FLD 404
pp: 1-8 (col.fig.: nil)
PROD. TYPE: COM
ED: Uma
PAGN: Usha.K – SCAN: Nil

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:000–000 (DOI: 10.1002/fld.404)

Non-linear additive Schwarz preconditioners and application in computational fluid dynamics

Xiao-Chuan Cai[1,*,†], David E. Keyes[2,‡] and Leszek Marcinkowski[3,§]

[1]*Department of Computer Science, University of Colorado, Boulder, CO 80309-0430, U.S.A.*
[2]*Department of Mathematics & Statistics, Old Dominion University, Norfolk, VA 23529-0077, U.S.A.*
[3]*Department of Mathematics, Warsaw University, Poland*

SUMMARY

The focus of this paper is on the numerical solution of large sparse non-linear systems of algebraic equations on parallel computers. Such non-linear systems often arise from the discretization of non-linear 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 non-linearities in the algebraic system become unbalanced. In this paper, we study some preconditioned inexact Newton algorithms, including the single-level and multilevel non-linear additive Schwarz preconditioners. Some results for solving the high Reynolds number incompressible Navier–Stokes equations are reported. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: non-linear preconditioning; inexact Newton methods; Krylov subspace methods; non-linear additive Schwarz; multilevel methods; domain decomposition; non-linear equations; parallel computing; incompressible flows

1. INTRODUCTION

Newton's method is one of the most popular techniques for solving large non-linear systems of equations in engineering applications due to the fact that the method is easy to implement,

---

[*] Correspondence to: Xiao-Chuan Cai, Department of Computer Science, University of Colorado at Boulder, CB 430, Boulder, Colorado 80309-0430, U.S.A.
[†] E-mail: cai@cs.colorado.edu
[‡] E-mail: keyes@icase.edu
[§] E-mail: lmarcin@mimuw.edu.pl

Contract/grant sponsor: NSF; contract/grant number: ACI-0083004, ACI-0072089, ACI-0112930
Contract/grant sponsor: LLNL; contract/grant number: B218772
Contract/grant sponsor: U.S. DOE; contract/grant number: DE-FC02-0IER25479
Contract/grant sponsor: NASA; contract/grant number: NASI-19480
Contract/grant sponsor: LLNL; contract/grant number: B347882
Contract/grant sponsor: U.S DOE; contract/grant number: DE-FC02-01ER25476
Contract/grant sponsor: NSF; contract/grant number: ACI-0072089
Contract/grant sponsor: LLNL; contract/grant number: B218772

Received 1 November 2001
Revised 12 June 2002

especially in its Jacobian-free Newton–Krylov form, 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 proportional to the relative non-linearity of the function; i.e. as the relative non-linearity increases the domain of convergence shrinks, and as a result, finding a good starting point becomes very difficult [1]. Many globalization techniques have been developed in order to find a good starting point, such as the line search and trust region methods [1], continuation methods [2], mesh sequencing methods [3], etc. In this paper, we present a different approach that increases the domain of convergence of Newton's method by reducing the non-linearity of the function.

Consider a given non-linear function $F: \Re^n \rightarrow \Re^n$. 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 $F = (F_1, \ldots, F_n)^{\mathrm{T}}$, $F_i = F_i(u_1, \ldots, u_n)$, and $u = (u_1, \ldots, u_n)^{\mathrm{T}}$. Inexact Newton algorithms (IN) [1, 4] are commonly used for solving such systems. In this paper, we work in the framework of non-linearly preconditioned inexact Newton algorithms (PIN), recently introduced in Reference [5]. In other words, we try to find the solution u_* of Equation (1) by solving an equivalent system of non-linear equations

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

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

2. SINGLE-LEVEL NON-LINEAR ADDITIVE SCHWARZ PRECONDITIONING

In this section, we describe a non-linear preconditioner based on the additive Schwarz method [6, 7]. 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

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

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

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

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

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

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 non-linear function as

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

We next define the major novel feature of the algorithm, namely the non-linearly preconditioned function. For any given $v\in\Re^n$, define $T_i(v)\in V_i$ as the solution of the following subspace non-linear system:

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

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

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

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

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

with an inexact Newton method using $u^{(0)}$ as the initial guess. As shown in Reference [5], ASPIN(1) is non-linearly 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 non-linearly and linearly.

3. TWO-LEVEL NON-LINEAR ADDITIVE SCHWARZ PRECONDITIONING

In this section, we describe a parallel non-linear preconditioner based on the two-level additive Schwarz method [6, 7]. The focus is on the construction of the coarse space operator. We refer to the non-linear algebraic system (1) as the *fine system* which has n unknowns and n equations. We also need a *coarse system*,

$$F^{\mathrm{c}}(u_*^{\mathrm{c}}) = 0 \tag{5}$$

which is a non-linear algebraic system with n^{c} unknowns and n^{c} equations. The coarse and fine functions $F^{\mathrm{c}}(u^{\mathrm{c}})$ and $F(u)$ approximate each other in a certain sense.

We next define the grid transfer operators. Note that our definitions are quite general; for example, the coarse and fine grids need not be nested. Let $S^{\mathrm{c}}=(1,\ldots,n^{\mathrm{c}})$ be an index set, i.e. one integer for each unknown of the coarse system, and assume that $S_1^{\mathrm{c}},\ldots,S_N^{\mathrm{c}}$ is a partition of S^{c} in the sense that $\bigcup_{i=1}^{N} S_i^{\mathrm{c}} = S^{\mathrm{c}}$. For simplicity, we partition the fine and the coarse systems into the same number of subsets. Also for simplicity, in our parallel implementation, we allocate the subsystems corresponding to the index sets S_i, and S_i^c to the same processor. We define the subdomain fine-to-coarse restriction operator as $R_i: S_i \rightarrow S_i^c$, 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

$$v_i^c = R_i v_i$$

where R_i is a n_i by n_i^c matrix. In a similar way, we can introduce an extension operator from the coarse subspace S_i^c to the fine subspace S_i, $E_i: S_i \rightarrow S_i^c$. In practice, E_i is usually taken as the transpose of the matrix R_i. Even though the subsets S_i^c and S_j^c may overlap each other, the restriction operators R_i and R_j are consistent in the sense that for any $v \in \mathfrak{R}^n$, if $k \in S_i^c \cap S_j^c$, 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 $R^c: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n^c}$ as follows: For any $v \in \mathfrak{R}^n$, the kth component of $R^c v$ is defined as

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

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

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

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

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

with an inexact Newton method using $u^{(0)}$ as the initial guess. A more complete description of ASPIN(2) can be found in Reference [8].

4. A BRIEF REVIEW OF INEXACT NEWTON METHODS

Consider a non-linear 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

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

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 [4, 9]. $\lambda^{(k)}$ is another scalar that determines how far one should go in the selected inexact Newton direction [1]. 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 non-linear equations that have unbalanced non-linearities [10]. The step length $\lambda^{(k)}$ is often determined by the components with the strongest non-linearities, and this may lead to an extended period of stagnation in the non-linear residual curve [11, 3].

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

$$\tfrac{1}{2}\,\|\mathscr{F}\|^2$$

which, by design, has more balanced non-linearity than $\tfrac{1}{2}\,\|F\|^2$.

5. A DRIVEN CAVITY FLOW PROBLEM

In this section, we present some numerical results for 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 $\Omega = (0,1)\times(0,1)$,

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

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 non-linear 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 grids. 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 [13], and the results are obtained on a cluster of workstations. Double precision is used throughout the computations. For both the coarse and the fine grid problems, 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

$$\|\mathscr{F}(u^{(k)})\| \leqslant 10^{-10} \|\mathscr{F}(u^{(0)})\|$$

The same stopping condition is used for the coarse grid non-linear systems, which are solved by a Newton–Krylov–Schwarz method.

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

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

is satisfied. We remark that, unlike the Jacobian matrix of F, the Jacobian matrix $\mathscr{F}'$ is usually not sparse and cannot be computed explicitly. Following the techniques developed in Reference [5], we approximate $\mathscr{F}'$ 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 linear preconditioning when solving the Jacobian problems.

At the kth global non-linear iteration, non-linear subsystems

$$F_{S_i}(u^{(k)} - g_i^{(k)}) = 0$$

must 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 non-linear iteration in subdomain S_i is stopped if the following condition is satisfied:

$$\|F_{S_i}(g_{i,l}^{(k)})\| \leqslant 10^{-3} \|F_{S_i}(g_{i,0}^{(k)})\|$$

In Tables I and II, we report the total number of global non-linear iterations, the total number of linear iterations, and the average number of linear iterations per non-linear iteration.

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

	Number of processors	Global non-linear iterations	Average linear iteration per non-linear step
ASPIN(1)	$2\times 2 = 4$	6	15
	$4\times 4 = 16$	6	22
ASPIN(2)	$2\times 2 = 4$	5	11
	$4\times 4 = 16$	7	12

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

	Number of processors	Global non-linear iterations	Average linear iteration per non-linear step
ASPIN(1)	$2\times 2 = 4$	10	17
	$4\times 4 = 16$	7	24
ASPIN(2)	$2\times 2 = 4$	8	15
	$4\times 4 = 16$	7	21

For this particular test problem, the non-linearity is determined mostly by the Reynolds number. As *Re* increases the non-linear system becomes increasingly difficult to solve with the standard inexact Newton method [5]. However, as shown in Tables I and II, 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. For example, in Table I, when we increase the number of processors from 4 to 16, the average number of global linear iterations per non-linear step stays nearly the same. We observe, in Table II, that when *Re* is high the size of the coarse grid has to be sufficiently fine in order for the coarse grid problem to be solvable. 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. More experience on this issue will be provided in a forthcoming paper [8].

ACKNOWLEDGEMENTS

Cai was supported in part by NSF under ACI-0083004, ACI-0072089, and ACI-0112930, by LLNL under subcontract B218772, and by the U.S. DOE under cooperative agreement DE-FC02-01ER25479. Keyes was supported in part by NASA under contract NASI-19480, by LLNL under subcontract B347882, and by the U.S. DOE under cooperative agreement DE-FC02-01ER25476. Marcinkowski was supported in part by NSF under ACI-0072089, and by LLNL under subcontract B218772.

REFERENCES

1. Dennis J, Schnabel R. *Numerical Methods for Unconstrained Optimization and Non-linear Equations*. Prentice-Hall: Englewood Cliffs, NJ, 1983.
2. Kelley C, Keyes D. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis* 1998; **35**:508–523.
3. Young D, Mervin R, Bieterman M, Johnson F, Samant S, Bussoletti J. A locally refined rectangular grid finite element method: application to computational fluid dynamics and computational physics. *Journal of Computational Physics* 1991; **92**:1–66.
4. Eisenstat S, Walker H. Globally convergent inexact Newton methods. *SIAM Journal on Optimization* 1994; **4**:393–422.
5. Cai X-C, Keyes D. Non-linearly preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing*, 2002; to appear.
6. Cai X-C, Dryja M. Domain decomposition methods for monotone non-linear elliptic problems. *Contemporary Mathematics* 1994; **180**:21–27.
7. Dryja M, Hackbusch W. On the non-linear domain decomposition method. *BIT* 1997; 296–311.
8. Cai X-C, Keyes D, Marcinkowski L. Two-level non-linear Schwarz preconditioned inexact Newton methods. 2002, in preparation.
9. Eisenstat S, Walker H. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing* 1996; **17**:16–32.
10. Lanzkron P, Rose D, Wilkes J. An analysis of approximate non-linear elimination. *SIAM Journal on Scientific Computing* 1996; **17**:538–559.
11. Cai X-C, Gropp W, Keyes D, Melvin R, Young D. Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation. *SIAM Journal on Scientific Computing* 1998; **19**:246–265.
12. Hirsch C. *Numerical Computation of Internal and External Flows*. Wiley: New York, 1990.
13. Balay S, Gropp W, McInnes L, Smith B. *The Portable*, *Extensible Toolkit for Scientific Computing*, *version 2.1.0*, `http://www.mcs.anl.gov/petsc`, code and documentation, 2001.
14. Cai X-C, Keyes D, Young D. A non-linear additive Schwarz preconditioned inexact Newton method for shocked duct flow. In *Domain Decomposition Methods in Science and Engineering*, Debit N, Garbey M, Hoppe R, Periaux J, Keyes D, Kuznetsov Y (eds). CIMNE, UPS, Barcelone, 2001.
15. Dryja M, Widlund O. Domain decomposition algorithms with small overlap. *SIAM Journal on Scientific Computing* 1994; **15**:604–620.
16. Smith B, Bjørstad P, Gropp W. *Domain Decomposition*: *Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: Cambridge, 1996.