Nonlinear Additive Schwarz Preconditioners and Applications in Computational Fluid Dynamics

Xiao-Chuan Cai* David E. Keyes† Leszek Marcinkowski‡

Abstract

The focus of this paper is on the numerical solution of large sparse nonlinear systems of algebraic equations on parallel computers. These nonlinear systems often arise from the discretization of nonlinear partial differential equations, such as the Navier-Stokes equations for fluid flows, using finite element or finite difference methods. A traditional inexact Newton method, applied directly to the discretized system, does not work well when the nonlinearities in the algebraic system become unbalanced. In this paper we study some preconditioned inexact Newton algorithms, including the single-level and multilevel nonlinear additive Schwarz preconditioners. Some results for solving the high Reynolds number incompressible Navier-Stokes equations will be reported.

1 Introduction

Newton's method is one of the most popular techniques for solving large nonlinear systems of equations in engineering applications due to the fact that the method is easy to implement and converges quickly if the starting point is inside the domain of convergence. However, it is well-known that the radius of the domain of convergence of Newton's method is inversely

*Department of Computer Science, University of Colorado, Boulder, CO 80309-0430 (*cai@cs.colorado.edu*). The work was supported in part by NSF under ACI-0083004, ACI-0072089, and ACI-0112930, and by LLNL under subcontract B218772.

†Department of Mathematics & Statistics, Old Dominion University, Norfolk, VA 23529-0077 (*keyes@icase.edu*). This work was supported in part by NASA under contract NAS1-19480 and by LLNL under subcontract B347882.

‡Department of Computer Science, University of Colorado, Boulder, CO 80309-0430 (*leszek@cs.colorado.edu*). The work was supported in part by NSF under ACI-0072089, and by LLNL under subcontract B218772.

proportional to the relative nonlinearity of the function; i.e., as the relative nonlinearity increases the domain of convergence shrinks, and as a result, finding a good starting point becomes very difficult [7]. Many globalization techniques have been developed in order to find a good starting point, such as the linesearch and trust region methods [7], continuation methods [13], mesh sequencing methods [16], etc. In this paper, we present a different approach that increases the domain of convergence of Newton's method by reducing the nonlinearity of the function. Consider a given nonlinear function $F : \Re^n \to \Re^n$. We are interested in calculating a vector $u_* \in \Re^n$, such that

$$F(u_*) = 0, \tag{1}$$

starting from an initial guess $u^{(0)} \in \Re^n$. Here $F = (F_1, \ldots, F_n)^T$, $F_i = F_i(u_1, \ldots, u_n)$, and $u = (u_1, \ldots, u_n)^T$. Inexact Newton algorithms (IN) [7, 10] are commonly used for solving such systems. In this paper, we work in the framework of nonlinearly preconditioned inexact Newton algorithms (PIN), recently introduced in [4]. In other words, we try to find the solution u_* of equation (1) by solving an equivalent system of nonlinear equations

$$\mathcal{F}(u_*) = 0. \tag{2}$$

(1) and (2) are equivalent in the sense that they have the same solution. Other than having the same solution, the nonlinear functions $F(\ )$ and $\mathcal{F}(\ )$ may have completely different forms.

2 Single-level nonlinear additive Schwarz preconditioning

In this section, we describe a nonlinear preconditioner based on the additive Schwarz method [2, 8]. Let

$$S = (1, \ldots, n)$$

be an index set; i.e., one integer for each unknown u_i and F_i. We assume that $S_1, \ldots, S_N$ is a partition of S in the sense that

$$\bigcup_{i=1}^{N} S_i = S, \text{ and } S_i \subset S.$$

Here we allow the subsets to have overlap. Let n_i be the dimension of S_i; then, in general,

$$\sum_{i=1}^{N} n_i \geq n.$$

Using the partition of S, we introduce subspaces of $\Re^n$ and the corresponding restriction and extension matrices. For each S_i we define $V_i \subset \Re^n$ as

$$V_i = \{v | v = (v_1, \ldots, v_n)^T \in \Re^n, v_k = 0, \text{ if } k \notin S_i\}$$

and a $n \times n$ restriction (also extension) matrix I_{S_i} whose kth column is either the kth column of the $n \times n$ identity matrix $I_{n\times n}$ if $k \in S_i$ or zero if $k \notin S_i$. Note that the matrix I_{S_i} is always symmetric and the same matrix can be used as both restriction and extension operator. Many other forms of restriction/extension are available in the literature; however, we only consider the simplest form in this paper.

Using the restriction operator, we define the subdomain nonlinear function as

$$F_{S_i} = I_{S_i} F.$$

We next define the major component of the algorithm, namely the nonlinearly preconditioned function. For any given $v \in \Re^n$, define $T_i(v) \in V_i$ as the solution of the following subspace nonlinear system

$$F_{S_i}(v - T_i(v)) = 0,$$

for $i = 1, \ldots, N$. We introduce a new function

$$\mathcal{F}^{(1)}(u) = \sum_{i=1}^{N} T_i(u), \tag{3}$$

which we will refer to as the nonlinearly preconditioned $F(u)$. The one-level nonlinear additive Schwarz preconditioned inexact Newton algorithm (ASPIN or ASPIN(1)) is defined as: Find the solution u^* of (1) by solving the nonlinearly preconditioned system

$$\mathcal{F}^{(1)}(u) = 0 \tag{4}$$

with an inexact Newton method using $u^{(0)}$ as the initial guess. As shown in [4], ASPIN(1) is nonlinearly scalable, but the number of iterations in the global linear solver increases as the number of subdomains (or the number of processors as in our implementation) increases. A multilevel version of ASPIN(1) is therefore introduced below, which is scalable both nonlinearly and linearly.

3 Two-level nonlinear additive Schwarz preconditioning

In this section, we describe a parallel nonlinear preconditioner based on the two-level additive Schwarz method [2, 8]. The focus is on the construction of the coarse space operator. We will refer to the nonlinear algebraic system (1) as the *fine system* which has n unknowns and n equations. We also need a *coarse system*,

$$F^c(u^c_*) = 0, \tag{5}$$

which is a nonlinear algebraic system with n^c unknowns and n^c equations. The coarse and fine functions $F(u)$ and $F^c(u^c)$ approximate each other in a certain sense.

We next define the grid transfer operators. Note that we definitions are quite general, for example, the coarse and fine grids don't need to be nested. Let $S^c = (1, \ldots, n^c)$ an index set, i.e., one integer for each unknown of the coarse system, and assume that $S^c_1, \ldots, S^c_N$ is a partition of S^c in the sense that $\bigcup_{i=1}^N S^c_i = S^c$. For simplicity, we partition the fine and the coarse systems into the same number of subsets. Also for simplicity, in our parallel implementation, we will allocate the subsystems corresponding to the index sets S_i and S^c_i to the same processor. We define the subdomain fine to coarse restriction operator as $R_i : S_i \longrightarrow S^c_i$, in the sense that for each vector $v_i \in V_i$, there is a unique vector $v^c_i \in V^c_i$, such that

$$v^c_i = R_i v_i,$$

where R_i is a n_i by n^c_i matrix. In a similar way, we can introduce an extension operator from the coarse subspace S^c_i to the fine subspace S_i, $E_i : S_i \longrightarrow S^c_i$. In practice, E_i is usually taken as the transpose of the matrix R_i. Even though the subsets S^c_i and S^c_j may overlap each other, the restriction operators R_i and R_j are consistent in the sense that for any $v \in \Re^n$, if $k \in S^c_i \cap S^c_j$, then

$$(R_i v)_k = (R_j v)_k,$$

where $(\ )_k$ indicates the value of the kth component of the vector. We define a global fine to coarse restriction operator $R^c : \Re^n \longrightarrow \Re^{n^c}$ as follows: For any $v \in \Re^n$, the k component of $R^c v$ is defined as

$$(R^c v)_k = (R_i v)_k, \text{ if } k \in S^c_i.$$

A global coarse to fine extension operator E^c can be defined as the transpose of R^c. To define the coarse function $T_0 : \Re^n \longrightarrow \Re^n$, we first introduce a projection $T^c : \Re^n \longrightarrow \Re^{n^c}$ as follows: For any given $v \in \Re^n$, $T^c v$ satisfies the coarse nonlinear system

$$F^c(T^c(v)) = R^c F(v). \tag{6}$$

We assume (6) has a unique solution. Associated with T^c, we define an operator $T_0 : \Re^n \longrightarrow \Re^n$ by

$$T_0(v) = E^c T^c(v). \tag{7}$$

Suppose that T_0 is given as in (7); it is easy to see that $T_0(u_*)$ can be computed without knowing the exact solution u_* itself. In fact, from (6), we have

$$T_0(u_*) = E^c u_*^c,$$

which is the exact solution of the coarse system (5). Throughout this paper, we assume that the coarse solution u_*^c is given, through a pre-processing step. We can introduce a new nonlinear function $\Re^n \longrightarrow \Re^n$ by

$$\mathcal{F}^{(2)}(u) = T_0(u) - T_0(u_*) + \sum_{i=1}^{N} T_i(u), \tag{8}$$

which we will refer to as the nonlinearly preconditioned $F(u)$. The two-level nonlinear additive Schwarz preconditioned inexact Newton algorithm (ASPIN(2)) is defined as follows: Find the solution u_* of (1) by solving the nonlinearly preconditioned system

$$\mathcal{F}^{(2)}(u) = 0 \tag{9}$$

with an inexact Newton method using $u^{(0)}$ as the initial guess. A more complete description of ASPIN(2) can be found in [5].

4 Brief review of inexact Newton methods

Consider a nonlinear system, for example (1). Suppose $u^{(k)}$ is the current approximate solution; a new approximate solution $u^{(k+1)}$ can be computed through the following steps (IN):

Step 1: Find the inexact Newton direction $p^{(k)}$ such that

$$\|F(u^{(k)}) - F'(u^{(k)})p^{(k)}\| \leq \eta_k \|F(u^{(k)})\| \tag{10}$$

Step 2: Compute the new approximate solution

$$u^{(k+1)} = u^{(k)} - \lambda^{(k)} p^{(k)}. \tag{11}$$

Here $\eta_k \in [0, 1)$ is a scalar that determines how accurately the Jacobian system needs to be solved using, for example, Krylov subspace methods [10, 11]. $\lambda^{(k)}$ is another scalar that determines how far one should go in the selected inexact Newton direction [7]. IN has two well-known features, namely, (a) if the initial guess is close enough to the desired solution then the convergence is very fast provided that the η_k's are sufficiently small, and (b) such a good initial guess is generally very difficult to obtain, especially for nonlinear equations that have unbalanced nonlinearities [14]. The step length $\lambda^{(k)}$ is often determined by the components with the strongest nonlinearities, and this may lead to an extended period of stagnation in the nonlinear residual curve, [3, 16].

In this paper, we apply IN to systems (4) or (9), instead of (1). The line-search parameter $\lambda^{(k)}$ is determined using the preconditioned merit function

$$\frac{1}{2}\|\mathcal{F}\|^2$$

which, by design, has more balanced nonlinearity than $1/2\|F\|^2$.

5 A driven cavity flow problem

In this section, we present some numerical results on the following two-dimensional driven cavity flow problem [12], using the velocity-vorticity formulation, in terms of the velocity u, v, and the vorticity ω, defined on the unit square $\Omega = (0, 1) \times (0, 1)$,

$$\begin{cases} -\Delta u - \dfrac{\partial \omega}{\partial y} &= 0 \\ -\Delta v + \dfrac{\partial \omega}{\partial x} &= 0 \\ -\frac{1}{Re}\Delta \omega + u\dfrac{\partial \omega}{\partial x} + v\dfrac{\partial \omega}{\partial y} &= 0. \end{cases} \tag{12}$$

Here Re is Reynolds number. The boundary conditions are:

- bottom, left and right: $u = v = 0$
- top: $u = 1$, $v = 0$

The boundary condition on ω is given by its definition:

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

We test several different Reynolds numbers in the experiments and the numbers are given in the tables below. The usual uniform mesh finite difference approximation with the 5-point stencil is used to discretize the boundary value problem. Upwinding is used for the divergence (convective) terms and central differencing for the gradient (source) terms. To obtain a nonlinear algebraic system of equations F, we use natural ordering for the mesh points, and at each mesh point, we arrange the knowns in the order of u, v, and ω. The partitioning of F is through the partitioning of the mesh points in a checkerboard fashion for both the fine and the coarse grid. The coarse to fine interpolation is defined using the coarse grid bilinear finite element basis functions. $overlap = 1$ is used for all the calculations. The implementation is done using PETSc [1], and the results are obtained on a cluster of DEC workstations. Double precision is used throughout the computations. The initial iterate is zero for u, v and ω. We report here only the machine independent properties of the algorithms.

We stop the global PIN iterations if

$$\|\mathcal{F}(u^{(k)})\| \leq 10^{-10}\|\mathcal{F}(u^{(0)})\|.$$

The same stopping condition is used for the coarse grid nonlinear systems, which can be solved by either a Newton-Krylov-Schwarz method, or the one-level ASPIN, based on the same mesh partition.

The Jacobian systems are solved with GMRES restarting at 30. The global linear iteration for solving the global Jacobian system is stopped if the relative tolerance

$$\|\mathcal{F}(u^{(k)}) - \mathcal{F}^{'}(u^{(k)})p^{(k)}\| \leq 10^{-3}\|\mathcal{F}(u^{(k)})\|$$

is satisfied. We remark that, unlike the Jacobian matrix of F, the Jacobian matrix $\mathcal{F}^{'}$ is usually not sparse and can not be computed explicitly. Following the techniques developed in [4], we approximate $\mathcal{F}^{'}$ on each subdomain by $J_{S_i}^{-1}J$, where $J = F^{'}$ and J_{S_i} is the restriction of J on the subdomain S_i.

Similarly on the coarse grid, we use $J_{S^c}^{-1}J$, where J_{S^c} is the restriction of J on the coarse grid. We do not use any preconditioners when solving the Jacobian problems.

At the kth global nonlinear iteration, nonlinear subsystems

$$F_{S_i}\left(u^{(k)} - g_i^{(k)}\right) = 0,$$

have to be solved. We use the standard IN with a cubic line search for such systems with initial guess $g_{i,0}^{(k)} = 0$. The local nonlinear iteration in subdomain S_i is stopped if the following condition is satisfied:

$$\left\| F_{S_i}\left(g_{i,l}^{(k)}\right)\right\| \leq 10^{-3}\left\| F_{S_i}\left(g_{i,0}^{(k)}\right)\right\|.$$

In Tables 1 and 2, we report the total number of global nonlinear iterations, the total number of linear iterations, and the average number of linear iterations per nonlinear iteration. For this particular test problem, the nonlinearity is determined mostly by the Reynolds number. As Re increases the nonlinear system becomes harder and harder to solve with the regular inexact Newton method [4]. However, as shown in Tables 1 and 2, ASPIN is not very sensitive to the increase of Re.

As expected from the classical theory of additive Schwarz methods, the one-level algorithm, ASPIN(1), is not scalable with respective to the number of subdomains, which is the same as the number of processors in our parallel implementation. This is reflected in the average number of global linear iterations. By adding a coarse space, as in ASPIN(2), the number of global linear iterations can be reduced. We observe that the size of the coarse grid has to be sufficiently fine in order to remove the dependence on the number of subdomains. This can be seen clearly by comparing the last row in Table 2 and 3. In practice, a good coarse grid size is usually not easy to determine since it depends not only on the number of subdomains but also on the Reynolds number.

References

[1] S. Balay, W. Gropp, L. McInnes, and B. Smith, *The Portable, Extensible Toolkit for Scientific Computing, version 2.1.0*, `http://www.mcs.anl.gov/petsc`, code and documentation, 2001.

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

Table 1: $Re = 10^3$, fine mesh size 64×64, coarse mesh size 16×16.

	number of processors	*global nonlinear iterations*	*global linear iterations*	*average linear iteration per nonlinear step*
ASPIN(1)	$2 \times 2 = 4$	6	90	15
	$4 \times 4 = 16$	6	136	22
ASPIN(2)	$2 \times 2 = 4$	5	56	11
	$4 \times 4 = 16$	7	86	12

Table 2: $Re = 10^4$, fine mesh size 128×128, coarse mesh size 32×32 .

	number of processors	*global nonlinear iterations*	*global linear iterations*	*average linear iteration per nonlinear step*
ASPIN(1)	$2 \times 2 = 4$	10	175	17
	$4 \times 4 = 16$	7	172	24
ASPIN(2)	$2 \times 2 = 4$	8	124	15
	$4 \times 4 = 16$	7	150	21

[3] X.-C. Cai, W. Gropp, D. Keyes, R. Melvin, and D. Young, *Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246–265.

[4] X.-C. Cai and D. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 2000. (submitted)

[5] X.-C. Cai, D. Keyes, and L. Marcinkowski, *Two-level nonlinear Schwarz preconditioned inexact Newton methods*, 2001. (in preparation)

[6] X.-C. Cai, D. Keyes, and D. Young, *A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow*, Proceedings of the 13th International Conference on Domain Decomposition Methods, 2001.

[7] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, NJ, 1983.

Table 3: $Re = 10^4$, fine mesh size 128×128, coarse mesh size 64×64 .

	number of processors	*global nonlinear iterations*	*global linear iterations*	*average linear iteration per nonlinear step*
ASPIN(2)	$4 \times 4 = 16$	6	76	12

[8] M. Dryja and W. Hackbusch, *On the nonlinear domain decomposition method*, BIT, (1997), pp. 296-311.

[9] M. Dryja and O. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comp., 15 (1994), pp. 604–620.

[10] S. Eisenstat and H. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optimization, 4 (1994), pp. 393–422.

[11] S. Eisenstat and H. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16-32.

[12] C. Hirsch, *Numerical Computation of Internal and External Flows*, John Wiley & Sons, New York, 1990.

[13] C. Kelley and D. Keyes, *Convergence analysis of pseudo-transient continuation*, SIAM J. Num. Anal., 35 (1998), pp. 508-523.

[14] P. Lanzkron, D. Rose, and J. Wilkes, *An analysis of approximate nonlinear elimination*, SIAM J. Sci. Comput., 17 (1996), pp. 538–559.

[15] B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[16] D. Young, R. Mervin, M. Bieterman, F. Johnson, S. Samant, and J. Bussoletti, *A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics*, J. Comput. Phys., 92 (1991), pp. 1-66.