2. Solve $G_u y_1 = w_1$ and $H_v y_2 = w_2$.
3. Form the result $y = [y_1, y_2]^T$.

Remark 2.1. In the linear case, this algorithm is the same as physics-based block Jacobi linear preconditioning. If we have the linear system

$$F(x) = Ax - b, \tag{2.14}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{2.15}$$

then the preconditioned system has the form of

$$\mathcal{F}_J(x) = M^{-1}(Ax - b), \qquad M = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}. \tag{2.16}$$

2.2. MSPIN. In distinction from the ASPIN algorithm, the submodels are solved sequentially for the physical variable corrections in the MSPIN algorithm, and the preconditioned system again consists of these corrections. We describe the details of the construction of the Gauss–Seidel-like nonlinear preconditioner.

For any given $x = [u, v]^T \in R^n$, the preconditioned function

$$\mathcal{F}_{GS}(x) = \begin{bmatrix} g(u, v) \\ h(u, v) \end{bmatrix} \tag{2.17}$$

is obtained by solving

$$G(u - g, v) = 0 \tag{2.18}$$

for g. With the values of u, v, g, the system

$$H(u - g, v - h) = 0 \tag{2.19}$$

is solved for h. Since the preconditioned function $\mathcal{F}_{GS}(x)$ is implicitly defined via (2.18) and (2.19), the calculation of the corresponding Jacobian is not straightforward. Below we discuss a computable form of the inverse action.

As shown in section 2.1, the derivatives of $g(u, v)$ with respect to u and v are

$$\frac{\partial g}{\partial u} = I_u \tag{2.20}$$

and

$$\frac{\partial g}{\partial v} = \left(\frac{\partial G}{\partial p}\right)^{-1} \frac{\partial G}{\partial v}, \tag{2.21}$$

respectively. Taking the derivative of (2.19) with respect to u,

$$\frac{\partial H}{\partial p}\left(I_u - \frac{\partial g}{\partial u}\right) + \frac{\partial H}{\partial q}\left(-\frac{\partial h}{\partial u}\right) = 0. \tag{2.22}$$

Assuming $\frac{\partial H}{\partial q}$ is nonsingular, (2.20) and (2.22) imply

$$\frac{\partial h}{\partial u} = 0. \tag{2.23}$$

Taking the derivative of (2.19) with respect to v,

$$\frac{\partial H}{\partial p}\left(-\frac{\partial g}{\partial v}\right) + \frac{\partial H}{\partial q}\left(I_v - \frac{\partial h}{\partial v}\right) = 0, \tag{2.24}$$

which is equivalent to

$$\frac{\partial h}{\partial v} = I_v - \left(\frac{\partial H}{\partial q}\right)^{-1}\frac{\partial H}{\partial p}\frac{\partial g}{\partial v}. \tag{2.25}$$

Substituting (2.21) into (2.25), we have

$$\frac{\partial h}{\partial v} = \left(\frac{\partial H}{\partial q}\right)^{-1}\left(\frac{\partial H}{\partial q} - \frac{\partial H}{\partial p}\left(\frac{\partial G}{\partial p}\right)^{-1}\frac{\partial G}{\partial v}\right). \tag{2.26}$$

Integrating (2.20), (2.21), (2.23), and (2.26) into the Jacobian of $\mathcal{F}_{GS}$, we get

$$\mathcal{J} = \begin{bmatrix} g_u & g_v \\ h_u & h_v \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial G}{\partial p}\right)^{-1} & \\ -\left(\frac{\partial H}{\partial q}\right)^{-1}\frac{\partial H}{\partial p}\left(\frac{\partial G}{\partial p}\right)^{-1} & \left(\frac{\partial H}{\partial q}\right)^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}, \tag{2.27}$$

or, more concisely,

$$\mathcal{J}(u,v) = \begin{bmatrix} \frac{\partial G}{\partial p} & \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial G}{\partial p} & \frac{\partial G}{\partial v} \\ \frac{\partial H}{\partial p} & \frac{\partial H}{\partial q} \end{bmatrix}. \tag{2.28}$$

In our implementation, it is more convenient to use the following approximate Jacobian:

$$\hat{\mathcal{J}}(u,v) = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} \begin{bmatrix} G_p & G_v \\ H_p & H_v \end{bmatrix} = \begin{bmatrix} G_p & \\ H_p & H_v \end{bmatrix}^{-1} J(p,v). \tag{2.29}$$

Similarly to ASPIN, Krylov subspace methods such as GMRES are used to solve the Jacobian system, which require the multiplication of $\hat{\mathcal{J}}$ with a given vector x, $\hat{\mathcal{J}}x = y$, instead of any explicit calculation and storage of $\hat{\mathcal{J}}$.

Remark 2.2. If the matrices A_{11} and A_{22} are nonsingular, then we have

$$\begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ 0 & A_{22}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{21} & I \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & I \end{bmatrix}. \tag{2.30}$$

With Remark 2.2, the matrix-vector multiplication $\hat{\mathcal{J}}x = y$ is carried out as follows:

1. Perform the multiplication $w = Jx$, $w = [w_1, w_2]^T$.
2. Let $z_2 = w_2$ and solve $G_p z_1 = w_1$.
3. Perform $z_2 = z_2 - H_p z_1$ and set $y_1 = z_1$.
4. Solve $H_v y_2 = z_2$.
5. Form the result $y = [y_1, y_2]^T$.

Remark 2.3. In the linear case, this algorithm is the same as physics-based block Gauss–Seidel linear preconditioning. If we have the linear system

$$F(x) = Ax - b, \tag{2.31}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{2.32}$$

then

$$\mathcal{F}_{GS}(x) = M^{-1}(Ax - b), \qquad M = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}. \tag{2.33}$$

3. Implementation of the field-split ASPIN and MSPIN. In this section, we describe a complete field-split preconditioned inexact Newton algorithm and implementation details. We also present an illustrative example.

3.1. The field-split ASPIN and MSPIN. Splitting $F(x)$ into $G(u, v)$ and $H(u, v)$, $x = [u, v]^T$. We set $x^{(0)} = [u^{(0)}, v^{(0)}]^T$ as the initial guess. The current approximate solution is denoted by $x^{(k)}$ and $x^{(k+1)}$ is the new approximate solution obtained through the following algorithm.

ALGORITHM 2.

1. Compute the nonlinear residual $\mathcal{F}(x^{(k)})$ by solving the submodels.
   (a) Starting from $g_0^{(k)} = 0$ and $h_0^{(k)} = 0$, find $g^{(k)}$ and $h^{(k)}$ by solving subproblems

ASPIN	MSPIN
Solve for $g_i^{(k)}$ and $h_j^{(k)}$ simultaneously	Solve for $g_i^{(k)}$
(3.1) $G(u^{(k)} - g_i^{(k)}, v^{(k)}) = 0,$	(3.3) $G(u^{(k)} - g_i^{(k)}, v^{(k)}) = 0,$
	Solve for $h_j^{(k)}$ with the new $g_i^{(k)}$
(3.2) $H(u^{(k)}, v^{(k)} - h_j^{(k)}) = 0,$	(3.4) $H(u^{(k)} - g_i^{(k)}, v^{(k)} - h_j^{(k)}) = 0,$

   where $i = 0, 1, \ldots$ until

$$\|G(u^{(k)} - g_i^{(k)}, v^{(k)})\| \leqslant \epsilon_{sub-nonlinear-rtol} \|G(u^{(k)} - g_0^{(k)}, v^{(k)})\|,$$

   and $j = 0, 1, \ldots$ until

$$\|H(u^{(k)}, v^{(k)} - h_j^{(k)})\| \leqslant \epsilon_{sub-nonlinear-rtol} \|H(u^{(k)}, v^{(k)} - h_0^{(k)})\|,$$

   or

$$\|H(u^{(k)} - g^{(k)}, v^{(k)} - h_j^{(k)})\| \leqslant \epsilon_{sub-nonlinear-rtol} \|H(u^{(k)} - g^{(k)}, v^{(k)} - h_0^{(k)})\|,$$

   where $g^{(k)}$ is the solution of (3.3).

(b) Form the global residual,

$$\mathcal{F}(x^{(k)}) = \begin{bmatrix} g^{(k)} \\ h^{(k)} \end{bmatrix}. \tag{3.5}$$

(c) Check the stopping conditions on $\mathcal{F}(x^{(k)})$,

$$\|\mathcal{F}(x^{(k)})\| \leqslant \epsilon_{global-nonlinear-rtol}\|\mathcal{F}(x^{(0)})\|. \tag{3.6}$$

2. The initial guess is zero for $d^{(k)}$. Find the inexact Newton direction $d^{(k)}$ by approximately solving

$$\hat{\mathcal{J}}d^{(k)} = \mathcal{F}(x^{(k)}), \tag{3.7}$$

such that

$$\left\|\mathcal{F}(x^{(k)}) - \hat{\mathcal{J}}d^{(k)}\right\| \leq \epsilon_{global-linear-rtol}\|\mathcal{F}(x^{(k)})\|, \tag{3.8}$$

where $\hat{\mathcal{J}}$ has the form (2.13) or (2.29) corresponding to ASPIN and MSPIN, respectively.

3. Compute the new approximate solution

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)}d^{(k)},$$

where the step length $\lambda^{(k)}$ is determined by performing linesearch along $d^{(k)}$.

In Algorithm 2, the determination of the partition of the physical variables can be the most interesting part of implementation, because the best choice is generally problem-specific. Once the partition and the initial guess are given, we move on to step 1.

In step 1(a), we solve nonlinear subproblems (3.1), (3.2), (3.3), and (3.4) using the INB method. It is noted that we need the solution $g^{(k)}$ of the submodel (3.3) before we solve the second submodel (3.4).

In step 2, no preconditioning is employed for Krylov subspace iterative methods such as GMRES when we solve the global Jacobian system (3.7) using the INB framework. In fact, nonlinear preconditioning automatically offers a linear preconditioner for the original Jacobian system. The Jacobian formulae (2.13) and (2.29) correspond to the block Jacobi preconditioning and the block Gauss–Seidel preconditioning for the original unpreconditioned equation, respectively.

3.2. A simple example. Following [15], we present a simple example that demonstrates how the field-split preconditioned inexact Newton algorithm rounds out a complex contour landscape for the function whose root is sought.

Consider the system $F(x)$ of two nonlinear equations in two unknowns,

$$F_1(x_1, x_2) = (x_1 - x_2^3 + 1)^3 - x_2^3, \tag{3.9}$$

$$F_2(x_1, x_2) = 2x_1 + 3x_2 - 5. \tag{3.10}$$

It is easy to verify that $x^* = [1, 1]^T$ is a root of this system.

By setting

$$F_1(x_1 - \delta_1^J, x_2) = 0, \tag{3.11}$$

$$F_2(x_1, x_2 - \delta_2^J) = 0, \tag{3.12}$$

TABLE 1
The number of nonlinear iterations. The outer global tolerance is 10^{-8}, and the inner component tolerances are both 10^{-3}.

Initial guess x_0	INB	ASPIN	MSPIN
$x_0 = (0,0)^T$	11	7	6
$x_0 = (0,2)^T$	10	7	5
$x_0 = (2,0)^T$	1	8	6
$x_0 = (2,2)^T$	11	7	5

we can solve explicitly for $\delta_i^J(x_1, x_2)$, $i = 1, 2$, because of the algebraic simplicity of the system, and derive a nonlinearly preconditioned system $\mathcal{F}_J(x)$ corresponding to ASPIN as follows:

$$\delta_1^J(x_1, x_2) = x_1 - x_2^3 + 1 - x_2, \tag{3.13}$$

$$\delta_2^J(x_1, x_2) = \frac{2}{3}x_1 + x_2 - \frac{5}{3}. \tag{3.14}$$

Similarly, we derive a nonlinearly preconditioned system $\mathcal{F}_{GS}(x)$ corresponding to MSPIN,

$$\delta_1^{GS}(x_1, x_2) = x_1 - x_2^3 + 1 - x_2, \tag{3.15}$$

$$\delta_2^{GS}(x_1, x_2) = \frac{2}{3}x_2^3 + \frac{5}{3}x_2 - \frac{7}{3}, \tag{3.16}$$

by solving

$$F_1(x_1 - \delta_1^{GS}, x_2) = 0, \tag{3.17}$$

$$F_2(x_1 - \delta_1^*, x_2 - \delta_2^{GS}) = 0, \tag{3.18}$$

where δ_1^* is the solution of (3.17). Equations (3.13)–(3.14) are the components of $\mathcal{F}_J$ and (3.15)–(3.16) are the components of $\mathcal{F}_{GS}$ in the rootfinding problems $\mathcal{F}_J = 0$ and $\mathcal{F}_{GS} = 0$, respectively. In a practical implementation of the field-split preconditioned inexact Newton algorithm, we usually cannot write the preconditioned system explicitly, but we solve a Jacobian-free linear system (3.7) for the Newton correction and evaluate the function via submodel solvers.

The three columns of Table 1 show, respectively, the number of Newton iterations for solving $F(x) = 0$ using inexact Newton methods with the cubic backtracking technique (INB), for solving $\mathcal{F}_J = 0$, and for solving $\mathcal{F}_{GS} = 0$, respectively, starting from the four different initial guesses: $x_0 = (0,0)^T$, $x_0 = (0,2)^T$, $x_0 = (2,0)^T$, $x_0 = (2,2)^T$. Compared with ASPIN and MSPIN, the number of nonlinear iterations using the INB method is more sensitive to the initial estimates. The respective convergence histories are interpreted in light of the contours distribution of nonlinear systems that the three methods solve, respectively. From Figures 1 through 3, it is seen that the preconditioned systems are better balanced, in that the contours are more elliptical.

4. Some analysis of the field-split ASPIN and MSPIN. In this section, under some reasonable assumptions, we show that the preconditioned system (2.2) based on physical variable partition has the same solution as the original system (2.1).

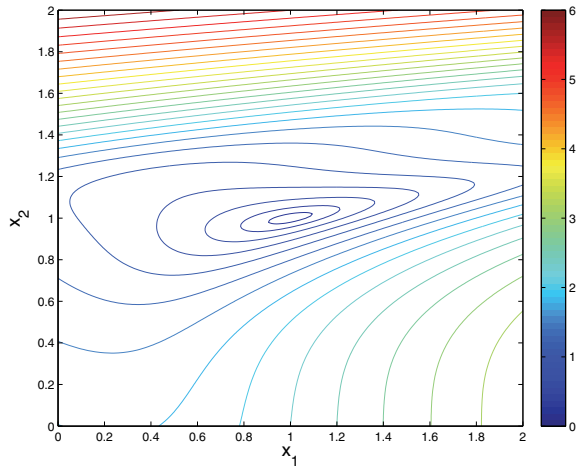

FIG. 1. *Contours of* $\log(\|F(x)\| + 1)$.

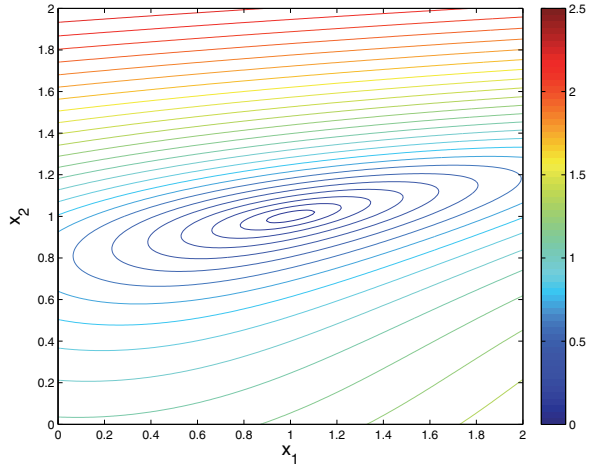

FIG. 2. *Contours of* $\log(\|\mathcal{F}_J(x)\| + 1)$.

Considering the nonlinear problem (2.1), let $D \subset R^n$ be a neighborhood of the exact solution x^*, and we make the following assumptions for $F(x)$.

Assumption 4.1. The function $F(x)$ is well-defined in D, and the Jacobian

$$F'(x) = \begin{bmatrix} G_u & G_v \\ H_u & H_v \end{bmatrix}, \quad x = [u, v]^T, \tag{4.1}$$

is continuous in D and the matrix $F'(x^*)$ is nonsingular. In addition, G_u and H_v are invertible at the exact solution x^*.

LEMMA 4.2 (see [9]). *Assume that $F'(x)$ exists for all $x \in D$; we define a uniformly bounded linear operator $DF(x', x'')$ for all $x', x'' \in D$,*

$$DF(x', x'') = \int_0^1 F'(x' + t(x'' - x'))dt, \tag{4.2}$$

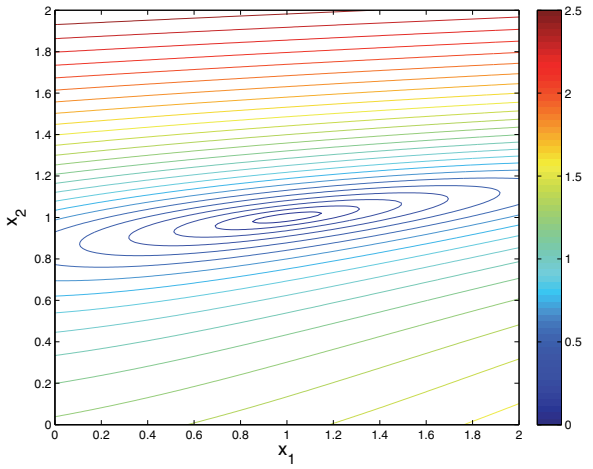

FIG. 3. *Contours of* $\log(\|\mathcal{F}_{GS}(x)\| + 1)$.

such that

$$F(x') - F(x'') = DF(x', x'')(x' - x''). \tag{4.3}$$

If $F'(x)$ *is continuous at* x^**, then*

$$\|DF(x', x'') - F'(x^*)\| \to 0, \quad x', x'' \to x^*. \tag{4.4}$$

It can be shown that these assumptions are satisfied, for instance, for the class of monotone nonlinear elliptic partial differential equations [4, 9]. Under these assumptions, the theory about the local solvability of subproblems [9] can be applied for the cases (ASPIN or MSPIN) based on physics-based partition. Here we briefly provide the proof for our special cases.

LEMMA 4.3. *Under Assumption* 4.1*, the nonlinear submodels* (2.3), (2.4), (2.18), and (2.19) *are all uniquely solvable in a sufficiently small neighborhood* U *of the exact solution* x^* *in* D.

Proof. We show unique solvability for the submodel (2.19) here, and other cases hold due to Theorem 1.1 in [9]. Let $x = [u, v] \in D \subset R^n$, where $u \in D_1 \subset R^{n_1}$ and $v \in D_2 \subset R^{n_2}$, $n_1 + n_2 = n$. Let

$$S = \{1, \ldots, n\} \tag{4.5}$$

be an index set, and we define $S_H = \{i_1, i_2, \ldots, i_{n_2}\} \subset S$ as

$$S_H = \{i \mid F_i \text{ belongs to the function } H, i \in S\}, \tag{4.6}$$

and a restriction matrix $R_H \in R^{n_2 \times n}$ is defined by

$$(R_H)_{k,l} = \begin{cases} 1, & l = i_k, \\ 0, & l \neq i_k, \end{cases} \tag{4.7}$$

and then $H(x) = R_H F(x)$ in (2.1). According to Assumption 4.1, $H_v(x^*)$ is invertible. We choose a sufficiently small neighborhood D' of the exact solution $x^*$and let $\hat{x} = [\hat{u}, \hat{v}]^T$. Then we describe a mapping $\phi : D_2 \times D' \to D_2$ defined by

$$\phi(h, \hat{x}) = h + H_v(x^*)^{-1} H(\hat{u} - \hat{g}, \hat{v} - h), \tag{4.8}$$

where $\hat{g}$ is the solution of

$$G(\hat{u} - g, \hat{v}) = 0. \tag{4.9}$$

It should be noted that we must have $\hat{g} \to 0$ as $\hat{x} \to x^*$ due to the continuity assumption on F and the local unique solvability of (4.9).

Next, we prove that the mapping ϕ is a contraction with respect to h.

$$\begin{aligned}\phi(h', \hat{x}) - \phi(h'', \hat{x}) &= h' - h'' + H_v(x^*)^{-1} R_H [F(\hat{u} - \hat{g}, \hat{v} - h') - F(\hat{u} - \hat{g}, \hat{v} - h'')] \\ &= [I - H_v(x^*)^{-1} R_H DF(x', x'') R_H^T](h' - h''),\end{aligned}$$

where R_H^T is the transpose of the matrix R_H defined in (4.7), $x' = [\hat{u} - \hat{g}, \hat{v} - h']^T$, and $x'' = [\hat{u} - \hat{g}, \hat{v} - h'']^T$. Under Assumption 4.1 and (4.4), we have $DF(x', x'') \to F'(x^*)$ as $h', h'' \to 0$ and $\hat{x} \to x^*$. Hence, there exists a neighborhood $U' \subset D$ of x^* and a positive real number r such that

$$\| I - H_v(x^*)^{-1} R_H DF(x', x'') R_H^T \| \leqslant \frac{1}{2} \tag{4.10}$$

holds for $\hat{x}, x', x'' \in U'$ and $h', h'' \in B_r(\mathbf{0})$, where $B_r(\mathbf{0})$ is an open ball of $\mathbf{0}$ with a radius r. From (4.8), we have $\phi(0, \hat{x}) = H_v(x^*)^{-1} H(\hat{u} - \hat{g}, \hat{v})$. Since $H(x^*) = 0$ and $\hat{g} \to 0$ as $\hat{x} \to x^*$, which implies $[\hat{u} - \hat{g}, \hat{v}]^T \to x^*$ as $\hat{x} \to x^*$, we could choose a suitable neighborhood U'' of x^* such that

$$\| \phi(0, \hat{x}) \| \leqslant \frac{1}{2} r \tag{4.11}$$

hold as long as $\hat{x} \in U''$. Let $U \subset U' \bigcap U''$ be a sufficiently small neighborhood; then, according to the local version of the well-known Banach fixed point theorem (Corollary 1.2 in [13]), there exists a unique fixed point h^* close to $\mathbf{0}$ that satisfies $h^* = \phi(h^*, \hat{x})$, which implies $H(\hat{u} - \hat{g}, \hat{v} - h^*) = 0$ in a small neighborhood U of x^*. ∎

LEMMA 4.4. *There exists a neighborhood $\hat{D} \subset D$ of x^* such that the Jacobian matrix $\mathcal{J}$ defined by* (2.12) *or* (2.28) *is nonsingular for any $x \in \hat{D}$.*

Proof. Since $F'(x)$ is continuous and $G_u(x^*)$ and $H_v(x^*)$ are invertible, Lemma 2.3.3 in [23] shows that there exists a neighborhood $D' \subset D$ of x^* such that G_u and H_v are invertible, and G_u^{-1} and H_v^{-1} are continuous in D'. Let $p = u - g(u, v)$ and $q = v - h(u, v)$, $x = [u, v]^T \in D$; we know that $g, h \to 0$ as $x \to x^*$ due to the continuity of F.

For the Jacobian matrix $\mathcal{J}$ defined by (2.12), we have

$$\lim_{x \to x^*} \mathcal{J}(u, v) = \begin{bmatrix} G_u^{-1}(x^*) & \\ & H_v^{-1}(x^*) \end{bmatrix} F'(x^*) = \mathcal{J}(x^*). \tag{4.12}$$

For the Jacobian matrix $\mathcal{J}$ defined by (2.28), we have

$$\lim_{x \to x^*} \mathcal{J}(u, v) = \begin{bmatrix} G_u(x^*) & \\ H_u(x^*) & H_v(x^*) \end{bmatrix}^{-1} F'(x^*) = \mathcal{J}(x^*). \tag{4.13}$$

Whether (4.12) or (4.13), $\mathcal{J}(x^*)$ is nonsingular. Hence, we may choose a suitable neighborhood $\hat{D} \subset D' \subset D$ of x^* such that $\mathcal{J}$ is nonsingular for any $x \in \hat{D}$. ∎

It should be pointed out that the regularity of $\mathcal{J}$ in (2.28) close to x^* has been given in [12], but it is obtained in a different way. Using Lemmas 4.3 and 4.4, we have the following theorem.

THEOREM 4.5. *Under Assumption* 4.1*, the original system* (2.1) *and the preconditioned system* (2.2) *based on physical variable partition have the same solution in a neighborhood of* x^* *in* D.

Proof. Assume that $x^* = [u^*, v^*]^T$ is the exact solution of the original system (2.1); we have

$$G(u^*, v^*) = 0, \quad H(u^*, v^*) = 0. \tag{4.14}$$

According to the definition of the physical variable correction in (2.3), (2.4), (2.18), and (2.19), we have

ASPIN	MSPIN
$G(u^* - g(u^*, v^*), v^*) = 0,$ (4.15)	$G(u^* - g(u^*, v^*), v^*) = 0,$ (4.17)
$H(u^*, v^* - h(u^*, v^*)) = 0.$ (4.16)	$H(u^* - g(u^*, v^*), v^* - h(u^*, v^*)) = 0.$ (4.18)

Due to the local unique solvability shown in Lemma 4.3 and comparing (4.14) with (4.15), (4.16), (4.17), and (4.18), whether in ASPIN or MSPIN, we have $g(u^*, v^*) = 0$ and $h(u^*, v^*) = 0$, i.e., x^* is a solution of the preconditioned system $\mathcal{F}$ in (2.2).

We next prove that x^* is also a unique solution of $\mathcal{F}$ in some neighborhood $\hat{U}$ of x^*. From Lemma 4.4 and Proposition 2.1 in [1], it is shown that $\mathcal{F}$ is a continuously differentiable function and $\mathcal{F}'$ is nonsingular in a neighborhood $U_1 \subset \hat{D}$ of x^* ($\hat{D}$ from Lemma 4.4). According to the inverse function theorem of calculus, the solution is unique in a neighborhood $U_2 \subset U_1 \subset \hat{D}$.

Hence, there exists a neighborhood $U' \subset U_2 \bigcap U$ (U from Lemma 4.3) such that F and $\mathcal{F}$ have the same solution x^* in U'. ∎

Remark 4.6. We can generalize the notation and theory to $N > 2$ components for both ASPIN and MSPIN algorithms. The extension for the ASPIN framework is straightforward. Appendix A provides the extension of the Jacobian form (2.28) of the preconditioned systems corresponding to the MSPIN framework.

5. Numerical results. In this section, we apply a two-component physics-based partitioning strategy to three model boundary value problems that are paradigmatic of its uses in practice. In the first, which involves a single unknown with different behavior in different subdomains of a priori known locations, one partition contains unknowns comprising an interior spike, and the other contains the unknowns in the regions of solution smoothness. In the second, which is a prototype for coupling in multiphysics problems, one partition contains unknowns involved in a nonlinear ODE, and the other contains the unknowns in an algebraic system, serving as coefficients in the ODE. In the third, 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.

TABLE 2

Comparison of the number of nonlinear iterations for different methods in the rightmost three columns. "No. points" indicates the number of grid points used to discretize the ODE. The points between G_{left} and G_{right} are unknowns in G.

No. points	G_{left}	G_{right}	Its. INB	Its. ASPIN	Its. MSPIN
100	48	53	5	2	1
500	236	265	5	2	2
1000	471	530	6	2	1
5000	2351	2650	5	2	2

5.1. Nonlinear boundary value problem. In this example, we consider a nonlinear boundary value problem [20] given by

$$-u'' + u^3 + (4 \times 10^8 (x - 0.5)^2 - 2 \times 10^4)u - 10^9 e^{-3(\frac{x-0.5}{0.01})^2} = 0, \ x \in (0, 1), \tag{5.1}$$

$$u(0) = 0, \qquad u(1) = 0, \tag{5.2}$$

which has a solution satisfying the boundary conditions to well within the machine-roundoff

$$u(x) = 10^3 e^{-(\frac{x-0.5}{0.01})^2}. \tag{5.3}$$

There is a large spike around $x = 0.5$. Using the usual second-order central difference approximation for u'', the discretization of this ODE on a fine uniform grid results in a system of nonlinear equations. The resulting nonlinear problem is solved using three methods: the inexact Newton backtracking method (INB), ASPIN, and MSPIN. For the latter two, the equations are split into two sets denoted by G and H, where the system G refers to the equations discretized at the points around $x = 0.5$.

We clarify the details of the field-split preconditioned inexact Newton algorithms for this example using 100 discretized equations. The system G consists of the discretized equations 48–53 with unknowns u_{48}–u_{53}, and other equations belong to the system H with unknowns u_1–u_{47} and u_{54}–u_{100}. The values for u_{47} and u_{54} from the previous iterate for H are treated as the boundary values for the system G, and the values of u_{48} and u_{53} from the previous iterate for G as the boundary values for the system H. The initial guess is zero for $u_1, u_2, \ldots, u_{100}$. Following the experience in [16], the global system is solved using the INB method with the parameters $\epsilon_{global-nonlinear-rtol} = 10^{-6}$, $\epsilon_{global-linear-rtol} = 10^{-8}$, and $\epsilon_{sub-nonlinear-rtol} = 10^{-3}$ is the stopping condition for the subproblems nonlinear iterations.

Table 2 shows the number of Newton iterations for the different methods on problems of different resolution. The nonlinearly preconditioned Newton's methods are effective in reducing the number of global nonlinear iterations.

5.2. An ODE coupled to an algebraic system in one dimension. We consider an ODE coupled to an algebraic system in one dimension:

$$-(vu_x)_x + \lambda u^2 = 1 \text{ on } (0, 1) \quad \text{subject to } u(0) = 0, \ u(1) = 1, \tag{5.4}$$

$$\exp(v - 1) + v = \frac{1}{\frac{1}{1+u} + \frac{1}{1+u_x^2}}, \tag{5.5}$$

TABLE 3

Global nonlinear and linear iterations using globalized INB, ASPIN, and MSPIN. $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, $\epsilon_{global-linear-rtol} = 10^{-8}$, $\epsilon_{sub-nonlinear-rtol} = 10^{-3}$. "$*$" *indicates that linear iterations are not available, since the nonlinear methods stagnate at the line search.*

	Number of PIN iterations				
Methods	$\lambda = 0$	$\lambda = 100$	$\lambda = 1000$	$\lambda = 3000$	$\lambda = 5000$
INB	6	6	15	-	9
ASPIN	6	-	-	-	-
MSPIN	5	5	5	5	5
	Average number of GMRES iterations per PIN				
INB	12	18	14	$*$	7
ASPIN	21	$*$	$*$	$*$	$*$
MSPIN	10	9	8	8	7

where u_x is discretized using forward differences, and the discretization places v at staggered points. This is example 28 in the scalable nonlinear equations solvers subdirectory of PETSc, modified by the addition of a quadratic term in u, so that both components are nonlinear after splitting. Fifty grid points are used for this problem. The function $u_0(x) = x(1-x)$ and $v_0(x) = 1+0.5\sin(2\pi x)$ are used to be the initial guesses for $u(x)$ and $v(x)$. Considering the partition with respect to u and v, we split the problem into two submodels G and H that are dominant in u and v, respectively. The system G consists of the discretized equations of (5.4) with u-unknowns using the most recent values of v as the coefficients, while the discretized equations of (5.5) with v-unknowns belong to the system H, where the values of u_x are computed using the most recent values of u. We set the tolerance parameters as $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, $\epsilon_{global-linear-rtol} = 10^{-8}$, and $\epsilon_{sub-nonlinear-rtol} = 10^{-3}$ is the stopping condition for the subproblems nonlinear iterations. The analytical Jacobian matrices are used in this example.

Table 3 shows the number of Newton iterations and linear iterations for the different methods on problems with different parameters λ. The MSPIN method is effective in reducing the number of global Newton iterations, but the ASPIN method is very sensitive to the parameter λ. The failure of the ASPIN method and the INB method happens when the line search fails.

5.3. Driven cavity flow problem. We consider the two-dimensional lid-driven cavity flow problem [5, 6, 22] in the domain $\Omega = (0,1) \times (0,1)$ with three unknowns: the velocity u, v and the vorticity ω.

$$\begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \\ -\frac{1}{Re}\Delta \omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = 0. \end{cases} \tag{5.6}$$

Here $u = 1, v = 0$ on the top boundary and $u = 0, v = 0$ on the other boundaries. The boundary condition on ω is given by its definition:

$$\omega(x,y) = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{5.7}$$

Considering the partition with respect to velocity unknowns and the vorticity unknowns, we split the system (5.6) into two submodels:

$$G: \quad \begin{cases} -\Delta u - \frac{\partial \omega}{\partial y} = 0, \\ -\Delta v + \frac{\partial \omega}{\partial x} = 0, \end{cases} \tag{5.8}$$

and

$$H: \quad -\frac{1}{Re}\Delta\omega + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = 0. \tag{5.9}$$

A finite difference scheme with the five-point stencil is used to discretize the PDEs, which is a standard PETSc example [2]. Upwinding is employed 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. As the Reynolds number is increased on a fixed mesh, the discretization artificially diffuses the boundary layers, but the point of pushing the Reynolds number beyond the discretization is to obtain parameterized nonlinear algebraic problems of fixed size that are increasingly difficult for (unpreconditioned) globalized Newton methods.

Here, the subproblems obtained from the discretization of (5.8) and (5.9) are linear, which are solved by GMRES with the ILU(0) preconditioner. We set the tolerance parameters as $\epsilon_{global-nonlinear-rtol} = 10^{-10}$, $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{sub-linear-rtol} = 10^{-3}$. The Jacobian matrices are formed using a finite difference scheme. We run the test for the uniform meshes 64×64, 128×128, and 256×256.

Newton can be sensitive to the initial guess. Our first tests set the initial guess to be zero for u, v, and ω. Table 4 compares a global inexact Newton-GMRES-ILU with backtracking (INB) against ASPIN and MSPIN. (A fill level of 6 is chosen for ILU since level 5 or 6 minimizes runtime for the linear solver overall across all convergent experiments, balancing cost per iteration against the number of iterations.) For ASPIN and MSPIN, the two linear problems involved in (2.13) and (2.29) are solved directly. For small Reynolds numbers, corresponding to weak nonlinearity, the INB can have the smallest number of outer Newton steps. However, the advantage is quickly lost with increasing Reynolds number, and for modest Reynolds number, INB fails to converge nonlinearly altogether even with exact linear solves. The threshold for the 128×128 mesh for this problem and discretization was determined to be $Re = 770.0$ in [5]. Both the field-split preconditioned inexact Newton methods converge for all Reynolds numbers. The MSPIN method is superior in nonlinear iterations to the ASPIN method. The average number of GMRES iterations increases when the mesh size is increased for a fixed Reynolds number. In contrast, the number of nonlinear iterations using the ASPIN method increases, and the average number of GMRES iterations increases considerably, when the mesh size varies from 64×64 to 128×128 to 256×256.

Next we change the initial guess. The initial guess is still zero for u, v, and ω, except $u = 1$ on the top boundary. The ASPIN method fails to converge once the Reynolds number passes the value $Re = 1136$ on the 128×128 mesh, and it suffers from stagnation before the Reynolds number reaches 1136, while MSPIN converges for a much larger range of Reynolds numbers, as shown in Figure 4. The MSPIN method is not very sensitive to changes of parameters via the tests above, such as the initial guess, the mesh size, and the Reynolds number, and is therefore more robust than the ASPIN method.

TABLE 4

Global nonlinear and linear iterations using globalized INB, ASPIN, and MSPIN on different mesh sizes. The initial guess is zero for u, v, and ω. $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{global-nonlinear-rtol} = 10^{-10}$. The finite difference step size for the matrix-free Jacobian applications is 10^{-8}. "∗∗" indicates that nonlinear iterations are not available, because linear iterations fail to converge after 10,000 steps.

64×64 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	17
ASPIN	4	9	10
MSPIN	4	5	4
Average number of GMRES iterations per PIN			
INB	19	24	26
ASPIN	23	37	39
MSPIN	13	18	15
128×128 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	∗∗
ASPIN	4	9	13
MSPIN	4	5	5
Average number of GMRES iterations per PIN			
INB	37	62	-
ASPIN	24	42	64
MSPIN	14	21	20
256×256 mesh			
Number of PIN iterations			
Methods	$Re = 10$	$Re = 100$	$Re = 1000$
INB	3	5	∗∗
ASPIN	4	10	18
MSPIN	4	6	6
Average number of GMRES iterations per PIN			
INB	93	200	-
ASPIN	25	47	139
MSPIN	15	24	28

The nonlinear preconditioning demonstrated here is one of several globalization techniques for systems for which INB fails if applied directly, such as the driven cavity on a fine mesh at high Reynolds number. One may apply mesh sequencing, initializing the fine mesh with converged results from recursively coarsened meshes. One may invoke parameter continuation, initializing the high Reynolds problem with solutions at recursively lower values (noting that at infinitesimal Reynolds number, the problem becomes linear and SPD in each variable). One may also apply pseudo-transient continuation by prepending a temporal evolution term to the vorticity transport equation, and approaching the desired steady state through a physically motivated transient, beginning with a very small timestep and adaptively increasing it in inverse proportion to fractional power of steady-state residual reductions. These and other strategies are discussed in [19], with references to the literature. Our purpose in this contribution is to offer a complementary technique to desensitize Newton to the initial guess and improve the nonlinear conditioning in a fundamental way that may, if necessary, be further combined with the others.

Finally we show some parallel results on a 16-rack IBM Blue Gene/P supercomputer with quad-core PowerPC 450 I/O nodes (850 MHz, 4 GB RAM). In Table 5, we consider the most nonlinear and the most linearly ill-conditioned problem of Table 4, on which INB does not converge unassisted. On this, we demonstrate modest strong scalability for ASPIN and MSPIN within the PETSc framework. For ASPIN and MSPIN, all subproblems and linear problems involved in the nonlinearly preconditioned Jacobian application are solved by GMRES with BoomerAMG

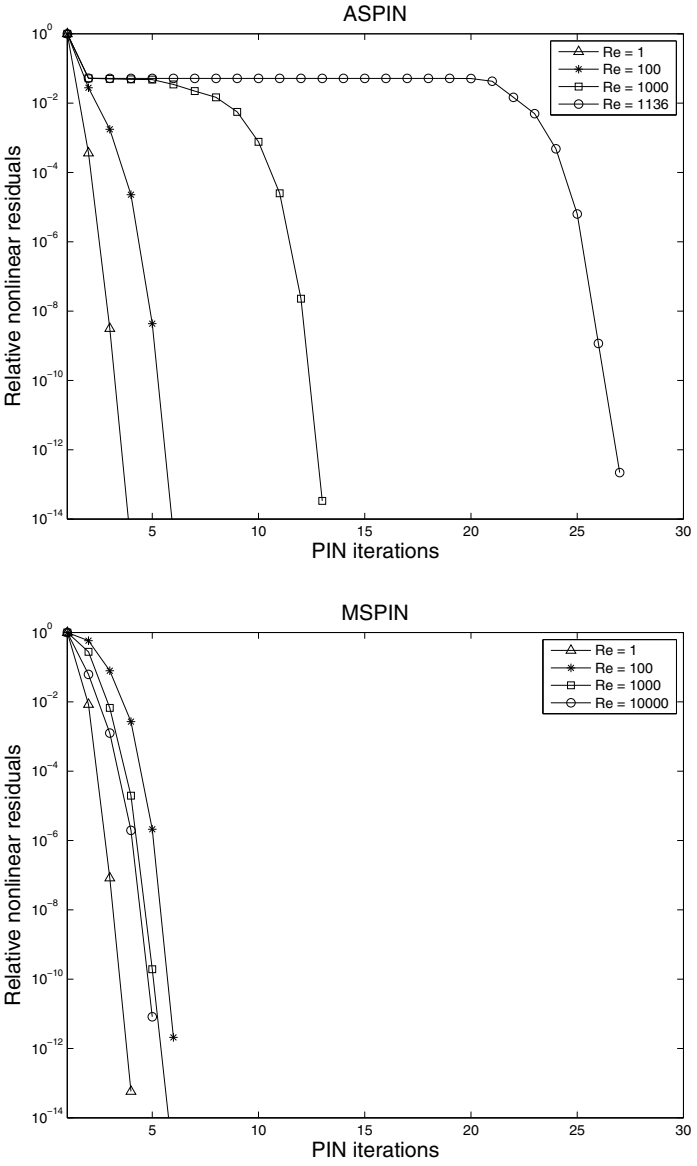

FIG. 4. *Nonlinear residual history for the driven cavity flow problem with different Reynolds numbers. The initial guess is still zero for* u, v, ω, *except* $u = 1$ *on the top boundary.*

preconditioners [14]. We vary the number of processors at Reynolds number 1000 and the scaling behaviors are shown for both methods and all four relative tolerance tunings. Because performance is generally dependent on linear convergence tolerances, we experiment with four combinations of loose and tight tolerances for the individual subproblems and the systems in (2.13) and (2.29). We see that execution times are more sensitive to the outer tolerances than to the inner.

Figure 5 shows strong scaling behaviors for ASPIN and MSPIN on a 1024×1024 mesh at Reynolds number 1000, using 16, 32, 64, 128, 256, 512, and 1024

TABLE 5

Execution times for strong scaling of the lid-driven cavity for ASPIN and MSPIN on a 256×256 *mesh at Reynolds number 1000, for tight and loose relative convergence tolerances on the subproblems and global preconditioner linear systems solutions. The initial guess is zero for* u, v, *and* ω. $\epsilon_{global-linear-rtol} = 10^{-6}$, $\epsilon_{global-nonlinear-rtol} = 10^{-8}$. $\epsilon_{sub-rtol}$ *denotes the relative tolerance for the subproblems (which are linear in this example), and we specify* $\epsilon_{Jac-rtol}$ *as the relative tolerance for the linear problems in* (2.13) *and* (2.29). *The finite difference step size for the matrix-free Jacobian applications is* 10^{-8}. *"* N_p*" indicates the number of processors, which does not have to be square. Performance for INB is not shown since it fails to converge on this mesh and this Reynolds number from a zero initial guess.*

Execution time (s)					
	256 × 256 mesh				
Methods	N_p	$\epsilon_{sub-rtol} = 10^{-3}$ $\epsilon_{Jac-rtol} = 10^{-3}$	$\epsilon_{sub-rtol} = 10^{-3}$ $\epsilon_{Jac-rtol} = 10^{-6}$	$\epsilon_{sub-rtol} = 10^{-6}$ $\epsilon_{Jac-rtol} = 10^{-3}$	$\epsilon_{sub-rtol} = 10^{-6}$ $\epsilon_{Jac-rtol} = 10^{-6}$
ASPIN	4	2363.98	3273.88	2194.43	3219.34
	16	687.68	943.91	654.32	976.95
	32	397.12	589.86	395.26	588.95
	64	272.4	412.27	276.03	405.92
MSPIN	4	175.31	245.59	199.64	248.18
	16	57.04	74.06	56.74	74.48
	32	32.17	45.86	34.55	46.33
	64	22.31	31.64	23.90	31.79

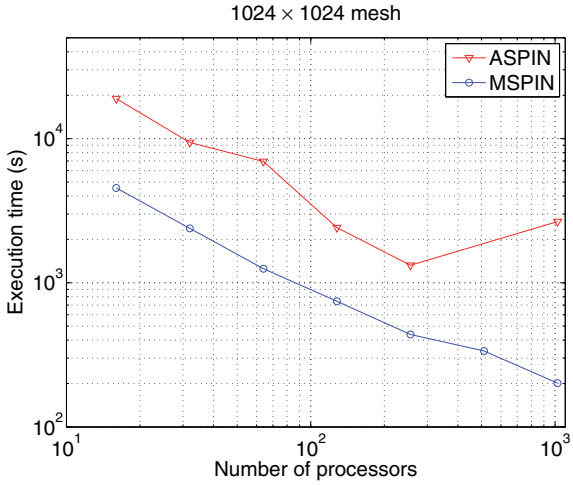

FIG. 5. *Strong scaling for the driven cavity flow problem on a* 1024×1024 *mesh at Reynolds number* 1000. *The initial guess is still zero for* u, v, ω. $\epsilon_{global-linear-rtol} = 10^{-3}$, $\epsilon_{global-nonlinear-rtol} = 10^{-8}$, $\epsilon_{sub-rtol} = 10^{-3}$, *and* $\epsilon_{Jac-rtol} = 10^{-3}$. $\epsilon_{sub-rtol}$ *denotes the relative tolerance for the subproblems (which are linear in this example), and we specify* $\epsilon_{Jac-rtol}$ *as the relative tolerance for the linear problems in* (2.13) *and* (2.29). *The finite difference step size for the matrix-free Jacobian applications is* 10^{-8}. *Execution time for ASPIN using* 512 *processors is not shown since it fails to converge on this mesh and this Reynolds number from a zero initial guess.*

processors. For ASPIN and MSPIN, all subproblems and linear problems involved in the nonlinearly preconditioned Jacobian application are also solved by GMRES with BoomerAMG preconditioners. In terms of the execution time, MSPIN scales well for up to 1024 processors, while ASPIN scales well for up to 256 processors and it fails to converge using 512 processors on this mesh and this Reynolds number from a zero initial guess.

6. Concluding comments. In this paper, we present nonlinearly preconditioned inexact Newton methods with physics-based field-split partitioning and extend the existing theory for existence and uniqueness for monotone nonlinear systems in a straightforward way to the Gauss–Seidel version. The theory does not predict convergence rates; however, numerical results illustrate that the nonlinear preconditioners are effective in improving on the performance of the global Newton iterations. The MSPIN algorithm is more robust than ASPIN.

Field splitting is an option that can be combined with domain-splitting. The former is often useful for coarse granularity where intuition can play a role in problem decomposition. The latter is extensible to fine granularity. A combination of the two may be natural for large-scale simulations with natural partitions between the models. Even though the domain-based ASPIN is robust for some problems [5, 7, 16], local subproblems may still fail to converge due to unbalanced nonlinearities or the lack of a good initial guess [21]. The field-split algorithms provide additional options.

Appendix A. The nonlinear function $F(x)$ is split into $N \geqslant 2$ components representing different physical aspects as

$$F(x) = \begin{bmatrix} \hat{F}_1(u_1, \ldots, u_N) \\ \vdots \\ \hat{F}_N(u_1, \ldots, u_N) \end{bmatrix} = 0, \quad x = [u_1, \ldots, u_N]^T. \tag{A.1}$$

The preconditioned function

$$\mathcal{F}_{GS}(x) = \begin{bmatrix} T_1(u_1, \ldots, u_N) \\ \vdots \\ T_N(u_1, \ldots, u_N) \end{bmatrix} \tag{A.2}$$

is obtained by solving the following equations sequentially:

$$\begin{aligned} &\hat{F}_1(\delta_1, u_2, u_3, \ldots, u_N) = 0, \\ &\hat{F}_2(\delta_1, \delta_2, u_3, u_4 \ldots, u_N) = 0, \\ &\qquad \vdots \\ &\hat{F}_N(\delta_1, \delta_2, \delta_3, \delta_4, \ldots, \delta_N) = 0, \end{aligned} \tag{A.3}$$

where $\delta_i = u_i - T_i$, $i = 1, 2, \ldots, N$.

We define

$$\hat{u}_i = \begin{bmatrix} u_1 \\ \vdots \\ u_i \end{bmatrix}, \quad \hat{u}_i^c = \begin{bmatrix} u_{i+1} \\ \vdots \\ u_N \end{bmatrix}, \quad \hat{T}_i = \begin{bmatrix} T_1 \\ \vdots \\ T_i \end{bmatrix}, \quad \hat{\delta}_i = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_i \end{bmatrix}, \tag{A.4}$$

and then $\hat{F}_i$ in (A.3) is written as

$$\hat{F}_i(\hat{u}_i - \hat{T}_i, \hat{u}_i^c) = 0, \tag{A.5}$$

where $i = 1, 2, \ldots, N$. Taking the derivative of (A.5) with respect to $\hat{u}_i$, we have

$$\frac{\partial \hat{F}_i}{\partial \hat{\delta}_i} \left(I_i - \frac{\partial \hat{T}_i}{\partial \hat{u}_i} \right) = 0, \tag{A.6}$$

where I_i is the identity matrix that has the same dimension as the $\hat{u}_i$ block. Next, we take the derivative of (A.5) with respect to $\hat{u}_i^c$, yielding

$$\frac{\partial \hat{F}_i}{\partial \hat{\delta}_i}\left(-\frac{\partial \hat{T}_i}{\partial \hat{u}_i^c}\right)+\frac{\partial \hat{F}_i}{\partial \hat{u}_i^c}=0. \tag{A.7}$$

Using (A.6) and (A.7), we have

$$\begin{bmatrix}\frac{\partial \hat{F}_i}{\partial \hat{\delta}_i} & \frac{\partial \hat{F}_i}{\partial \hat{u}_i^c}\end{bmatrix}=\frac{\partial \hat{F}_i}{\partial \hat{\delta}_i}\begin{bmatrix}\frac{\partial \hat{T}_i}{\partial \hat{u}_i} & \frac{\partial \hat{T}_i}{\partial \hat{u}_i^c}\end{bmatrix}, \tag{A.8}$$

from which we deduce that

$$A=\begin{bmatrix}\frac{\partial \hat{F}_1}{\partial \delta_1} & \frac{\partial \hat{F}_1}{\partial u_2} & \frac{\partial \hat{F}_1}{\partial u_3} & \dots & \dots & \frac{\partial \hat{F}_1}{\partial u_N}\\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & \frac{\partial \hat{F}_2}{\partial u_3} & \dots & \dots & \frac{\partial \hat{F}_2}{\partial u_N}\\ \vdots & \vdots & \vdots & & & \vdots\\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \frac{\partial \hat{F}_N}{\partial \delta_3} & \dots & \dots & \frac{\partial \hat{F}_N}{\partial \delta_N}\end{bmatrix} \tag{A.9}$$

$$=\begin{bmatrix}\frac{\partial \hat{F}_1}{\partial \delta_1} & & & & \\ \frac{\partial \hat{F}_2}{\partial \delta_1} & \frac{\partial \hat{F}_2}{\partial \delta_2} & & & \\ \vdots & \vdots & \ddots & & \\ \frac{\partial \hat{F}_N}{\partial \delta_1} & \frac{\partial \hat{F}_N}{\partial \delta_2} & \frac{\partial \hat{F}_N}{\partial \delta_3} & \dots & \frac{\partial \hat{F}_N}{\partial \delta_N}\end{bmatrix}\begin{bmatrix}\frac{\partial T_1}{\partial u_1} & \dots & \frac{\partial T_1}{\partial u_N}\\ \vdots & & \vdots\\ \frac{\partial T_N}{\partial u_1} & \dots & \frac{\partial T_N}{\partial u_N}\end{bmatrix}.$$

Hence, the Jacobian of the preconditioned system (A.2) corresponds to the block Gauss–Seidel linear preconditioning for the matrix A.

Acknowledgments. The authors thank anonymous referees for excellent suggestions on improving this paper. The PETSc group of Argonne National Laboratory is gratefully acknowledged for their software support.

REFERENCES

[1] H.-B. An, *On convergence of the additive Schwarz preconditioned inexact Newton method*, SIAM J. Numer. Anal., 43 (2005), pp. 1850–1871.

[2] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, *The Portable, Extensible Toolkit for Scientific Computing*, (2014). http://www.mcs.anl.gov/petsc

[3] P. Brune, M. G. Knepley, B. F. Smith, and X. Tu, *Composing Scalable Nonlinear Algebraic Solvers*, Preprint ANL/MCS-P2010-0112, Argonne National Laboratory, Argonne, IL, 2013.

[4] X.-C. Cai and M. Dryja, *Domain decomposition methods for monotone nonlinear elliptic problems*, Contemp. Math., 180 (1994), pp. 21–27.

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

[6] X.-C. Cai, D. E. Keyes, and L. Marcinkowski, *Nonlinear additive Schwarz preconditioners and application in computational fluid dynamics*, Internat. J. Numer. Methods. Fluids, 40 (2002), pp. 1463–1470.

[7] X.-C. Cai, D. E. Keyes, and D. P. Young, *A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow*, in Proceedings of the 13th International Conference on Domain Decomposition Methods, Lyon, France, 2000, pp. 343–350.
[8] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.
[9] M. Dryja and W. Hackbusch, *On the nonlinear domain decomposition method*, BIT, 37 (1997), pp. 296–311.
[10] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.
[11] 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.
[12] R. Ernst, B. Flemisch, and B. Wohlmuth, *A multiplicative Schwarz method and its application to nonlinear acoustic-structure interaction*, ESAIM Math. Model. Numer. Anal., 43 (2009), pp. 487–506.
[13] A. Granas and J. Dugundji, *Fixed Point Theory*, Springer, New York, 2003.
[14] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.
[15] F.-N. Hwang, *Some Parallel Linear and Nonlinear Schwarz Methods with Applications in Computational Fluid Dynamics*, Ph.D. thesis, University of Colorado at Boulder, Boulder, CO, 2004.
[16] 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.
[17] F.-N. Hwang and X.-C. Cai, *A combined linear and nonlinear preconditioning technique for incompressible Navier-Stokes equations*, in Applied Parallel Computing: State of the Art in Scientific Computing Lecture Notes in Comput. Sci. 3732, J. Dongarra, K. Madsen, and J. Wasniewski, eds., Springer, New York, 2006, pp. 313–322.
[18] F.-N. Hwang and X.-C. Cai, *A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1603–1611.
[19] 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.
[20] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes, *An analysis of approximate nonlinear elimination*, SIAM J. Sci. Comput., 17 (1996), pp. 538–559.
[21] S. H. Lui, *Nonlinearly preconditioned Newton's method*, in Proceedings of the 14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico, 2002, pp. 95–105.
[22] L. Marcinkowski and X.-C. Cai, *Parallel performance of some two-level ASPIN algorithms*, in Domain Decomposition Methods in Science and Engineering, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. H. W. Hoppe, D. E. Keyes, J. Periaux, O. Pironneau, and J. Xu, eds., Springer, New York, 2005, pp. 639–646.
[23] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, Philadelphia, 2000.
[24] M. Pernice and H. F. Walker, *NITSOL: A Newton iterative solver for nonlinear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 302–318.
[25] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.