Parallel Performance of Some Two-Level ASPIN Algorithms

Leszek Marcinkowski[1⋆] and Xiao-Chuan Cai[2⋆⋆]

[1] Department of Mathematics, Informatics and Mechanics, Warsaw University, Banacha 2, 02-096 Warszawa, Poland
(lmarcin@mimuw.edu.pl,http://www.mimuw.edu.pl/~lmarcin/)
[2] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309-0430, USA
(cai@cs.colorado.edu,http://www.cs.colorado.edu/~cai/)

Summary. In this paper we study the parallel performance of some nonlinear additive Schwarz preconditioned inexact Newton methods for solving large sparse system of nonlinear equations arising from the discretization of partial differential equations. The main idea of nonlinear preconditioning is to replace an ill-conditioned nonlinear system by an equivalent nonlinear system that has more balanced nonlinearities. In addition to balance the nonlinearities through nonlinear preconditioning, we also need to make sure that the multilayered iterative solver is scalable with respect to the number of processors. We focus on some two-level nonlinear additive Schwarz preconditioners, and show numerically that these two-level methods can reduce the nonlinearities and at the same time maintain the parallel scalability. Parallel numerical results for some high Reynolds number incompressible Navier-Stokes equations will be presented.

1 Introduction

We study Newton type algorithms for solving a nonlinear system of equations

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

starting from an initial guess $u^{(0)} \in \Re^n$. Here $F = (F_1, \ldots, F_n)^T$, $F_i = F_i(u_1, \ldots, u_n)$ are given functions which are often the result of the discretization of some nonlinear partial differential equations, such as the incompressible Navier-Stokes equations for fluid flows, using finite element or finite difference methods. For such nonlinear systems, some parallel nonlinear additive Schwarz

⋆ This work was partially supported by Polish Scientific Grant 2/P03A/005/24.
⋆⋆ The work was partially supported by the Department of Energy, FC02-01ER-25479, and by the National Science Foundation, CCR-0219190, ACI-0072089 and ACI-0305666.

preconditioned inexact Newton methods (ASPIN) were recently proposed in Cai and Keyes [2002], and Cai et al. [2002]. ASPIN has been applied successfully to some rather difficult problems such as the transonic full potential flows (Cai et al. [2000]), and the high Reynolds number incompressible Navier-Stokes flows (Hwang and Cai [2003]). In this paper we compare the parallel performance of a one-level method (Cai and Keyes [2002]), a two-level method (Cai et al. [2002]), and a slightly modified two-level method to be presented in this paper. In the modified two-level method, the initial guess $u^{(0)}$ is replaced by a fine grid interpolation of the coarse grid solution. It turns out in some situations the small modification has some major impact on the overall performance of the algorithm. The focus of this paper is on the linear and nonlinear scalability issues of the methods, and our discussions will be based on the numerical results for solving some high Reynolds number incompressible Navier-Stokes equations on distributed memory parallel computers with modest number of processors.

2 Algorithm description

In the rest of the paper we shall refer to the nonlinear algebraic system (1) as the *fine grid system*, or simply the fine system, which has n unknowns and n equations. In order to introduce the two-level algorithm, we assume that there is a "coarse" version of (1) in the following form

$$F^c(u^c_*) = 0, \tag{2}$$

which is a nonlinear algebraic system with n^c unknowns and n^c equations. Usually $n^c << n$. Such a coarse system can be obtained by the discretization of the same differential equations on a coarser grid. The coarse and fine functions $F(u)$ and $F^c(u^c)$ approximate each other in certain sense.

Inexact Newton algorithms (IN) (Eisenstat and Walker [1994]) 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 (Cai and Keyes [2002]). In other words, we try to find the solution u_* of equation (1) by solving an equivalent system of nonlinear equations

$$\mathcal{F}(u_*) = 0. \tag{3}$$

(1) and (3) 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. We will define the function $\mathcal{F}$ using the restriction of F on subspaces, and the coarse function F^c in the case of multilevel methods.

2.1 A one-level method

We first introduce the subspaces by an overlapping partition of $S = (1, \ldots, n)$, which is an index set for the system (1); 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 $\bigcup_{i=1}^{N} S_i = S$, 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 $k \notin S_i$. 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 $i = 1, \ldots, N$. Taking the sum of the all T_is, we have a new function

$$\mathcal{F}^{(1)}(u) = \sum_{i=1}^{N} T_i(u), \tag{4}$$

The operators T_i and $\mathcal{F}^{(1)}$ were introduced by Dryja and Hackbusch [1997] in which a version of a nonlinear Richardson method was applied to solve the nonlinear system corresponding to (4).

Algorithm 2.1 (ASPIN(1)) *Obtain an approximate solution of u_* by solving*

$$\mathcal{F}^{(1)}(u_*) = 0$$

using the inexact Newton method with $u^{(0)}$ as the initial guess (Cai and Keyes [2002]).

It is worth to note that under some assumptions it was proven by Dryja and Hackbusch [1997] and Cai and Keyes [2002] that the local problems have unique solutions, thus T_i are well defined. It is also shown there that the Jacobian of the preconditioned system is well defined.

To apply an inexact Newton method to (4) we have to know how to compute the Jacobian of $\mathcal{F}^{(1)}$. As shown in (Cai and Keyes [2002]) how one can obtain the Jacobian of $\mathcal{F}^{(1)}$, denoted by $\mathcal{J}^{(1)}$, by the following formula:

$$\mathcal{J}^{(1)}(u) = \sum_{i=1}^{N} J_{S_i}^{-1}(u - T_i(u)) \cdot J(u),$$

where $J(u) = DF(u)$ is the Jacobian of the original function F and $J_{S_i}(u) = I_{S_i} J(u) I_{S_i}$. In practice since $T_i(u)$ converges to zero, we can assumed that a good approximation of the Jacobian is given by

$$\mathcal{J}^{(1)}(u) \approx \sum_{i=1}^{N} J_{S_i}^{-1}(u) J(u),$$

which is, as a matter of fact, the original Jacobian matrix preconditioned by a one-level additive Schwarz method, thus it should be well-conditioned as long as the number of subdomains is not very large.

2.2 Two two-level methods

Similarly, let $S^c = (1, \ldots, n^c)$ be an index set for the coarse system, and we assume that $\{S_1^c, \ldots, S_N^c\}$ is a partition. For simplicity, we partition the fine and the coarse systems into the same number of subsets. Also for simplicity, in our parallel implementation, we allocate the subsystems corresponding to the index sets S_i and S_i^c to the same processor. We introduce the subdomain fine to coarse restriction operator $R_i : S_i \longrightarrow S_i^c$, in the sense that for each vector $v_i \in V_i$, there is a unique vector $v_i^c \in V_i^c$, such that

$$v_i^c = R_i v_i.$$

Assuming the R_is are consistent in the overlapping part of the subdomains, we can 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

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

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

$$F^c(T^c(v)) = R^c F(v). \tag{5}$$

We assume that (5) has a unique solution. Then we define an operator $T_0 : \Re^n \longrightarrow \Re^n$ by

$$T_0(v) = E^c T^c(v). \tag{6}$$

Suppose that T_0 is given as in (6); it is easy to see that $T_0(u_*)$ can be computed without knowing the exact solution u_* itself. In fact, from (5), we have $T_0(u_*) = E^c u_*^c$. Throughout this paper, we assume that the coarse solution u_*^c is given, through a pre-processing step. We can now introduce a new nonlinear function $\Re^n \longrightarrow \Re^n$ by

$$\mathcal{F}^{(2)}(u) = T_0(u) - T_0(u_*) + \sum_{i=1}^{N} T_i(u). \tag{7}$$

Algorithm 2.2 (ASPIN(2)) *Obtain an approximate solution of* u_* *by solving*

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

using the inexact Newton method with $u^{(0)}$ *as the initial guess (Cai et al. [2002]).*

In this paper, we propose a slight modification of the above algorithm in the selection of the initial guess. The algorithm takes the following form.

Algorithm 2.3 (ASPIN(2')) *Obtain an approximate solution of* u_* *by solving*

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

using the inexact Newton method with $T_0(u_*)$ *as the initial guess.*

No additional cost is needed to switch from the original initial guess $u^{(0)}$ to $T_0(u_*)$ since the vector $T_0(u_*)$ is needed anyway in the nonlinear function evaluation.

3 Numerical studies

We next present some numerical results on a two-dimensional lid driven cavity flow problem (Hirsch [1990]). Consider the velocity-vorticity formulation of the incompressible Navier-Stokes equations on the unit square $\Omega = (0,1) \times (0,1)$:

$$\begin{cases} -\Delta u - \dfrac{\partial \omega}{\partial y} & = 0 \\ -\Delta v + \dfrac{\partial \omega}{\partial x} & = 0 \\ -\dfrac{1}{Re}\Delta \omega + u\dfrac{\partial \omega}{\partial x} + v\dfrac{\partial \omega}{\partial y} & = 0, \end{cases} \tag{8}$$

where Re is the Reynolds number, $(u,\ v)$ is the velocity and ω is the vorticity. The boundary conditions are: $u = v = 0$ for bottom, left and right, and $u = 1$, $v = 0$ for top. The boundary condition on ω is given by its definition: $\omega(x,y) = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$. The usual uniform mesh 5-point finite difference approximation is used to discretize the boundary value problem. Upwinding is used for the first derivative terms and central differencing for the second derivative terms. To obtain a nonlinear algebraic system of equations F, we use natural ordering inside each subdomain, and at each mesh point we arrange the unknowns 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 grids. The coarse to fine interpolation is defined using the coarse grid bilinear finite element basis functions. The implementation is done using PETSc (Balay et al. [2002]), and the results are obtained on an IBM SP supercomputer. Double precision is used throughout the computations. The

Table 1. Varying Reynolds numbers. Fine mesh size 128×128, coarse mesh size 32×32, number of processors 16.

	Reynolds number	global nonlinear iterations	global linear iterations	average linear iteration per nonlinear step
ASPIN(1)	10^1	3	112	37
	10^2	4	162	40
	10^3	7	216	30
	10^4	6	156	26
ASPIN(2)	10^1	4	38	9
	10^2	6	89	14
	10^3	7	99	14
	10^4	22	9517	432
ASPIN(2')	10^1	3	28	9
	10^2	4	51	12
	10^3	4	48	12
	10^4	3	40	13

initial guess $u^{(0)}$ is zero for u, v and ω in ASPIN(1) and ASPIN(2). 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 are solved by a Newton-Krylov-Schwarz method based on the same mesh partition. The global 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. 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: $\|F_{S_i}(g_{i,l}^{(k)})\| \leq 10^{-3}\|F_{S_i}(g_{i,0}^{(k)})\|$.

We first compare the three ASPIN algorithms for different Reynolds numbers. In Tables 1, 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 more difficult to solve with the regular inexact Newton method (Cai and Keyes [2002]). However, as shown in Table 1, the numbers of linear and nonlinear iterations of ASPIN(2') are not very sensitive to the increase of Re.

In Table 2, we test the algorithms with different level of overlaps in the Schwarz preconditioner. It is quite interesting to see that ASPIN(2') is not sensitive to this parameter, which is a bit surprising.

Table 2. Varying the overlapping size. Reynolds number = 10^3. Fine mesh size 128×128, coarse mesh size 32×32, number of processors 16.

	overlap	global nonlinear iterations	global linear iterations	average linear iteration per nonlinear step
ASPIN(1)	1	7	216	30
	2	6	141	23
	4	6	112	18
ASPIN(2)	1	8	167	20
	2	8	122	15
	4	7	100	14
ASPIN(2')	1	5	62	12
	2	4	46	11
	4	4	45	11

To use the two-level algorithms on large number of processors and for large fine meshes, the coarse grid size has to be sufficiently fine. This leads to some difficult coarse grid nonlinear systems to solve. Although the coarse problems are, in general, easier to solve than the fine grid problem but sometimes NKS may not be good enough to converge the coarse nonlinear iterations. In the next set of experiments we use an ASPIN(1) coarse solver. That is instead of solving problem (2) by NKS we solve it using ASPIN(1). The stopping criteria for the coarse solver and the fine solver are the same.

In Table 3 we present some experiments for ASPIN(2') on an 1024×1024 mesh, Reynolds number 10^4, and the coarse mesh is 64×64.

Table 3. ASPIN(2'). Varying the number of processors. Fine mesh size 1024×1024, coarse mesh size 64×64, Reynolds number = 10^4.

processors #	nonlinear iterations	average linear iter. per nonlin. step	total CPU time (sec)
32	7	34	1377
64	8	32	653
128	8	39	418
256	10	44	374

The results show that both the number of linear and nonlinear iterations are nearly independent of the number of processors, which is the same as the number of subdomains. In terms of the CPU time, the algorithm scales well for up to 128 processors. The CPU time for 256 processors is only slightly smaller than for 128 processors. We suspect that for large number of processors the ASPIN(1) coarse solver becomes less effective. Thus we measure the computational time the coarse solver takes, and the results are summarized in Table 4 in which we also report the percentage of time spent on the coarse

Table 4. Performance of the ASPIN(1) based coarse solver. Fine grid 1024×1024, Reynolds number $= 10^4$.

processors #	coarse grid	total CPU time (sec)	coarse CPU time (sec)	percentage
32	64×64	1377	20	1.4 %
64	64×64	653	19	2.9%
128	64×64	418	16	3.8%
256	64×64	374	35	9.3%
256	128×128	361	155	42.9%

solver. It seems that the ASPIN(1) based coarse solver takes much more computing time for large number of processors. Our current approach works fine for modest number of processors, but for larger number of processors a more efficient parallel coarse solver is definitely needed.

References

Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matt Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2002.

Xiao-Chuan Cai, David E. Keyes, and Leszek Marcinkowski. Non-linear additive Schwarz preconditioners and application in computational fluid dynamics. *Internat. J. Numer. Methods Fluids*, 40(12):1463–1470, 2002.

Xiao-Chuan Cai, David E. Keyes, and David P. Young. A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flow. In N. Debit, M. Garbey, R. Hoppe, J. Periaux, D. Keyes, and Yu. Kuznetsov, editors, *13th Int. Conference on Domain Decomposition Methods, (Lyon, France, October 9-12 2000)*, pages 343–350. DDM.org, Augsburg, 2000.

Xiao-Chuan Cai and David E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.*, 24(1):183–200, 2002.

Maksymilian Dryja and Wolfgang Hackbusch. On the nonlinear domain decomposition method. *BIT*, 37(2):296–311, 1997.

Stanley C. Eisenstat and Homer F. Walker. Globally convergent inexact Newton methods. *SIAM J. Optim.*, 4(2):393–422, 1994.

Charles Hirsch. *Numerical computation of internal and external Flows: computational methods for inviscid and viscous flows*. John Wiley & Sons, New York, 1990.

Feng-Nan Hwang and Xiao-Chuan Cai. A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations. Preprint, Department of Computer Science, University of Colorado at Boulder, 2003.