Domain Decomposition Methods for PDE Constrained Optimization Problems*

Ernesto Prudencio, Richard Byrd, and Xiao-Chuan Cai

Department of Computer Science, University of Colorado, Boulder, CO80309, USA
({prudenci,richard,cai}@cs.colorado.edu)

Abstract. Optimization problems constrained by nonlinear partial differential equations have been the focus of intense research in scientific computing lately. Current methods for the parallel numerical solution of such problems involve sequential quadratic programming (SQP), with either reduced or full space approaches. In this paper we propose and investigate a class of parallel full space SQP Lagrange-Newton-Krylov-Schwarz (LNKSz) algorithms. In LNKSz, a Lagrangian functional is formed and differentiated to obtain a Karush-Kuhn-Tucker (KKT) system of nonlinear equations. Inexact Newton method with line search is then applied. At each Newton iteration the linearized KKT system is solved with a Schwarz preconditioned Krylov subspace method. We apply LNKSz to the parallel numerical solution of some boundary control problems of two-dimensional incompressible Navier-Stokes equations. Numerical results are reported for different combinations of Reynolds number, mesh size and number of parallel processors.

1 Introduction

In this paper we describe a general framework for solving optimization problems in interaction with nonlinear partial differential equations (PDEs). The focus is on how to adapt state-of-the-art PDE solvers to the requirements of optimization methods, while allowing for an efficient parallel implementation. Our method treats the differential equations as equality constraints. In order to demonstrate its effectiveness, the problem of optimizing fluid flows modeled by incompressible Navier-Stokes equations on two-dimensional domains is considered. In optimal control problems one usually searches for the best feasible values of the control variables, such as boundary values or external forces that minimize or maximize a certain system behavior, such as turbulence. In this paper, we only consider boundary control problems, which refer to the control of the system through boundary conditions.

* The work of Prudencio and Cai was partially supported by the Department of Energy, DE-FC02-01ER25479, and by the National Science Foundation, CCR-0219190, ACI-0072089 and ACI-0305666. The work of Byrd was supported in part by the National Science Foundation, CCR-0219190, and by Army Research Office contract DAAD19-02-1-0407.

Reduced space SQP methods have been the most widely used SQP approaches for PDE constrained problems until recently since they require much less memory, even though many sub-iterations are needed to converge the outer-iterations and the parallel scalability is less ideal. As more powerful computer systems with lots of memory and many processors become available, full space methods seem to be more appropriate due to their increased scalability. One such method, Lagrange-Newton-Krylov-Schur (LNKSr), was introduced in [2, 3], where four block factorization based preconditioners, as well as globalization techniques and heuristics, are proposed and tested. In this paper we replace the Schur type preconditioner with an overlapping Schwarz method which has a better asymptotical convergence rate and is easier to use as a nonlinear preconditioner [6, 7] for highly nonlinear problems.

The rest of the paper is organized as follows. Section 2 discusses the numerical solution of optimal control problems with equality constraints. Section 3, the core of this paper, presents the full space SQP *Lagrange-Newton-Krylov-Schwarz* (LNKSz) method for the parallel numerical solution of such problems. Section 4 presents a boundary flow control problem, which is then solved in Section 5 with LNKSz. Numerical experiments are performed and analyzed for different combinations of Reynolds number, mesh size and number of parallel processors. Final conclusions are given in Section 6.

2 Optimal control with equality constraints

In this paper we focus on optimal control problems with equality constraints:

$$\begin{cases} \min\limits_{(\mathbf{s},\mathbf{u})\in\mathbf{S}\times\mathbf{U}} \mathcal{F}(\mathbf{s},\mathbf{u}) \\ \quad \text{s.t. } \mathbf{C}(\mathbf{s},\mathbf{u}) = \mathbf{0} \in \mathbf{Y}. \end{cases} \tag{2.1}$$

Here $\mathbf{S}$ and $\mathbf{U}$ are called state and control spaces respectively; the state variables $\mathbf{s}$ represent the state of the system being controlled; the control variables $\mathbf{u}$ represent the means one has to control the system; the objective or cost functional $\mathcal{F} : \mathbf{S} \times \mathbf{U} \to \mathbb{R}$ to be minimized (or maximized) represents the reason why one wants to control the system; constraints $\mathbf{C}(\mathbf{s},\mathbf{u}) = \mathbf{0}$ represent the system behavior and other constraints imposed to state and control variables.

The Lagrangian functional $\mathcal{L} : \mathbf{S} \times \mathbf{U} \times \mathbf{Y}^* \to \mathbb{R}$ associated with (2.1) is defined by

$$\mathcal{L}(\mathbf{s},\mathbf{u},\boldsymbol{\lambda}) \equiv \mathcal{F}(\mathbf{s},\mathbf{u}) + \langle \boldsymbol{\lambda}, \mathbf{C}(\mathbf{s},\mathbf{u})\rangle_{\mathbf{Y}}, \quad \forall\ (\mathbf{s},\mathbf{u},\boldsymbol{\lambda}) \in \mathbf{S}\times\mathbf{U}\times\mathbf{Y}^*, \tag{2.2}$$

where $\mathbf{Y}^*$ is the adjoint space of $\mathbf{Y}$, $\langle\cdot,\cdot\rangle_{\mathbf{Y}}$ denotes the duality pairing and variables $\boldsymbol{\lambda}$ are called Lagrange multipliers or adjoint variables.

When the constraints are PDEs over a domain Ω, the discretization necessary for the solution of (2.1) can occur at two different points of the logical development of an algorithm. In the first case one demonstrates that, if $(\hat{\mathbf{s}},\hat{\mathbf{u}})$ is a (local) solution of (2.1) then there exist Lagrange multipliers $\hat{\boldsymbol{\lambda}}$ such that

$(\hat{\mathbf{s}}, \hat{\mathbf{u}}, \hat{\boldsymbol{\lambda}})$ is a critical point of $\mathcal{L}$. So, under sufficient smoothness assumptions, one obtains, as necessary condition for a solution, a system of equations, called Karush-Kuhn-Tucker (KKT) or optimality system, which is then discretized, generating a finite dimensional system of nonlinear equations [14, 15, 18, 24].

In the second case one begins by creating a mesh Ω_h of characteristic size $h > 0$ and then discretizing problem (2.1), obtaining a finite dimensional equality constrained optimization problem with $\mathbf{S} = \mathbb{R}^{n_{s,h}}$, $\mathbf{U} = \mathbb{R}^{n_{u,h}}$ and $\mathbf{Y} = \mathbb{R}^{m_h} = \mathbf{Y}^*$. Under sufficient smoothness conditions the KKT system becomes $\nabla \mathcal{L}_h(\hat{\mathbf{s}}, \hat{\mathbf{u}}, \hat{\boldsymbol{\lambda}}) = \mathbf{0} \in \mathbb{R}^{n_{s,h}+n_{u,h}+m_h}$. The theory for finite dimensional constrained optimization problems guarantees, under appropriate assumptions, the existence of such Lagrange multipliers $\hat{\boldsymbol{\lambda}}$. It should be pointed out that the discrete KKT systems from both approaches are *not* necessarily the same.

From now on, we only work with the second approach. For simplicity, let us omit the symbols "h" and "$\hat{\ }$", and use the notations $N \equiv n_s + n_u + m$ and $\mathbf{X} \equiv (\mathbf{x}, \boldsymbol{\lambda}) \equiv (\mathbf{s}, \mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^N$. The KKT system becomes

$$\mathbf{F}(\mathbf{X}) \equiv \begin{pmatrix} \nabla_{\mathbf{x}} \mathcal{L} \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L} \end{pmatrix} = \begin{pmatrix} \nabla \mathcal{F} + [\boldsymbol{\nabla} \mathbf{C}]^T \boldsymbol{\lambda} \\ \mathbf{C}(\mathbf{s}, \mathbf{u}) \end{pmatrix} = \begin{pmatrix} \nabla_{\mathbf{s}} \mathcal{F} + [\boldsymbol{\nabla}_{\mathbf{s}} \mathbf{C}]^T \boldsymbol{\lambda} \\ \nabla_{\mathbf{u}} \mathcal{F} + [\boldsymbol{\nabla}_{\mathbf{u}} \mathbf{C}]^T \boldsymbol{\lambda} \\ \mathbf{C}(\mathbf{s}, \mathbf{u}) \end{pmatrix} = \mathbf{0}, \quad (2.3)$$

where $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$, $\nabla_{\mathbf{x}} \mathcal{L}$ denotes the gradient of $\mathcal{L}$ w.r.t. state and control variables, with similar meaning holding for $\nabla_{\boldsymbol{\lambda}} \mathcal{L}$, $\nabla_{\mathbf{s}} \mathcal{F}$ and $\nabla_{\mathbf{u}} \mathcal{F}$, $\boldsymbol{\nabla} \mathbf{C}$ denotes the Jacobian of $\mathbf{C}$ and $\boldsymbol{\nabla}_{\mathbf{s}} \mathbf{C}$ and $\boldsymbol{\nabla}_{\mathbf{u}} \mathbf{C}$ denote the Jacobian of $\mathbf{C}$ w.r.t. state and control variables, respectively. In (2.3), we refer to the first equation as the adjoint equation, the second as the control equation, the third as the state equation or the forward problem, and $\boldsymbol{\nabla}_{\mathbf{s}} \mathbf{C}$ as the linearized forward operator.

System (2.3) can be solved with an inexact Newton method [12, 13]. Given an initial guess $\mathbf{X}^{(0)}$, at each iteration $k = 0, 1, 2, \ldots$ an approximate solution

$$\mathbf{p}^{(k)} \equiv \left(\mathbf{p}_{\mathbf{x}}^{(k)}, \mathbf{p}_{\boldsymbol{\lambda}}^{(k)} \right) \equiv \left(\mathbf{p}_{\mathbf{s}}^{(k)}, \mathbf{p}_{\mathbf{u}}^{(k)}, \mathbf{p}_{\boldsymbol{\lambda}}^{(k)} \right)$$

of the linearized KKT system

$$\left[\mathbf{K}^{(k)} \right] \ \mathbf{p}^{(k)} \ = \ - \mathbf{F}^{(k)} \quad (2.4)$$

is computed, where $\mathbf{K}^{(k)} = \boldsymbol{\nabla} \mathbf{F}(\mathbf{X}^{(k)})$ and $\mathbf{F}^{(k)} = \mathbf{F}(\mathbf{X}^{(k)})$. The KKT matrix $\mathbf{K}^{(k)}$ is the transpose of the Hessian of the Lagrangian $\mathcal{L}$, is symmetric indefinite under sufficient smoothness assumptions and can be computed by a finite difference approximation. If $\boldsymbol{\nabla} \mathbf{C}^{(k)}$ has full rank and $\boldsymbol{\nabla}_{\mathbf{xx}} \mathcal{L}^{(k)}$ is positive definite in the tangent space of the constraints (i.e., $\mathbf{d}^T [\boldsymbol{\nabla}_{\mathbf{xx}} \mathcal{L}^{(k)}] \mathbf{d} > 0$ for all $\mathbf{d} \neq \mathbf{0}$ such that $[\boldsymbol{\nabla} \mathbf{C}^{(k)}] \mathbf{d} = \mathbf{0}$), then we can interpret the solution $(\mathbf{p}_{\mathbf{x}}^{(k)}, \mathbf{p}_{\boldsymbol{\lambda}}^{(k)})$ of (2.4) as being the *unique* solution and respective Lagrange multipliers of

$$\begin{cases} \min\limits_{\mathbf{p_x} \in \mathbf{S} \times \mathbf{U}} \ \frac{1}{2} \mathbf{p}_{\mathbf{x}}^T [\boldsymbol{\nabla}_{\mathbf{xx}} \mathcal{L}^{(k)}] \mathbf{p}_{\mathbf{x}} + \left(\nabla_{\mathbf{x}} \mathcal{L}^{(k)} \right)^T \mathbf{p}_{\mathbf{x}} \\ \quad \text{s.t.} \ [\boldsymbol{\nabla} \mathbf{C}^{(k)}] \mathbf{p}_{\mathbf{x}} + \mathbf{C}^{(k)} = \mathbf{0} \in \mathbf{Y}. \end{cases}$$

This interpretation justifies the use of terminology *sequential quadratic programming* (SQP) for methods involving (2.4), [19].

After approximately solving (2.4), one may use a globalization method like *line search* or *trust region*. In this study we focus on a line search approach, where the next iterate is $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}$ and the step length $\alpha^{(k)}$ is selected by backtracking until the sufficient decrease condition

$$\phi(\mathbf{X}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}) \leqslant \phi(\mathbf{X}^{(k)}) + \alpha^{(k)} c_1 \left(\nabla\phi(\mathbf{X}^{(k)})\right)^T \mathbf{p}^{(k)} \tag{2.5}$$

is satisfied. Here ϕ is a merit function and c_1 is a constant satisfying $0 < c_1 < 1/2$. For constrained optimization, merit functions such as l_1 or augmented Lagrangian are most commonly used. In contrast to the standard merit function $\|\mathbf{F}(\mathbf{X})\|_2^2/2$, which is commonly used for systems of nonlinear equations, these merit functions try to balance the sometimes conflicting goals of reducing the objective function and satisfying the constraints [19]. We use the augmented Lagrangian $\phi_{AL} : \mathbb{R}^N \to \mathbb{R}$ in our experiments in this paper. Given a penalty parameter $\rho > 0$, it is defined by

$$\phi_{AL}(\mathbf{X};\rho) = \mathcal{L}(\mathbf{s},\mathbf{u},\boldsymbol{\lambda}) + \frac{\rho}{2}\,\|\mathbf{C}(\mathbf{s},\mathbf{u})\|_2^2 \quad \forall\ \mathbf{X} = (\mathbf{s},\mathbf{u},\boldsymbol{\lambda}) \in \mathbf{S}\times\mathbf{U}\times\mathbf{Y}.$$

At iteration k, $\rho = \rho^{(k)}$ must be such that we obtain descent directions $\mathbf{p}^{(k)}$, i.e.,

$$(\nabla\phi_{AL}(\mathbf{X}^{(k)};\rho^{(k)}))^T\mathbf{p}^{(k)} = (\nabla\mathcal{L}^{(k)})^T\mathbf{p}^{(k)} + \rho^{(k)}\ {\mathbf{C}^{(k)}}^{\mathbf{T}}[\boldsymbol{\nabla}\mathbf{C}^{(k)}]\mathbf{p}_{\mathbf{x}}^{(k)} < 0. \tag{2.6}$$

Since ${\mathbf{C}^{(k)}}^{\mathbf{T}}[\boldsymbol{\nabla}\mathbf{C}^{(k)}]\mathbf{p}_{\mathbf{x}}^{(k)} = -\|\mathbf{C}^{(k)}\|_2^2$ for an exact step, it is reasonable to expect

$${\mathbf{C}^{(k)}}^{\mathbf{T}}[\boldsymbol{\nabla}\mathbf{C}^{(k)}]\mathbf{p}_{\mathbf{x}}^{(k)} < 0 \tag{2.7}$$

for approximate steps, if $\mathbf{C}^{(k)} \neq \mathbf{0}$ and the tolerances for the Krylov subspace method are small enough and the preconditioner is good enough to guarantee that the Krylov subspace method does not stop by achieving the maximum allowed number of iterations. If (2.7) does not hold we can continue the Krylov iterations, with eventual smaller tolerances, until it does. Once (2.7) holds, we then use a fairly common strategy where we demand $\rho^{(k)}$ to satisfy

$$(\nabla\phi_{AL}(\mathbf{X}^{(k)};\rho^{(k)}))^T\mathbf{p}^{(k)} \leqslant \frac{\rho^{(k)}}{2}\ {\mathbf{C}^{(k)}}^{\mathbf{T}}[\boldsymbol{\nabla}\mathbf{C}^{(k)}]\mathbf{p}_{\mathbf{x}}^{(k)},$$

that is,

$$\rho^{(k)} \geqslant \overline{\rho}^{(k)} = -2\frac{(\nabla\mathcal{L}^{(k)})^T\mathbf{p}^{(k)}}{{\mathbf{C}^{(k)}}^{\mathbf{T}}[\boldsymbol{\nabla}\mathbf{C}^{(k)}]\mathbf{p}_{\mathbf{x}}^{(k)}}.$$

We then choose

$$\rho^{(k)} = \max\{\overline{\rho}^{(k)}, \rho^{(k-1)}\},$$

where $\rho^{(-1)} > 0$ is a given positive value.

However, if $\mathbf{C}^{(k)} = \mathbf{0}$ then there is no way to guarantee a descent direction. This is a fundamental issue with line search methods. Some algorithms handle

this by modifying the Hessian to make it positive definite on the null space of $\nabla \mathbf{C}^{(k)}$, but for problems of the size we are considering there is no efficient way to check positive definiteness.

In all tests described in this paper, (2.7) held for every step generated, as long as we made the absolute Krylov tolerance small enough that at least one Krylov iteration was performed. In addition, it is worth noting that, in each run, the value of $\rho^{(k)}$ became fixed before 60% of the iterations had been made. Thus the heuristic merit parameter updating strategy described above appeared to work well for the examples of this paper.

3 Parallel full space SQP Lagrange-Newton-Krylov-Schwarz

A key element of a successful full space approach is the preconditioning of the Jacobian of the KKT system, which is indefinite and extremely ill-conditioned. A good preconditioner has to be able to substantially reduce the condition number and, at the same time, to provide good scalability, so that the potential of massively parallel computers can be realized. The Schur complement preconditioner used in LNKSr [2, 3] is an *operator-splitting* type technique, in which a sequential block elimination step is needed to form the Schur complement w.r.t. the control variable. In contrast to operator-splitting, Schwarz type preconditioners are *fully coupled* in the sense that all variables are treated equally and the partition is based completely on the physical domain Ω. Because there is no need to eliminate any variables from the system, there is one less sequential step in the preconditioning process. Another advantage of LNKSz method is that it does not demand $m = n_s$. With LNKSz we can, for instance, deal directly with full [17] boundary control problems, where an equation like (4.3) is explicitly added to the constraints.

Schwarz preconditioners can be used in one-level or multi-level approaches and, at each case, with a combination of additive and/or multiplicative algorithms [23]. In this paper we deal with one-level additive algorithms only. Let $\Omega \subset \mathbb{R}^2$ be a bounded open domain on which the control problem is defined. We only consider simple box domains with uniform mesh of characteristic size h here. To obtain the overlapping partition, we first partition the domain into non-overlapping subdomains Ω_j^0, $j = 1, \cdots, N_s$. Then we extend each subdomain Ω_j^0 to a larger subdomain Ω_j^δ, i.e., $\Omega_j^0 \subset \Omega_j^\delta$. Here $\delta > 0$ indicates the size of the overlap. Only simple box decomposition is considered in this paper; i.e., all the subdomains Ω_j^0 and Ω_j^δ are rectangular and made up of integral number of fine mesh cells. For boundary subdomains, we simply cut off the part that is outside Ω. Let $H > 0$ denote the characteristic diameter of subdomains Ω_j. Let N, N_j^0 and N_j^δ denote the number of degrees of freedom associated to Ω, Ω_j^0 and Ω_j^δ, respectively. Let $\mathbf{K}$ be a $N \times N$ matrix of a linear system that needs to be solved during the numerical solution process of the differential problem. For each subdomain Ω_j^0, we define $\mathbf{R}_j^0$ as an $N_j^0 \times N_j^0$ block sub-identity matrix whose diagonal element, $(\mathbf{R}_j^0)_{k,k}$, is either an $d \times d$ identity matrix if the mesh

point $x_k \in \Omega_j^0$ or a $d \times d$ zero matrix if x_k is outside of Ω_j^0, where d indicates the degree of freedom per mesh point. Similarly we introduce a block sub-identity matrix $(\mathbf{R}_j^\delta)_{k,k}$ for each Ω_j^δ. The multiplication of $\mathbf{R}_j^\delta$ with a vector will zero out all of its components outside of Ω_j^δ. We denote by $\mathbf{K}_j$ the subdomain matrix given by

$$\mathbf{K}_j = \mathbf{R}_j^\delta \ \mathbf{K} \ (\mathbf{R}_j^\delta)^T.$$

Let $\mathbf{B}_j^{-1}$ be either a subspace inverse of $\mathbf{K}_j$ or a preconditioner for $\mathbf{K}_j$. The classical one-level additive Schwarz preconditioner $\mathbf{B}_{asm}^{-1}$ for $\mathbf{K}$ is defined as

$$\mathbf{B}_{asm}^{-1} = \sum_{j=1}^{N_s} \mathbf{R}_j^\delta {\mathbf{B}_j}^{-1} \mathbf{R}_j^\delta.$$

In addition to this standard additive Schwarz method (ASM) described above, we also consider the newly introduced restricted version (RAS) of the method [4, 8]. For some applications, the restricted version requires less communication time since one of the restriction or extension operations does not involve any overlap. The RAS preconditioner is defined as

$$\mathbf{B}_{ras}^{-1} = \sum_{j=1}^{N_s} \mathbf{R}_j^\delta {\mathbf{B}_j}^{-1} \mathbf{R}_j^0.$$

Some numerical comparisons of the ASM and RAS are presented later in the paper.

When the Schwarz preconditioner is applied to symmetric positive definite systems arising from the discretization of elliptical problems in $H_0^1(\Omega)$, the condition number κ of the preconditioned system satisfies $\kappa \leqslant C\,(1 + H/\delta)\,/H^2$, where C is independent of h, H, δ and the shapes of Ω and Ω_j^δ [23], that is, a Schwarz preconditioned Krylov subspace method is expected to have the following properties:

The number of iterations grows approximately proportional to $1/H$; (3.1)

If δ is maintained proportional to H, the number of iterations is bounded independently of h and H/h (a parameter related to the number of degrees of freedom of each subproblem); (3.2)

The convergence gets better as δ is increased. (3.3)

Theoretically, results (3.1)-(3.3) may not be applied immediately to Krylov subspace methods, e.g. GMRES [22], for the solution of indefinite linearized KKT systems. Nonetheless, we carefully examine all the properties in our numerical experiments. In particular, let $\bar{l}$ be the average number of Schwarz preconditioned GMRES iterations per linearized KKT system. We look for the following scalability properties:

For fixed h and δ, $\bar{l}$ increases as H decreases; (3.4)

For fixed H and δ, $\bar{l}$ is not very sensitive to the mesh refinement; (3.5)

For fixed h and H, $\bar{l}$ decreases as δ increases. (3.6)

4 Boundary control of incompressible Navier-Stokes flows

In this section we discuss the boundary control of the two-dimensional steady-state incompressible Navier-Stokes equations in the velocity-vorticity formulation [21]. The velocity is denoted by $\mathbf{v} = (v_1, v_2)$ and the vorticity by ω. Let Ω be an open and bounded polygonal domain in the plane, $\Gamma = \partial\Omega$ its boundary and $\boldsymbol{\nu}$ the unit outward normal vector along Γ. Let $\mathbf{f}$ be a given external force defined in Ω. Re is the Reynolds number and curl $\mathbf{f} = -\frac{\partial f_1}{\partial x_2} + \frac{\partial f_2}{\partial x_1}$. A boundary control problem consists on finding $(v_1, v_2, \omega, u_1, u_2)$ such that the minimization

$$\min_{(\mathbf{s},\mathbf{u})\in\mathbf{S}\times\mathbf{U}} \mathcal{F}(\mathbf{s},\mathbf{u}) = \frac{1}{2}\int_\Omega \omega^2 \, d\Omega + \frac{c}{2}\int_{\Gamma_{u,1}} u_1^2 \, d\Gamma + \frac{c}{2}\int_{\Gamma_{u,2}} u_2^2 \, d\Gamma \tag{4.1}$$

is achieved subject to the constraints

$$\begin{cases} -\Delta v_1 - \frac{\partial \omega}{\partial x_2} = 0 & \text{in } \Omega, \\ -\Delta v_2 + \frac{\partial \omega}{\partial x_1} = 0 & \text{in } \Omega, \\ -\Delta\omega + Re\ v_1 \frac{\partial \omega}{\partial x_1} + Re\ v_2 \frac{\partial \omega}{\partial x_2} - Re \text{ curl } \mathbf{f} = 0 & \text{in } \Omega, \\ v_i - v_{D,i} = 0 & \text{on } \Gamma_{D,i}, \quad i = 1,2, \\ v_i - u_i = 0 & \text{on } \Gamma_{u,i}, \quad i = 1,2, \\ \omega + \frac{\partial v_1}{\partial x_2} - \frac{\partial v_2}{\partial x_1} = 0 & \text{on } \Gamma, \\ \int_\Gamma \mathbf{v}\cdot\boldsymbol{\nu}\, d\Gamma = 0, \end{cases} \tag{4.2}$$

where, for $i = 1, 2$, $\Gamma = \Gamma_{D,i} \cup \Gamma_{u,i}$, $\Gamma_{D,i}$ is the part of the boundary where the v_i velocity component is specified through a Dirichlet condition with a prescribed velocity $v_{D,i}$, and $\Gamma_{u,i}$ is the part of the boundary where the v_i velocity component is specified through a Dirichlet condition with a control velocity u_i. The positive constant parameter c is used to adjust the relative importance of the control norms on achieving the minimization, so indirectly constraining the size of those norms. The physical objective behind problem (4.1)-(4.2) is the minimization of turbulence [16, 17]. The last constraint, given by

$$\int_\Gamma \mathbf{v}\cdot\boldsymbol{\nu}\, d\Gamma = 0, \tag{4.3}$$

is necessary for the consistency with the physical law of mass conservation. So the control $\mathbf{u} = (u_1, u_2)$ cannot be any control: it must belong to the space of functions satisfying (4.3). We denote problems like $(4.1) - (4.2)$, where controls are allowed to assume nonzero normal values at the boundary, as *full* boundary control problems (BCPs), [17]. In these kind of problems one has $m \neq n_s$ due to the extra constraint (4.3). This fact also complicates the parallel finite differences approximation of Jacobian matrices, since one does not have only PDEs (i.e., equations with local behavior) anymore. One can also deal with *tangential* BCPs, where the control is allowed to be just tangential to the boundary and the velocity $\mathbf{v}_b = (v_{b,1}, v_{b,2})$, defined as, for $i = 1, 2$,

$$v_{b,i} = \begin{cases} v_{D,i} \text{ on } \Gamma_{D,i}, \\ 0 \quad \text{on } \Gamma \setminus \Gamma_{D,i}, \end{cases}$$

is assumed to satisfy $\int_\Gamma \mathbf{v}_b \cdot \boldsymbol{\nu}\, d\Gamma = 0$, and so, $m = n_s$. Since tangential BCPs restrict even more the space where the control $\mathbf{u} = (u_1, u_2)$ can exist, one naturally expects better objective function values with full BCPs. In this paper we only study tangential boundary control problems.

5 Numerical experiments

We consider both a simulation and a tangential BCP over the cavity $\Omega = (0,1) \times (0,1)$, with

$$\mathbf{f} = \begin{pmatrix} -\sin^2(\pi x_1)\cos(\pi x_2)\sin^2(\pi x_2) \\ \sin^2(\pi x_2)\cos(\pi x_1)\sin^2(\pi x_1) \end{pmatrix}.$$

The simulation problem has slip boundary conditions. In the case of the tangential BCP, the objective function has $c = 10^{-2}$ and $\Gamma_{u,1} = \Gamma_{u,2} = \Gamma$. We run flow problems for $Re = 200$ and $Re = 250$.

To discretize the flow problems, we use a five-point second order finite difference method on a uniform mesh. All derivative terms of interior PDEs are discretized with a second order central difference scheme. The boundary condition $\omega + \partial v_1/\partial x_2 - \partial v_2/\partial x_1 = 0$ on Γ is also discretized with a second order approximation [20].

In all experiments, the Jacobian matrix is constructed approximately using a multi-colored central finite difference method with step size 10^{-5}, [9]. For control problems, central finite differences provide KKT matrices closer to be symmetric than the ones computed by forward finite differences. To solve the Jacobian systems we use restarted GMRES with an absolute tolerance equal to 10^{-6}, a relative tolerance equal to 10^{-4}, a restart parameter equal to 90 and a maximum number of iterations equal to 5000. The GMRES tolerances are checked over preconditioned residuals. Regarding the one-level additive Schwarz preconditioner, the number of subdomains is equal to the number of processors and the extended subdomain problems have zero Dirichlet interior boundary conditions and are solved with sparse LU. Line search with the merit function defined in Section 2 is performed with cubic backtracking, with $c_1 = 10^{-4}$ in (2.5) and a minimum allowed step length $\alpha^{(k)}$ equal to 10^{-6}. For augmented Lagrangian merit functions we follow the strategy explained in Section 2 with $\rho^{(-1)}$=10. For Newton iterations we use an absolute stopping tolerance equal to 10^{-6} and a relative tolerance equal to 10^{-10} times the initial residual. The maximum allowed number of Newton iterations is 100.

All tests were performed on a cluster of Linux PCs and our parallel software was developed using the Portable, Extensible Toolkit for Scientific Computing (PETSc) library [1], from Argonne National Laboratory. Our main concern is the scalability of the algorithms in terms of the linear and nonlinear iteration numbers. CPU times are also reported, but they should not be taken as a reliable measure of the scalability of the algorithms because our network is relatively slow and is shared with other processes.

Results are grouped into tables according to a unique combination of problem type (simulation or control), Reynolds number Re, ASM type (standard or

restricted), overlap δ and merit function (standard or augmented Lagrangian). Each table presents results for nine situations, related to three different sizes of meshes (parameter h) and three different numbers of processors (parameter H). For each case we report:

- the total number of Newton iterations: n,
- the average number of GMRES iterations per Newton iteration: $\bar{l}$,
- the total CPU time in seconds spent on all Newton iterations: t_n,
- the average CPU time, in seconds, per Newton iteration, spent on solving for the Newton steps: $\bar{t}_l$.

For each table we compare the behavior of $\bar{l}$ against (3.4)-(3.6). Predictions (3.4) and (3.5) can be checked by observing the values of $\bar{l}$ in a column (fixed h and δ) and in a row (fixed H and δ), respectively. Prediction (3.6) can be checked by observing the values of $\bar{l}$ at the same situation (fixed problem type, Re, ASM type, merit function, H and h) in different tables (different δ). We also compare approximate values of

$$\|\omega\|_h^2 = \int_{\Omega_h} \omega^2 \; d\Omega_h. \tag{5.1}$$

Table 1 presents results for the simulation problem with $Re = 200$. The preconditioner is ASM with $\delta = 1/64$ and the standard merit function is used in the line search. The total number of Newton iterations does not change with the mesh size or the number of processors. The average number of Krylov iterations per Newton iteration changes as expected in predictions (3.4) and (3.5).

The next three tables present results for the tangential BCP with $Re = 200$. An augmented Lagrangian merit function is used in the line search. Several different overlap values are used in the ASM preconditioner and the results are summarized as follows: Table 2 for $\delta = 1/64$, Table 3 for $\delta = 1/32$ and Table 4 for $\delta = 1/16$. Changes on the total number of Newton iterations w.r.t. the mesh size and the number of processors are not pronounced. We observe that $\bar{l}$ follows (3.6). With the same δ used on the simulation problem, we can see in Table 2 that the average number of GMRES iterations is now more sensitive to both h and, especially, H. Table 3 is the one where $\bar{l}$ best follows both (3.4) and (3.5) and the CPU times t_n and $\bar{t}_l$ for the finest mesh decrease with the increase on the number of processors. Table 4 shows that if δ gets too big then the consequent decrease on $\bar{l}$ might not compensate the increased time taken by sparse LU on the larger extended subdomains; that is, $\bar{t}_l$ increases. Comparing values of $\bar{t}_l$ in Tables 2 and 3 with the values in Table 1, we see that the average time spent on computing $\mathbf{p}^{(k)}$ can be more than 10 times bigger in control problems than in simulation problems on the same mesh, instead of being around $8/3 \approx 3$ times bigger, in accordance to the ratio between the number of variables per mesh point on control problems and on simulation problems.

As reported before we use a GMRES relative tolerance of 1.0×10^{-4}, a GMRES absolute tolerance of 1.0×10^{-6} and a Newton absolute tolerance of 1.0×10^{-6}. Case "(*)" in Table 3, however, gives results for a Newton absolute tolerance of 1.2×10^{-6}. When a Newton absolute tolerance of 1.0×10^{-6} is used,

although full steps are accepted in some iterations, the line search stalls once $||F||_2 \approx 1.14 \times 10^{-6}$. If we change the GMRES relative tolerance to 1.0×10^{-6} and the GMRES absolute tolerance to 1.0×10^{-13} (in order to obtain a more accurate Newton step) then we achieve $||F||_2 < 1.00 \times 10^{-6}$ with $n = 6$, $\bar{l} \approx 272$, $t_n \approx 9.38$ and $\bar{t}_l \approx 1.42$. Although we performed our tests with fixed GMRES tolerances, this experiment suggests that for more demanding Newton tolerances one might need to use decreasing GMRES tolerances as the outer loop proceeds, as expected by the theory for superlinear convergence of the inexact Newton method [10, 19].

In the next two tables, we change the preconditioner to RAS and everything else stays the same; i.e., these results are for the tangential BCP with $Re = 200$ and we use an augmented Lagrangian merit function in the line search. We increase the overlap size in the RAS preconditioner as follows: Table 5 for $\delta = 1/32$ and Table 6 for $\delta = 1/16$. The average number of GMRES iterations continues to follow (3.6) but now it better follows (3.4) and (3.5). The computing times t_n and $\bar{t}_l$ for the finest mesh in Table 6 decrease with the increase on the number of processors. The average number of GMRES iterations is larger in Table 5 than in Table 3, but the saving in communications of RAS compensates this increase so that $\bar{t}_l$ does not increase proportionally. By comparing finest mesh results on Tables 6 and 4 we see that RAS performs better than the standard ASM in terms of both $\bar{l}$ and $\bar{t}_l$, resulting on a smaller t_n.

Table 7 presents results for the tangential BCP with $Re = 250$, restricted ASM with $\delta = 1/16$ and augmented Lagrangian merit function. We can see that, with a Reynolds number greater than that in the previous table, both nonlinear (n) and average linear ($\bar{l}$) complexities increase.

Table 1. Results for the cavity flow simulation problem with $Re = 200$, standard ASM with overlap $\delta = 1/64$ and standard merit function $\|\mathbf{F}\|_2^2/2$. n is the total number of Newton iterations, $\bar{l}$ is the average number of Krylov iterations per Newton iteration, t_n is the total time in seconds spent on all Newton iterations and $\bar{t}_l$ is the average time in seconds, per Newton iteration, spent on solving for Newton steps. For the case of finest mesh, the number of variables is $198,147$ and $\|\omega\|_h^2 \approx 55.4$. See (5.1).

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 5$	$t_n \approx 0.83$	$n = 5$	$t_n \approx 3.6$	$n = 5$	$t_n \approx 18.3$
	$\bar{l} \approx 46$	$\bar{t}_l \approx 0.13$	$\bar{l} \approx 47$	$\bar{t}_l \approx 0.62$	$\bar{l} \approx 47$	$\bar{t}_l \approx 3.29$
32	$n = 5$	$t_n \approx 0.71$	$n = 5$	$t_n \approx 2.7$	$n = 5$	$t_n \approx 12.1$
	$\bar{l} \approx 60$	$\bar{t}_l \approx 0.091$	$\bar{l} \approx 63$	$\bar{t}_l \approx 0.46$	$\bar{l} \approx 63$	$\bar{t}_l \approx 2.20$
64	$n = 5$	$t_n \approx 0.70$	$n = 5$	$t_n \approx 1.99$	$n = 5$	$t_n \approx 7.50$
	$\bar{l} \approx 69$	$\bar{t}_l \approx 0.077$	$\bar{l} \approx 80$	$\bar{t}_l \approx 0.32$	$\bar{l} \approx 79$	$\bar{t}_l \approx 1.35$

Table 2. Results for the cavity tangential boundary flow control problem with $Re = 200$, standard ASM with overlap $\delta = 1/64$ and augmented Lagrangian merit function. n is the total number of Newton iterations, $\bar{l}$ is the average number of Krylov iterations per Newton iteration, t_n is the total time in seconds spent on all Newton iterations and $\bar{t}_l$ is the average time in seconds, per Newton iteration, spent on solving for Newton steps. For the case of finest mesh, the number of variables is $528,392$ and $\|\omega\|_h^2 \approx 32.5$. See (5.1).

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 7$	$t_n \approx 9.92$	$n = 8$	$t_n \approx 52.1$	$n = 6$	$t_n \approx 238$
	$\bar{l} \approx 92$	$\bar{t}_l \approx 1.21$	$\bar{l} \approx 85$	$\bar{t}_l \approx 5.78$	$\bar{l} \approx 100$	$\bar{t}_l \approx 36.4$
32	$n = 7$	$t_n \approx 10.3$	$n = 8$	$t_n \approx 53.3$	$n = 6$	$t_n \approx 264$
	$\bar{l} \approx 208$	$\bar{t}_l \approx 1.34$	$\bar{l} \approx 204$	$\bar{t}_l \approx 6.26$	$\bar{l} \approx 272$	$\bar{t}_l \approx 42.5$
64	$n = 7$	$t_n \approx 5.39$	$n = 8$	$t_n \approx 25.6$	$n = 6$	$t_n \approx 108$
	$\bar{l} \approx 187$	$\bar{t}_l \approx 0.67$	$\bar{l} \approx 182$	$\bar{t}_l \approx 2.96$	$\bar{l} \approx 216$	$\bar{t}_l \approx 17.1$

Table 3. Results for the cavity tangential boundary flow control problem with $Re = 200$, standard ASM with overlap $\delta = 1/32$ and augmented Lagrangian merit function. For the case of finest mesh, the number of variables is $528,392$ and $\|\omega\|_h^2 \approx 32.5$. Case "(*)" is discussed in Section 5.

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 7$	$t_n \approx 8.22$	$n = 8$	$t_n \approx 49.1$	$n = 7$	$t_n \approx 254$
	$\bar{l} \approx 58$	$\bar{t}_l \approx 0.96$	$\bar{l} \approx 59$	$\bar{t}_l \approx 5.38$	$\bar{l} \approx 62$	$\bar{t}_l \approx 33.1$
32	$n = 7$	$t_n \approx 8.32$	$n = 8$	$t_n \approx 46.6$	$n = 6$	$t_n \approx 199$
	$\bar{l} \approx 119$	$\bar{t}_l \approx 1.05$	$\bar{l} \approx 130$	$\bar{t}_l \approx 5.41$	$\bar{l} \approx 140$	$\bar{t}_l \approx 31.6$
64	$n = 7$ (*)	$t_n \approx 6.21$	$n = 8$	$t_n \approx 27.6$	$n = 6$	$t_n \approx 110$
	$\bar{l} \approx 155$	$\bar{t}_l \approx 0.78$	$\bar{l} \approx 132$	$\bar{t}_l \approx 3.18$	$\bar{l} \approx 143$	$\bar{t}_l \approx 17.4$

Table 4. Results for the cavity tangential boundary flow control problem with $Re = 200$, standard ASM with overlap $\delta = 1/16$ and augmented Lagrangian merit function. For the case of finest mesh, the number of variables is $528,392$ and $\|\omega\|_h^2 \approx 32.5$.

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 7$	$t_n \approx 9.65$	$n = 8$	$t_n \approx 59.2$	$n = 7$	$t_n \approx 744$
	$\bar{l} \approx 47$	$\bar{t}_l \approx 1.16$	$\bar{l} \approx 44$	$\bar{t}_l \approx 6.62$	$\bar{l} \approx 50$	$\bar{t}_l \approx 103$
32	$n = 7$	$t_n \approx 10.8$	$n = 8$	$t_n \approx 75.1$	$n = 7$	$t_n \approx 616$
	$\bar{l} \approx 100$	$\bar{t}_l \approx 1.39$	$\bar{l} \approx 108$	$\bar{t}_l \approx 8.96$	$\bar{l} \approx 149$	$\bar{t}_l \approx 86.3$
64	$n = 6$	$t_n \approx 6.36$	$n = 8$	$t_n \approx 56.5$	$n = 7$	$t_n \approx 331$
	$\bar{l} \approx 104$	$\bar{t}_l \approx 0.94$	$\bar{l} \approx 115$	$\bar{t}_l \approx 6.79$	$\bar{l} \approx 128$	$\bar{t}_l \approx 46.4$

Table 5. Results for the cavity tangential boundary flow control problem with $Re = 200$, restricted ASM with overlap $\delta = 1/32$ and augmented Lagrangian merit function. n is the total number of Newton iterations, $\bar{l}$ is the average number of Krylov iterations per Newton iteration, t_n is the total time in seconds spent on all Newton iterations and $\bar{t}_l$ is the average time in seconds, per Newton iteration, spent on solving for Newton steps. For the case of finest mesh, the number of variables is $528,392$ and $\|\omega\|_h^2 \approx 32.5$. See (5.1).

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 7$	$t_n \approx 8.33$	$n = 9$	$t_n \approx 53.1$	$n = 7$	$t_n \approx 253$
	$\bar{l} \approx 59$	$\bar{t}_l \approx 0.98$	$\bar{l} \approx 57$	$\bar{t}_l \approx 5.15$	$\bar{l} \approx 62$	$\bar{t}_l \approx 32.9$
32	$n = 7$	$t_n \approx 8.47$	$n = 8$	$t_n \approx 46.9$	$n = 6$	$t_n \approx 211$
	$\bar{l} \approx 131$	$\bar{t}_l \approx 1.07$	$\bar{l} \approx 134$	$\bar{t}_l \approx 5.45$	$\bar{l} \approx 154$	$\bar{t}_l \approx 33.6$
64	$n = 7$	$t_n \approx 6.38$	$n = 8$	$t_n \approx 30.1$	$n = 6$	$t_n \approx 132$
	$\bar{l} \approx 175$	$\bar{t}_l \approx 0.81$	$\bar{l} \approx 162$	$\bar{t}_l \approx 3.50$	$\bar{l} \approx 184$	$\bar{t}_l \approx 21.1$

Table 6. Results for the cavity tangential boundary flow control problem with $Re = 200$, restricted ASM with overlap $\delta = 1/16$ and augmented Lagrangian merit function. For the case of finest mesh, the number of variables is $528,392$ and $\|\omega\|_h^2 \approx 32.5$.

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 7$	$t_n \approx 9.66$	$n = 8$	$t_n \approx 60.7$	$n = 7$	$t_n \approx 743$
	$\bar{l} \approx 47$	$\bar{t}_l \approx 1.15$	$\bar{l} \approx 46$	$\bar{t}_l \approx 6.80$	$\bar{l} \approx 51$	$\bar{t}_l \approx 103$
32	$n = 7$	$t_n \approx 9.49$	$n = 8$	$t_n \approx 65.5$	$n = 6$	$t_n \approx 436$
	$\bar{l} \approx 90$	$\bar{t}_l \approx 1.21$	$\bar{l} \approx 86$	$\bar{t}_l \approx 7.7$	$\bar{l} \approx 98$	$\bar{t}_l \approx 71.0$
64	$n = 7$	$t_n \approx 7.18$	$n = 9$	$t_n \approx 56.9$	$n = 7$	$t_n \approx 309$
	$\bar{l} \approx 105$	$\bar{t}_l \approx 0.91$	$\bar{l} \approx 97$	$\bar{t}_l \approx 6.1$	$\bar{l} \approx 114$	$\bar{t}_l \approx 43.3$

Table 7. Results for the cavity tangential boundary flow control problem with $Re = 250$, restricted ASM with overlap $\delta = 1/16$ and augmented Lagrangian merit function. For the case of finest mesh, the number of variables is $528,392$ and $\|\omega\|_h^2 \approx 50.2$.

# Procs.	Mesh					
	64×64		128×128		256×256	
16	$n = 11$	$t_n \approx 15.8$	$n = 13$	$t_n \approx 110$	$n = 10$	$t_n \approx 1090$
	$\bar{l} \approx 51$	$\bar{t}_l \approx 1.22$	$\bar{l} \approx 55$	$\bar{t}_l \approx 7.67$	$\bar{l} \approx 55$	$\bar{t}_l \approx 106$
32	$n = 11$	$t_n \approx 17.2$	$n = 13$	$t_n \approx 126$	$n = 9$	$t_n \approx 726$
	$\bar{l} \approx 107$	$\bar{t}_l \approx 1.42$	$\bar{l} \approx 112$	$\bar{t}_l \approx 9.23$	$\bar{l} \approx 123$	$\bar{t}_l \approx 79.0$
64	$n = 10$	$t_n \approx 12.6$	$n = 13$	$t_n \approx 102$	$n = 9$	$t_n \approx 504$
	$\bar{l} \approx 135$	$\bar{t}_l \approx 1.16$	$\bar{l} \approx 139$	$\bar{t}_l \approx 7.62$	$\bar{l} \approx 180$	$\bar{t}_l \approx 55.1$

6 Conclusions

We have developed a general LNKSz algorithm for PDE constrained optimization problems and applied it to some tangential boundary control problems involving two-dimensional incompressible Navier-Stokes equations. In our numerical experiments the LNKSz algorithm, together with an augmented Lagrangian merit function, provides a fully parallel and robust solution method. The one-level additive Schwarz preconditioned GMRES, with a proper overlap, works well for the indefinite linearized KKT systems. A proper overlap for a control problem seems to be greater than a proper overlap for a simulation problem. More precisely, in our experiments the proper overlaps are two to four times greater in the control problems. For larger overlaps the restricted version of ASM seems to perform better than the standard ASM as a preconditioner for linearized KKT systems. Theoretically, as a full space SQP method, LNKSz does not guarantee descent directions, and the solution of the KKT system through proper steps is guaranteed only to be a local minimum. However, in our numerical tests, the computed steps are always descent directions and $\|\omega\|_h^2$ decreases with the computed boundary control.

References

1. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *Portable, Extensible Toolkit for Scientific computation (PETSc) home page*. http://www.mcs.anl.gov/petsc, 2003.
2. G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization, part I: the Krylov-Schur solver*, SIAM J. Sci. Comput., to appear.
3. ———, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization, part II: the Lagrange-Newton solver and its application to optimal control of steady viscous flows*, SIAM J. Sci. Comput., to appear.
4. X.-C. Cai, M. Dryja, and M. Sarkis, *Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems*, SIAM J. Numer. Anal., 41 (2003), pp. 1209-1231.
5. X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young, *Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246-265.
6. X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms,* SIAM J. Sci. Comput., 24 (2002), pp. 183-200.
7. X.-C. Cai, D. E. Keyes, and L. Marcinkowski, *Nonlinear additive Schwarz preconditioners and applications in computational fluid dynamics,* Int. J. Numer. Meth. Fluids, 40 (2002), pp. 1463-1470.
8. X.-C. Cai and M. Sarkis, *A restricted additive schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.
9. T. F. Coleman and J. J. Moré, *Estimation of sparse Jacobian matrices and graph coloring problem*, SIAM J. Numer. Anal., 20 (1983), pp. 243-209.
10. J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unsconstrained Optimization and Nonlinear Equations*, SIAM, second ed., 1996.

11. M. Dryja and O. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comp., 15 (1994), pp. 604–620.
12. S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton method*, SIAM J. Optim., 4 (1994), pp. 393-422.
13. ——, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16-32.
14. M. D. Gunzburger and L. S. Hou, *Finite-dimensional approximation of a class of constrained nonlinear optimal control problems*, SIAM J. on Control and Optimization, 34 (1996), pp. 1001–1043.
15. M. D. Gunzburger, L. S. Hou, and T. P. Svobodny, *Optimal control and optimization systems arising in optimal control of viscous incompressible flows*, Incompressible Computational Fluid Dynamics: Trends and Advances, Max D. Gunzburger and Roy A. Nicolaides, eds., Cambridge University Press, (1993), pp. 109–150.
16. L. S. Hou and S. S. Ravindran, *A penalized Neumann control approach for solving an optimal Dirichlet control problem for the Navier-Stokes equations*, SIAM J. Sci. Comput., 20 (1998), pp. 1795–1814.
17. ——, *Numerical approximation of optimal flow control problems by a penalty method: Error estimates and numerical results*, SIAM J. Sci. Comput., 20 (1999), pp. 1753–1777.
18. A. Ioffe and V. Tihomirov, *Theory of Extremal Problems*, North-Holland Publishing Company, 1979. Translation from russian edition ©NAUKA, Moscow, 1974.
19. J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, first ed., 2000.
20. E. Prudencio, R. Byrd, and X.-C. Cai, *Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems*, 2004, submitted.
21. L. Quartapelle, *Numerical Solution of the Incompressible Navier-Stokes Equations*, International Series of Numerical Mathematics, Vol. 113, Birkhäuser Verlag, 1996.
22. Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, second ed., 2003.
23. B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
24. V. M. Tihomirov, *Fundamental Principles of the Theory of Extremal Problems*, John-Wiley & Sons, 1986.