Two-Grid LNKSz for Distributed Control of Unsteady Incompressible Flows

Haijian Yang[1] and Xiao-Chuan Cai[2]

[1] College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, P. R. China, haijianyang@gmail.com
[2] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA, cai@cs.colorado.edu

Summary. The distributed control of unsteady incompressible flows has been the focus of intense research in scientific computing in the past few years. Most of the existing approaches for distributed control problems are based on the so-called reduced space method which is easier to implement but may have convergence issues in some situations. In this paper we investigate some fully coupled parallel two-grid Lagrange-Newton-Krylov-Schwarz (LNKSz) algorithms for the implicit solution of distributed control problems. In the full space approach we couple the control variables, the state variables and the adjoint variables in a single large system of nonlinear equations. Numerical experiments are presented to show the efficiency and scalability of the algorithm on supercomputers with more than one thousand processors.

1 Introduction

Flow optimal control problems have many important applications in science and engineering and many attempts have been made in the past few years to mathematically understand and numerically solve flow control problems in various forms; see e.g., [3, 6]. Popular approaches for solving unsteady flow control problems are explicit or semi-implicit methods, both are limited by a Courant-Friedrichs-Lewy (CFL) condition. Recently, the class of full space Lagrange-Newton-Krylov-Schwarz (LNKSz) algorithms was introduced for solving the steady state flow control problem [4, 5]. The methods include two parts: a Lagrange-Newton method for the nonlinear system obtained from the optimization problem and a Krylov subspace method for the Jacobian system arising from the Newton method. In this paper we propose a class of fully coupled parallel two-grid Lagrange-Newton-Krylov-Schwarz (LNKSz) algorithms for the distributed control of unsteady incompressible flows. Since we use a fully implicit scheme, the CFL condition can be completely relaxed. We show numerically that the proposed LNKSz is stable and converges well with relatively large times steps, and it is robust with respect to some of the physical parameters, such as the Reynolds number.

The rest of the paper is organized as follows. In Sect. 2, we present the unsteady distributed control problems and introduce a fully implicit discretization scheme.

R. Bank et al. (eds.), *Domain Decomposition Methods in Science and Engineering XX*, Lecture Notes in Computational Science and Engineering 91,
DOI 10.1007/978-3-642-35275-1__55, © Springer-Verlag Berlin Heidelberg 2013

Section 3 includes the main components and features of LNKSz. Some numerical results are given in Sect. 4. We end the paper with some concluding remarks in Sect. 5.

2 Mathematical Model and Discretization

We consider the two-dimensional unsteady incompressible Navier-Stokes equations in the velocity-vorticity formulation:

$$\begin{cases} -\Delta v_1 - \dfrac{\partial \omega}{\partial y} = 0 \ \text{ in } [0,T]\times\Omega, \\ -\Delta v_2 + \dfrac{\partial \omega}{\partial x} = 0 \ \text{ in } [0,T]\times\Omega, \\ \dfrac{\partial \omega}{\partial t} - \dfrac{1}{Re}\Delta\omega + v_1\dfrac{\partial \omega}{\partial x} + v_2\dfrac{\partial \omega}{\partial y} - \operatorname{curl}\mathbf{f} = 0 \ \text{ in } [0,T]\times\Omega, \end{cases} \tag{1}$$

where Ω is the computational domain and $[0,T]$ is the time interval. In the above equations the velocity field $\mathbf{v} = (v_1, v_2)$ and the vorticity ω are the state variables, $\mathbf{f} = (f_1, f_2)$ is the external force, $\operatorname{curl}\mathbf{f} = -\partial f_1/\partial y + \partial f_2/\partial x$, and Re is the Reynolds number.

In the distributed control problem we try to find an external force $\mathbf{f}$ over the control domain $\Omega_f \subseteq \Omega$ in order to achieve the goal

$$\min \mathscr{F}(\mathbf{v},\omega,\mathbf{f}) = \frac{1}{2}\int_0^T \mathscr{G}(\mathbf{v},\omega)\,dt + \frac{\gamma}{2}\int_0^T \int_{\Omega_f} \|\mathbf{f}\|_2^2 \, d\Omega \, dt \tag{2}$$

subject to the constraints (1) with some initial and boundary conditions. Here, $\mathscr{G}(\mathbf{v},\omega)$ is the objective function of the optimal control problem, $\gamma > 0$ is a regularization parameter used to restrict the magnitude of the external force so that it is not unrealistically large.

For solving unsteady distributed control problems, it typically requires a combination of a discretization in space and time with an optimization method. In this paper we follow the discretize-then-optimize approach with a finite difference method for the space discretization and a second-order backward differentiation formula for the time discretization. The original full-time-interval problem is too expensive to solve even on the latest supercomputers, we therefore replace it by a sequence of suboptimal problems, which are similar to the original problem but only defined on the time interval $[t^{(k-1)}, t^{(k)}]$, $k = 1, 2, \ldots, k_{\max}$, with $t^{(0)} = 0$ and $t^{(k_{\max})} = T$. Let $\mathbf{x} = (\mathbf{v}, \omega, \mathbf{f})$. Then on each time interval we write the discrete suboptimzation problem as follows:

$$\begin{cases} \min \ \mathscr{F}_h^{(k)}(\mathbf{x}) \\ \text{s.t. } \mathbf{C}_h^{(k)}(\mathbf{x}) = \mathbf{0}, \end{cases} \tag{3}$$

where $\mathscr{F}_h^{(k)}(\mathbf{x})$ is the restriction of $\mathscr{F}$ on the interval $[t^{(k-1)}, t^{(k)}]$, and $\mathbf{C}_h^{(k)}(\mathbf{x})$ are the constraints defined on the time interval $[t^{(k-1)}, t^{(k)}]$.

By introducing the Lagrange multipliers λ with respect to the state and control variables, we define the following Lagrangian functional

$$\mathscr{L}^{(k)}(\mathbf{x},\lambda) \equiv \mathscr{F}_h^{(k)}(\mathbf{x}) + \left(\lambda, \mathbf{C}_h^{(k)}(\mathbf{x})\right). \tag{4}$$

Let $X \equiv (\mathbf{x},\lambda)$. Then, for $k = 1,2,\ldots,k_{\max}$, the KKT system obtained by differentiating (4) becomes

$$G^{(k)}(X) = \begin{pmatrix} \nabla_{\mathbf{x}} \mathscr{L}^{(k)}(\mathbf{x},\lambda) \\ \nabla_{\lambda} \mathscr{L}^{(k)}(\mathbf{x},\lambda) \end{pmatrix} = 0. \tag{5}$$

The optimality system (5) is a large, nonlinear, coupled, and muti-components system. Moreover, the corresponding Jacobian matrix is indefinite and very ill-conditioned. Hence, a good preconditioner is essential to solve the optimality system efficiently.

3 Two-Grid Newton Method and Schwarz Preconditioners

The class of full space LNKSz method includes the following steps: the Lagrangian functional is formed and differentiated to obtain the KKT system; then the inexact Newton method with line search is applied; and at each Newton iteration the linear system is solved with a one-level or two-level Schwarz preconditioned Krylov subspace method. We refer to LNKSz combined with the one-level (two-level) Schwarz preconditioner as one-level (two-level) LNKSz method.

When using Newton's method to solve the nonlinear system (5) on a grid, one of the major problems is the deterioration of the convergence rate when the grid is refined, specially for the first time step, since in this case the initial guess is not good enough for the Newton iterations. After many experiments, we find that a solution to the problem is "grid-sequencing", which is quite effective in keeping the number of nonlinear iterations small. In order to use grid-sequencing, we assume there are two grids covering Ω, a coarse grid of size H and a fine grid of size h. We first use the one-level method to solve the nonlinear problem on the coarse grid with the initial guess obtained as a restriction of the fine grid solution from the previous timestep. Of course, at the first time step, we choose the initial condition as the initial guess. Then, we interpolate the solution to the fine grid and use it as an initial guess for the nonlinear problem on the fine grid. We refer to this LNKSz method combined with the grid-sequencing technique as the two-grid LNKSz method in which the same coarse grid is also used to build the two-level Schwarz preconditioner for solving the Jacobian problem.

We assume that Ω is covered by a non-overlapping and an overlapping partition as in [2]. Let J be the Jacobian matrix of the nonlinear problem (5) on the fine grid and let R_i^{δ} and R_i^0 be the restriction operator from Ω to its overlapping and non-overlapping subdomains, respectively. Here δ is the size of the overlap. Then the one-level restricted additive Schwarz (RAS) preconditioner [2] is defined as

$$M_{RAS}^{-1} = \sum_{i=1}^{N_p} (R_i^0)^T J_i^{-1} R_i^\delta. \tag{6}$$

with $J_i = R_i^\delta \; J \; (R_i^\delta)^T$ and N_p is the number of subdomains, which is the same as the number of processors. Let J_c be the Jacobian matrix on the coarse grid and I_h^H a restriction operator from the fine grid to the coarse grid. Then a multiplicative type two-level Schwarz preconditioner [8, 9] is defined as

$$M^{-1} = \Big(I - (I - M_{RAS}^{-1}J)(I - M_c^{-1}J)(I - M_{RAS}^{-1}J)\Big)J^{-1} \tag{7}$$

with $M_c^{-1} = (I_h^H)^T J_c^{-1} I_h^H$ and I is the identity matrix.

4 Numerical Experiments

Our algorithms are implemented based on the Portable Extensible Toolkit for Scientific computing (PETSc) [1]. All computations are performed on an IBM BlueGene/L supercomputer.

In the following, we describe a backward-facing step flow control problem [7]. Let $\Omega = (0,6) \times (0,1)$, $\Omega_f = (0,1) \times (0,0.5)$, $T = 1$, Γ be the boundary of the domain Ω, $\Gamma_2 = \{(x,y) \in \Gamma : 0 < y < 1, x = 6\}$, $\Gamma_4 = \{(x,y) \in \Gamma : 0 < y < 1, x = 0\}$, and $\Gamma_{4,a} = \{(x,y) \in \Gamma_4 : 0.5 \le y < 1\}$. Then the backward-facing step control problem consists of finding $(v_1, v_2, \omega, f_1, f_2)$ such that the minimization

$$\min \mathscr{F}(\omega, \mathbf{f}) = \frac{1}{2}\int_0^T \int_\Omega \omega^2 \, d\Omega \, dt + \frac{\gamma}{2}\int_0^T \int_{\Omega_f} \|\mathbf{f}\|_2^2 \, d\Omega \, dt \tag{8}$$

is achieved subject to the constraints (1) with the following boundary conditions:

$$\begin{cases} v_1 & = v_{in} \quad \text{on } [0,T] \times \Gamma_{4,a}, \\ v_1 & = v_{out} \quad \text{on } [0,T] \times \Gamma_2, \\ v_1 & = 0 \quad \text{on } [0,T] \times \Gamma_u, \\ v_2 & = 0 \quad \text{on } [0,T] \times \Gamma, \\ \omega + \dfrac{\partial v_1}{\partial y} - \dfrac{\partial v_2}{\partial x} & = 0 \quad \text{on } [0,T] \times \Gamma, \\ \mathbf{v}(0,x,y) - \mathbf{v}_0 & = \mathbf{0} \quad \text{in } \overline{\Omega}, \\ \omega(0,x,y) + \dfrac{\partial v_{0,1}}{\partial y} - \dfrac{\partial v_{0,2}}{\partial x} & = 0 \quad \text{in } \overline{\Omega}, \end{cases} \tag{9}$$

where $\Gamma_u = \Gamma \backslash (\Gamma_{4,a} \cup \Gamma_2)$. At the inflow boundary, a parabolic velocity profile $v_{in} = 8(1-y)(y-\frac{1}{2})cos(t)$ is imposed. At the outflow boundary, $v_{out} = y(1-y)cos(t)$ is applied. The following initial velocity is defined by $\mathbf{v}_0 = (v_{0,1}, v_{0,2})$ with

$$v_{0,1} = \begin{cases} y(1-y) + \dfrac{1}{16}y & \text{if } 0 \le y \le \dfrac{1}{2}, \\ \\ y(1-y) + \dfrac{1}{16}(1-y) & \text{if } \dfrac{1}{2} \le y \le 1, \end{cases}$$

and $v_{0,2}(x,y) = 0$. The parameter $\gamma = 0.1$.

In the experiments, we compare the following algorithms which are introduced in Sect. 3:

- One-level LNKSz: one-level additive Schwarz is used as the Jacobian solve, and inexact Newton is carried out on the fine grid;
- Two-level LNKSz: two-level multiplicative Schwarz is used as the Jacobian solve, and inexact Newton is carried out on the fine grid;
- Two-grid LNKSz: two-level multiplicative Schwarz is used as the Jacobian solve on the fine grid, inexact Newton is used on the coarse grid to generate the initial guess for the inexact Newton on the fine grid.

In all the experiments, all Jacobian matrices are constructed approximately using a multi-colored finite difference method. The size of the coarse grid H is taken as $4h$, where h is the size of the fine grid. GMRES(90) and FGMRES(90) are used to solve the linear system at each Newton step on the coarse and the fine grids, respectively. In the one-level method, the overlapping size is $\delta = 6$. In the two-level and two-grid methods, the overlapping sizes of the coarse grid and the fine grids are $\delta_c = 4$ and $\delta = 6$, respectively. There are several nested iterative procedures in the proposed algorithms, and each requires a proper stopping condition. We use 10^{-10} (10^{-6}) as the absolute (relative) condition for all linear and nonlinear solves, except for the linear coarse solve of the two-level preconditioner, for which we use 10^{-4} (10^{-2}) as the absolute (relative) condition. The subdomain problems are solved with a sparse LU factorization.

Next, we present results for the test problem and discuss some details of the two-grid LNKSz. First, we compare the three methods in Table 1. Note that, the one-level method doesn't converge when $N_p = 1{,}024$, which is caused by the divergence of GMRES. Moreover, we note that: (1) for the linear solver, the number of GMRES iterations for the one-level LNKSz is much larger than that for the two-level and two-grid methods; (2) for the nonlinear solver, the numbers of Newton iterations for the one-level and two-level methods are also larger than that for the two-grid method; and (3) compared with the one-level and two-level methods, the total computing time for the two-grid method is much smaller. When the Reynolds number increases from 200 to 400, for one-level and two-level methods, the average number of Newton iterations and the total computing time become larger. With the help of grid-sequencing, the convergence of the two-grid method is less sensitive to the Reynolds number. Based on the results of Table 1, it is clear that the two-grid method is better than the others.

An important implementation detail to consider in designing two-grid LNKSz is to balance the quality of the initial guess for the fine grid Newton iterations and the computing time on the coarse solver. In Table 2, we present a comparison of the computing time for the two-level and two-grid methods. In this table, we report the total time spent on the Newton iterations at some time steps, the time spent on the Newton iterations on the coarse solver, and the percentage between these two computational costs. We observe that the cost of Newton iterations on the coarse grid is very small compared with the total computational cost. It is important to note that the coarse

Table 1. A comparison of three methods. 768×128 grid, and $\Delta t = 0.1$ (i.e., there are 10 time steps). "N_p" stands for the number of processors which is the same as the number of subdomains, "IN" is the average number of inexact Newton iterations per time step on the fine grid, "RAS" is the average number of RAS preconditioned GMRES iterations per Newton iteration, and "Time" is the total computing time in seconds. "$**$" means the divergence of GMRES.

N_p	Method	IN	RAS	Time	IN	RAS	Time
		Re=200			Re=400		
64	One-level	3.2	165.4	1370.4	3.7	158.9	1557.5
64	Two-level	3.2	20.4	1342.8	3.7	19.2	1528.0
64	Two-grid	2.1	18.7	898.2	2.0	18.0	836.4
256	One-level	3.2	531.3	795.5	3.7	632.9	1052.3
256	Two-level	3.2	27.4	479.9	3.7	27.1	560.1
256	Two-grid	2.1	25.5	317.5	2.0	26.1	313.2
1024	One-level		$**$			$**$	
1024	Two-level	3.2	66.3	314.3	3.7	67.9	376.9
1024	Two-grid	2.1	64.2	208.5	2.0	68.5	209.8

grid has to be sufficiently fine so that the coarse solution has a reasonable accuracy, otherwise, it won't be able to provide a good initial guess for the fine grid nonlinear solver.

Table 2. A comparison of the computing time for the test problem at several different time steps. $Re = 400$, 768×128 grid, and $\Delta t = 0.1$ (i.e., there are 10 time steps). The heading "Timestep(k)" represents the time step k, "Time" is the total time spent on the Newton iterations at the time step k, "Coarse_time" is the time spent on the Newton iterations on the coarse solver at the time step k, and "Percent(%)" is ("Coarse_time"/"Time").

N_p	Timestep(k)	Time	Coarse_time	Percent(%)	Time
			Two-grid		Two-level
64	$k = 1$	110.0	3.87	3.52%	458.9
64	$k = 2$	80.0	2.39	2.99%	117.0
64	$k = 5$	82.5	2.50	3.03%	118.0
64	$k = 10$	84.7	2.51	2.96%	119.0
256	$k = 1$	38.6	1.71	4.43%	172.8
256	$k = 2$	29.7	0.99	3.33%	41.4
256	$k = 5$	30.0	1.04	3.43%	41.6
256	$k = 10$	30.8	1.06	3.44%	42.3
1024	$k = 1$	23.3	1.37	5.88%	115.1
1024	$k = 2$	20.6	0.68	3.30%	28.1
1024	$k = 5$	21.2	0.72	3.39%	28.4
1024	$k = 10$	21.5	0.74	3.44%	30.8

One of the difficulties in the nonlinear solver is the choice of the initial guess. In Fig. 1, we show the nonlinear residual history by using three different methods at the first time step (i.e., $k = 1$). One can see that the nonlinear system is difficult to solve by using one-level or two-level method. In fact, it takes 11 iterations for the one-level or two-level method to converge. By using the two-grid method only three Newton iterations are required to satisfy the desired stopping condition.

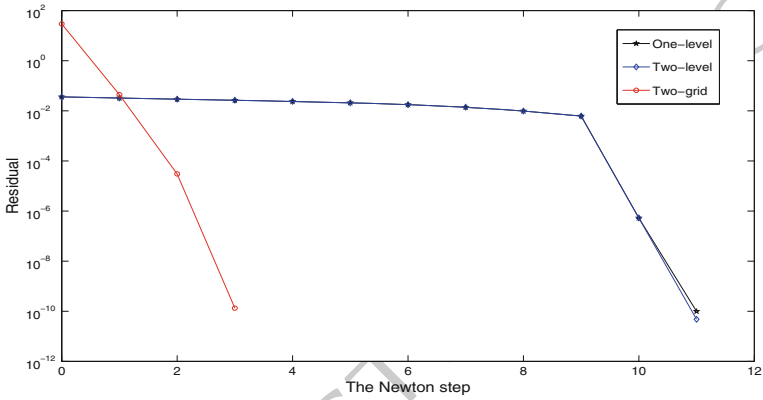

Fig. 1. Nonlinear residual history by using three different methods at the first time step, for $Re = 200$, 768×128 grid and 64 processors, and $\Delta t = 0.1$

5 Conclusions

In this paper, we developed a family of two-grid algorithms for distributed control of unsteady incompressible flows. With the help of the two-grid Newton method and the two-level Schwarz preconditioner, we showed numerically that these strategies provide substantial improvement of the overall method in terms of the total computing time, the number of linear iterations, and the number of Newton iterations, especially when the number of processors is large.

Bibliography

[1] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. *PETSc Users Manual*. Argonne National Laboratory, 2010.

[2] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21:792–797, 1999.

[3] M. Gunzburger. *Perspectives in Flow Control and Optimization*. SIAM, First ed., 2003.

[4] E. Prudencio and X.-C. Cai. Parallel multilevel restricted Schwarz preconditioners with pollution removing for PDE-constrained optimization. *SIAM J. Sci. Comput.*, 29:964–985, 2007.

[5] E. Prudencio, R. Byrd, and X.-C. Cai. Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems. *SIAM J. Sci. Comp.*, 27:1305–1328, 2006.

[6] L. Quartapelle. *Numerical Solution of the Incompressible Navier-Stokes Equations*. International Series of Numerical Mathematics, Birkhäuser Verlag, 1996.

[7] S. S. Ravindran. Numerical approximation of optimal control of unsteady flows using SQP and time decomposition. *Int. J. Numer. Meth. Fluids.*, 45:21–42, 2004.

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

[9] A. Toselli and O. Widlund. *Domain Decomposition Methods-Algorithms and Theory*. Springer-Verlag, Berlin, 2005.