NASA Contractor Report 187497
ICASE Report No. 91–2

ICASE

"ONE SHOT" METHODS FOR OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS I: FINITE DIMENSIONAL CONTROL

Shlomo Ta'asan

Contract No. NAS1–18605
January 1991

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665–5225

Operated by the Universities Space Research Association

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665-5225

"One Shot" Methods for Optimal Control of Distributed Parameter Systems I: Finite Dimensional Control

Shlomo Ta'asan[1]
Department of Applied Mathematics and Computer Science
The Weizmann Institute of Science
and
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23665

ABSTRACT

This paper discusses the efficient numerical treatment of optimal control problems governed by elliptic PDE's and systems of elliptic PDE's, where the control is finite dimensional. Distributed control as well as boundary control cases are discussed. The main characteristic of the new methods is that they are designed to solve the full optimization problem directly, rather than accelerating a descent method by an efficient multigrid solver for the equations involved. The methods use the adjoint state in order to achieve efficient smoother and a robust coarsening strategy. The main idea is the treatment of the control variables on appropriate scales, i.e., control variables that correspond to smooth functions are solved for on coarse grids depending on the smoothness of these functions. Solution of the control problems is achieved with the cost of solving the constraint equations about two to three times (by a multigrid solver). Numerical examples demonstrate the effectiveness of the method proposed in distributed control case, pointwise control and boundary control problems.

[1]This research was made possible in part by funds granted to the author through a fellowship program sponsored by the Charles H. Revson Foundation and in part by the National Aeronautics and Administration under NASA Contract No. NAS1-18605 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, Va 23665 and in part by the Air-Force Office of Scientific Research, United States Air Force under Grant AFOSR-86-0127.

1 Introduction

Computations in optimal control for distributed parameter systems demand huge computational resources. These are optimization problems with constraints which are partial differential equations. This paper focuses on the computational aspects of optimal control problems governed by elliptic partial differential equations or elliptic systems, where the control variables are of finite dimension.

The classical optimal control algorithms use the costate (a Lagrange multiplier) in order to define an iteration that converges to the minimum. These are descent type algorithms which involve the solution of the state and costate equations per iteration of updating the control variables. The state and costate equations, being discretizations of partial differential equations are quite expensive to solve. The over all solution of the optimization problem becomes therefore very expensive.

The need for more efficient methods is obvious. One way of improving the situation is by using fast solvers , i.e., multigrid methods, for the state and costate equations at each step of a descent algorithm. This may result in a significant saving in computational time, but will not affect the number of outer iterations for updating the control unknowns; it will just make the time required for each iteration smaller.

A much more effective solution process can be obtained by having an algorithm that is aimed at the full optimization problem directly, rather than accelerating a linear equation solver. This paper is devoted to this approach. Multigrid algorithms that were designed to tackle the full problem rather than being a mere fast linear solver have been developed for a class of problems. The first one to be developed was the FAS algorithm for nonlinear equations [B1]. Recently such an approach was successful in treating some bifurcation problems [T2] and stability calculation [T2]. In all these cases maximal efficiency has been obtained as a result of treating the full problem directly.

The high efficiency in the problems treated here is achieved by working on all unknowns of the problems (i.e., state, costate and control unknowns) simultaneously where scales of the different unknowns are taken into account. The main rule is to treat different control variables on different grids depending on their scale of influence. This means that a variable that has a non local effect on the solution should be treated mainly on coarse levels. On the other hand, a variable that has a non-smooth effect on the solution in some neighborhood, will be relaxed in that neighborhood on fine levels as well. More generally, local work on fine grids will be done to update a certain control variable only at the vicinity where it has a non smooth effect on the solution while work on coarser levels for that variable will be in larger and larger neighborhoods of that region.

As is done in the classical optimal control algorithms a costate variable is introduced in our solution process. This allows the construction of good relaxation schemes in general, and is even more important when coarsening is considered (as explained in section 4). In the relaxation part of the algorithm we distinguish two main steps. One is designed to smooth the errors in the state and costate for a given choice of the control variables. This is done for the state variable using the Gauss-Seidel (GS) relaxations for scalar elliptic equations of state and Distributed-Gauss-Seidel (DGS) for systems of

elliptic equations [B2]. The equations obtained for the costate are very similar to the state equations and are relaxed using essentially the same relaxation method as for the state equations. The other step of the relaxation is designed for updating the control variables. This step of the relaxation is done taking into account the smoothness of the change introduced in the state and costate as a result of a small change in one of the control variables. That is, some work is to be done on coarse grids only and some on fine levels in special subdomains. The update of the control variables is done in the following way. A small change is introduced in them, then its effect on the state and the costate is computed approximately. The amplitude of that change in the control is then chosen so as to reduce the functional or to minimize the residuals of certain equations.

The algorithms constructed that way are aimed at the full problem directly and need no iterations within iterations as the standard algorithms require.

Numerical examples show that solutions to the levels of discretization errors are obtained in just a few work units, where a work unit is the work involved in relaxing the state equations. The complexity of solving a control problem is about twice that of solving the corresponding constraint equations, which is essentially optimal. Examples for distributed control, boundary control and pointwise control are given and all show the same efficiency.

2 The Problem

Let $\mathcal{V}, \mathcal{W}, \mathcal{U}$ and $\mathcal{H}$ be Hilbert spaces. Let A, B and C be operators as follows $A : \mathcal{V} \rightarrow \mathcal{W}$, $B : \mathcal{U} \rightarrow \mathcal{W}$, $C : \mathcal{V} \rightarrow \mathcal{H}$ and let $\mathcal{U}_{ad} \subseteq \mathcal{U}$. Consider the problem

$$\begin{array}{c} \min_{u \in \mathcal{U}_{ad}} \|C\bar{x} - \bar{d}\|^2_{\mathcal{H}} \\ A\bar{x} = Bu + f \end{array} \tag{2.1}$$

Here A is assumed to be an elliptic partial differential equation (PDE) or an elliptic system of PDE's with smooth coefficients (see [ADN]). B is assumed to be a finite dimensional operator, that is, there exist functions $\psi_j \in \mathcal{W} \quad j = 1, \ldots, q$ such that

$$Bu = \sum_{j=1}^{q} u_i \psi_i \tag{2.2}$$

We assume that the functions ψ_j are smooth except for a finite number of singularities. $\mathcal{U}$ is then identified with $\mathbb{R}^q$. The adjoint of B, $B^* : \mathcal{W} \rightarrow \mathbb{R}^q$ is given by

$$B^*\phi = (< \psi_1, \phi >_{\mathcal{W}}, \ldots, < \psi_q, \phi >_{\mathcal{W}})^T \tag{2.3}$$

The following equations hold at the minimum,

$$\begin{array}{c} Ax = Bu + f \\ A^*p + C^*Cx = C^*d \\ B^*p = 0 \end{array} \tag{2.4}$$

where B and B^* are given by (2.2) and (2.3) respectively. Let x_0 be a solution of $Ax_0 = f$ and $d = \bar{d} - Cx_0$. Problem (2.1) can be written in terms of $x = \bar{x} - x_0$ and u as

$$\begin{array}{c} \min_{u \in \mathcal{U}_{ad}} \|Cx - d\|^2_{\mathcal{H}} \\ Ax = Bu \end{array} \tag{2.5}$$

So without loss of generality we can assume that $f = 0$.

2.1 Descent Methods

Iterative methods for solving the control problem are described. A triple (x, u, p) will be called compatible if it satisfies the equations

$$\begin{array}{c} Ax - Bu = 0 \\ A^*p + C^*Cx = C^*d \end{array} \tag{2.6}$$

Given an approximate solution (x, u) of (2.5), an improved one is obtained as follows. Denote by $E(x, u)$ the functional $\|Cx - d\|^2_{\mathcal{H}}$. For a compatible triple (x, u, p)

$$E(x, u) = - < Cx, Cx >_{\mathcal{H}} -2 < u, B^*p >_u + < d, d >_{\mathcal{H}} \tag{2.7}$$

We look for a change $\gamma\tilde{u}$ in u and corresponding changes in x, p that result in a new compatible triple, and for which the value of E is smaller. Let $(x + \gamma\tilde{x}, u + \gamma\tilde{u}, p + \gamma\tilde{p})$ be a new compatible triple. Then

$$\begin{array}{c} E(x + \gamma\tilde{x}, u + \gamma\tilde{u}) = E(x, u) + \\ -2\gamma < C\tilde{x}, Cx >_{\mathcal{H}} -\gamma^2 < C\tilde{x}, C\tilde{x} >_{\mathcal{H}} \\ -2\gamma < B^*p, \tilde{u} >_u -2\gamma < B^*\tilde{p}, u >_u -2\gamma^2 < B^*\tilde{p}, \tilde{u} >_{\mathcal{H}} \end{array} \tag{2.8}$$

Choosing $\tilde{u} = B^*p$, and $\tilde{x}, \tilde{p}$ satisfying $A\tilde{x} - B\tilde{u} = 0$ and $A^*\tilde{p} + C^*C\tilde{x} = 0$ we get

$$\begin{array}{c} E(x + \gamma\tilde{x}, u + \gamma\tilde{u}) = E(x, u) \\ -2\gamma(< C\tilde{x}, Cx >_{\mathcal{H}} + < B^*p, B^*p >_u + < B^*\tilde{p}, u >_u) \\ -\gamma^2(2 < B^*\tilde{p}, B^*p >_u + < C\tilde{x}, C\tilde{x} >_{\mathcal{H}}) \end{array} \tag{2.9}$$

Among all possible γ, the choice

$$\gamma = -\frac{1}{2}\frac{< C\tilde{x}, Cx >_{\mathcal{H}} + < B^*p, B^*p >_u + < B^*\tilde{p}, u >_u}{\|C\tilde{x}\|^2_{\mathcal{H}} + 2 < B^*\tilde{p}, B^*p >_u} \tag{2.10}$$

minimizes the functional in the direction chosen. This can be a basis for an algorithm for solving the optimal control problem.

A Descent Algorithm

Set $u_0 = 0, p_0 = 0$, $\gamma = 1$, $n = 1$.
While $\gamma > \epsilon$ **Do**
Begin

1. Solve for x_n the equation $Ax_n - Bu_n = 0$
2. Solve for p_n the equation $A^*p_n + C^*(Cx_n - d) = 0$
3. Set $\tilde{u} = B^*p_n$. Solve $A\tilde{x} - B\tilde{u} = 0$, $A^*\tilde{p} + C^*C\tilde{x} = 0$. (approximately) for $\tilde{x}, \tilde{p}$ respectively.
4. Set $u_{n+1} = u_n + \gamma\tilde{u}$, $x_{n+1} = x_n + \gamma\tilde{x}$, $p_{n+1} = p_n + \gamma\tilde{p}$, $n = n + 1$ with γ given by (2.10).

End

Approximate Descent Algorithm

Set $u_0 = 0, p_0 = 0$, $\gamma = 1, n = 1$.
While $\gamma > \epsilon$ **Do**
Begin

1. Starting with $x_n = x_{n-1}$ relax $Ax_n = Bu_n$, yielding a solution satisfying $Ax_n - Bu_n = r_x^n$
2. Starting with $p_n = p_{n-1}$ relax $A^*p_n + C^*Cx_n = C^*d$, yielding a solution satisfying $A^*p_n + C^*(Cx_n - d) = r_p^n$
3. Set $\tilde{u} = B^*p_n$. Solve $A\tilde{x} - B\tilde{u} = 0$, $A^*\tilde{p} + C^*C\tilde{x} = 0$ (approximately) for $\tilde{x}, \tilde{p}$ respectively.
4. Set $u_{n+1} = u_n + \gamma\tilde{u}$, $x_{n+1} = x_n + \gamma\tilde{x}$, $p_{n+1} = p_n + \gamma\tilde{p}$, $n = n + 1$ with γ given by (2.10).

End

ϵ is a small enough number, depending on the accuracy required for the solution. Note that if $\|r_x^n\| \to 0$ and $\|r_p^n\| \to 0$ then the above algorithm converges to the minimum of (2.5).

The expensive steps in this algorithm are the ones for obtaining $\tilde{x}$ and $\tilde{p}$ by approximately inverting A and A^*, which are elliptic operators (or systems). Moreover, the solution of the elliptic equations governing the state and costate needs to be done many times.

Remark Considering the form of the necessary conditions one may be tempted to choose γ based on minimizing the L_2 residuals of the third equations of (2.4). This leads to

$$\gamma = \frac{< B^*p, B^*\tilde{p} >}{< B^*\tilde{p}, B^*\tilde{p} >} \tag{2.11}$$

which may not be a good choice as $< B^*p, B^*\tilde{p} >$ may vanish, causing the iteration to get stuck away from the minimum.

3 Naive Multigrid Approaches

The approximate-descent algorithm presented in the previous section suggests an immediate acceleration procedure, namely, using some fast solver in steps 1,2, and 3. Since the form of the operator A^* is very similar to that of A the same fast solver can be used in all the steps. The operator A is assumed to be elliptic and therefore calls for multigrid methods.

Assume that we are given $\mathcal{V}_k, \mathcal{W}_k, \mathcal{U}_k$ and $\mathcal{H}_k$ Hilbert spaces approximation the original $\mathcal{V}, \mathcal{W}, \mathcal{U}$ and $\mathcal{H}$. Let A_k, B_k and C_k be operators as follows $A_k : \mathcal{V}_k \rightarrow \mathcal{W}_k$, $B_k : \mathcal{U}_k \rightarrow \mathcal{W}_k$, $C_k : \mathcal{V}_k \rightarrow \mathcal{H}_k$. Assume also that interpolation and restriction operators $P_x^k, P_p^k, P_u^k, R_x^k, R_p^k, R_u^k$ are given; $R_x^k : \mathcal{V}_{k+1} \rightarrow \mathcal{V}_k$, $R_p^k : \mathcal{W}_{k+1} \rightarrow \mathcal{W}_k$, $R_u^k : \mathcal{U}_{k+1} \rightarrow \mathcal{U}_k$, $P_x^k : \mathcal{V}_k \rightarrow \mathcal{V}_{k+1}$, $P_p^k : \mathcal{W}_k \rightarrow \mathcal{W}_{k+1}$, $P_u^k : \mathcal{U}_k \rightarrow \mathcal{U}_{k+1}$. The superscript k in these expressions represent a level of discretization where $k = 1$ correspond to the coarsest level.

Denote by $MG(A_k, b, x, y, \nu_1, \nu_2)$ a multigrid $V(\nu_1, \nu_2)$-cycle for solving $A_k x = b$ on level k, starting with initial approximation x ending with y. Assume that a solution of the minimization problem is needed for level $k = m$. A possible fast algorithm for it, starting with initial approximation (x^m, u^m, p^m) is denoted by

$$(x^m, u^m, p^m) \leftarrow MinMG(A_m, B_m, C_m, x^m, u^m, p^m, d^m) \tag{3.12}$$

and given by

Set $u_0^m = 0, P_0^m = 0, \gamma = 1, n = 1.$
While $\gamma > \epsilon$ **Do**
Begin
1. Perform a cycle $MG(A_m, B_m u_n^m, x_n^m, x_{n+1}^m, \nu_1, \nu_2)$
2. Perform a cycle $MG(A_m^*, C_m^* d^m - C_m^* C x_n^m, p_n^m, p_{n+1}^m, \nu_1, \nu_2)$
3. Perform a cycle $MG(A_m, B_m B_m^* p^m, \tilde{x}_n^m, \tilde{x}_{n+1}^m, \nu_1, \nu_2)$
4. Perform a cycle $MG(A_m^*, -C_m^* C_m \tilde{x}_{n+1}^m, \tilde{p}_n^m, \tilde{p}_{n+1}^m, \nu_1, \nu_2)$
5. Set $u_{n+1}^m = u_n^m + \gamma \tilde{u}_{n+1}^m$, $x_{n+1}^m = x_n^m + \gamma \tilde{x}_{n+1}^m$, $p_{n+1}^m = p_n^m + \gamma \tilde{p}_{n+1}^m$, $n = n + 1$ and γ given by (2.10).

End

An obvious improvement of this algorithm is to start with a good initial approximation for the optimization problem on level m. This can be achieved easily by solving the problem first on level $m - 1$. Applying this idea recursively one arrives at the following algorithm.

Set $u_0^1 = 0, x_0^1 = 0, p_0^1 = 0, k = 1.$

While $k \leq m$ **Do**

Begin

1. `Perform a cycle`
   $(x^k, u^k, p^k) \leftarrow MinMG(A_k, B_k, C_k, x^k, u^k, p^k, d^k)$
2. `If` $k = m$ `stop`
3. $x^{k+1} = \Pi_x^k x^k, p^{k+1} = \Pi_p^k p^k, u^{k+1} = P_u^k u^k, k = k + 1.$

End

where Π_x^k,Π_p^k are operators analogous to P_x^k and P_p^k, respectively, but of possibly a higher order (as is usual in multigrid algorithms for elliptic problems).

Our aim is to develop an algorithm in which the total computational cost of solving the control problem will be only few times (2-3) more than that of solving for the state variable x given that u is known. This is done in the next sections where a multigrid approach which is aimed at the full problem directly is constructed.

4 "One Shot" Multigrid Methods

In this section we describe a multigrid method that is aimed at the full optimization problem, rather than serving as a fast solver in a step of a basic optimization algorithm, as was done in the previous section.

The construction of this method has been achieved basically by following the next five steps.

1. Distinction between smooth versus non-smooth components in all types of variables involved
2. Construction of a basic non-expensive relaxation for the full problem.
3. Classification of rate of convergence of the basic relaxation for the different variables in the problem, with regard to their smoothness.
4. Based on 3, determination of the role of coarse grids in accelerating the fine grid convergence.
5. Construction of a coarse grid problem that approximate the *Full* fine grid residual problem.

Steps 1,2 and 3 are discussed in detail in section 4.2, the rest in section 4.1. The resulting algorithm involves a sequence of discrete optimization problems, starting from a very coarse discretization ending with a fine grid one on which a solution of the problem is required. The optimization problem on each level is served to accelerate the convergence of the next finer level problem. The iterative process for solving the finest grid problem involves a relaxation process on each level whose effect is to damp out the non-smooth part of the error, i.e., the part of the error which cannot be represented on the next coarse grid in the sequence. This together with a proper transferring of

information (residuals and current fine grid solution) from fine to coarse levels and back to fine levels results in a very high efficiency. A full description of the method includes a relaxation scheme and a coarsening strategy.

4.1 Coarsening

A coarsening scheme requires the definition of coarse grid spaces analogous to $\mathcal{V}, \mathcal{W}, \mathcal{U}, \mathcal{H}$ which we denote by the same letters with a superscript c. Also inter grid transfer P_x, P_p, P_u and R_x, R_p, R_u and coarse grid operators A_c, B_c, C_c are required. Having all this we can define the coarse grid minimization problem. An attempt to use

$$\begin{array}{c} \min_{u^c \in \mathcal{U}^c_{ad}} \|C_f(x^f + P_x x^c) - d^f\|^2 \\ A_c x^c - B_c u^c = R_x(g^f_x - A_f x^f + B_f u^f) \end{array} \tag{4.1}$$

as an approximation for the error on the fine grid fails. The reason for this is that this coarse grid minimization problem does not preserve the fine grid minimum once it was reached. This follows immediately from the necessary conditions of the coarse grid problem (4.1), namely,

$$\begin{array}{c} A_c x^c - B_c u^c = R_x(g^f_x - A_f x^f + B_f u^f) \\ A^*_c p^c + C^*_c C_c x^c = R_p C^*_f(d^f - C_f x^f) \\ B^*_c p^c = 0. \end{array} \tag{4.2}$$

The coarse grid minimization problem formulated above is a problem for the correction in the fine grid approximation. Therefore, upon reaching the fine grid minimum the coarse grid correction x^c, u^c should be zero. This however is not the case since the right hand side of the second equation is non-zero in general at the minimum of the fine grid problem.

In order to define a correct coarse grid problem we rewrite the fine grid minimization problem as

$$\begin{array}{c} \min_{u^f \in \mathcal{U}^f_{ad}} < C_f x^f, x^f > -2 < x^f, g^f_p > -2 < u^f, g^f_u > \\ A_f x^f - B_f u^f = g^f_x \end{array} \tag{4.3}$$

where $g^f_x = g$, $g^f_p = C^* d$ and $g^f_u = 0$ on the fine grid. Note that the problem is defined once g^f_x, g^f_u, g^f_p are given.

1. Correction Scheme. The coarse grid problem for the correction is of the same form, that is,

$$\begin{array}{c} \min_{u^c \in \mathcal{U}^c_{ad}} < C_c x^c, x^c > -2 < x^c, g^c_p > -2 < u^c, g^c_u > \\ A_c x^c - B_c u^c = g^c_x \end{array} \tag{4.4}$$

where

$$\begin{aligned} g_x^c &= R_x(g_x^f - A_f x^c + B_f u^f) \\ g_p^c &= R_p(g_p^f - A_f^* p^f - C_c^* C_c x^f) \\ g_u^c &= R_u(g_u^f - B_f^* p^f). \end{aligned} \tag{4.5}$$

The necessary conditions for this coarse grid problem are

$$\begin{aligned} A_c x^c - B_c u^c &= g_x^c \\ A_c^* p^c + C_c^* C_c x^c &= g_p^c \\ B_c^* p^c &= g_u^c \end{aligned} \tag{4.6}$$

which are the Correction Scheme (CS) (see [B2]) equations for the fine grid necessary conditions. In particular they have the property of introducing no change to the fine grid problem once the minimum has been reached.

Once an approximation to the coarse grid problem is found it is used to correct the fine grid approximation by

$$\begin{aligned} x^f &\leftarrow x^f + P_x x^c \\ u^f &\leftarrow u^f + P_u u^c \\ p^f &\leftarrow p^f + P_p p^c \end{aligned} \tag{4.7}$$

Note that this coarse grid problem depends on the costate p obtained after the fine grid relaxation process. Without using it the coarsening becomes very difficult as was explained before.

2. Full Approximation Scheme. The coarse grid problem for quantities that approximate the full fine grid solution rather than the correction for it have similar form. The minimization problem is

$$\begin{aligned} \min_{u^c \in \mathcal{U}_{ad}^c} &< C_c x^c, x^c > -2 < x^c, g_p^c > -2 < u^c, g_u^c > \\ A_c x^c - B_c u^c &= g_x^c \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} g_x^c &= R_x(g_x^f - A_f x^c + B_f u^f) + A^c \bar{R}_x x^f - B_c \bar{R}_u u^f \\ g_p^c &= R_p(g_p^f - A_f^* p^f - C_c^* C_c x^f) + A_c^* \bar{R}_p p^f + C_c^* C_c \bar{R}_x x^f \\ g_u^c &= R_u(g_u^f - B_f^* p^f) + B_c^* \bar{R}_p p^f. \end{aligned} \tag{4.9}$$

The operators $\bar{R}_x, \bar{R}_p, \bar{R}_u$ are fine to coarse grid transfers possibly different from R_x, R_p, R_u (see [B2]). The necessary conditions for this coarse grid problem are

$$\begin{aligned} A_c x^c - B_c u^c &= g_x^c \\ A_c^* p^c + C_c^* C_c x^c &= g_p^c \\ B_c^* p^c &= g_u^c \end{aligned} \tag{4.10}$$

which are the FAS equations for the fine grid necessary conditions. In particular they have the property of introducing no change to the fine grid problem once the minimum has been reached.

Once an approximation to the coarse grid problem is found it is used to correct the fine grid approximation by

$$\begin{aligned} x^f &\leftarrow x^f + P_x(x^c - \bar{R}_x x^f) \\ u^f &\leftarrow u^f + P_u(u^c - \bar{R}_u u^f) \\ p^f &\leftarrow p^f + P_p(p^c - \bar{R}_p p^f) \end{aligned} \tag{4.11}$$

4.2 Relaxation

The general equations for state and costate variables on any of the grids involved in the process have the form,

$$\begin{aligned} Ax - Bu &= f_x \\ A^*p + C^*Cx &= f_p \end{aligned} \tag{4.12}$$

These two equations are being relaxed on all levels using one of the standard relaxation methods for elliptic problems. That is, Gauss-Seidel or damped Jacobi in case A is a discretization of a scalar PDE, and Collective Gauss-Seidel (CGS) or Distributed Gauss-Seidel (DGS) when systems of elliptic PDE's are involved. So the problem of relaxation for x and p is quite standard. An important property of such relaxation schemes is that for a wide class of discretization methods (h-elliptic discretizations) they have fast convergence for h-non-smooth errors (defined later) (see [B2]). This implies that the errors in the x and p equations will have smooth residuals after just a few relaxation sweeps. This does not mean that the errors for the optimization problem are smooth. The relaxation of equations (4.12) serves as a process that brings the solution closer to the constraint surface defined by the first equation there.

As far as updating the control variables the situation is more complicated. Note that a small change in any of the control unknowns u_j implies a change in x and p everywhere in the domain. Good efficiency for the full algorithm is to be obtained if the changes in x and p that result from changes in u, can be calculated with a computational cost which is not much greater than that of relaxing the equations (4.12) once.

The smoothness assumption on the coefficients of the operator A implies that non-smooth errors correspond to non-smooth residuals and vice versa. Non-smooth errors are also fast to converge, for h-elliptic discretizations, as mentioned before (this is not the case for quasi-elliptic discretization schemes). This results in the following basic rule for updating the control variables.

- *Make changes in the control variables that result in non-smooth residuals*

Note that a change in any of the control variables may result in smooth changes on one grid but non-smooth changes on a coarser grid. This suggest that the different control variables will be relaxed on possibly different grids depending on their scale of

influence in the solution. To quantify more precisely these ideas we need the following definitions.

Definition: *A function $\psi_h(\xi)$ will be called h-non-smooth at $\xi \in \Omega^h$ iff*

$$|(\mu_h\psi_h)(\xi) - \psi_h(\xi)| \geq \alpha|\psi_h(\xi)| \quad 0 < \alpha_0 \leq \alpha < 1, \alpha_0 = O(1), \tag{4.13}$$

where μ_h is a local averaging operator.

Definition: *A function $\psi_h(\xi)$ will be called uniformly h-non-smooth in Ω iff it is h-non-smooth in every point $\xi \in \Omega^h$*

Thus if ψ_j^h is uniformly h-non-smooth its effect on the solution x and p can be calculated accurately with just a few relaxation sweeps. Typically 2-3 relaxation sweeps are enough to reduce the errors by an order of magnitude. This will be the key to our method of updating the control variables u_j. Let

$$\Xi_j^h = \left\{\xi \in \Omega^h : \psi_j^h \quad \text{is h-non-smooth at } \xi\right\} \tag{4.14}$$

If h correspond to the coarsest level and $\Xi_j^h \neq \phi$ then we define $\Xi_j^h = \Omega^h$. Define

$$\begin{aligned} D^h(\xi) &= \text{diag}(d_{11}^h(\xi), \ldots, d_{qq}^h(\xi)) \\ d_{kk}^h(\xi) &= \begin{cases} 1 & \text{if } \psi_k^h \text{ is h-non-smooth at } \xi \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.15}$$

The qxq matrix D^h will be used in defining the relaxation for the control variables.

In case the functions ψ_j are uniformly h-non-smooth the update of the control variables becomes much simpler. The perturbation in u that is done in minimizing the functional is by $\gamma D^h(B^h)^* p^h$, followed by a few relaxation sweeps of the state and costate equation in a vicinity of $\bigcup \Xi_j^h$, yielding a good enough approximation to $\tilde{x}^h$ and $\tilde{p}^h$. The actual perturbation is then chosen so as to minimize the functional or the other choice as explained in section 2. Note that the multiplication by D^h causes the perturbation in the right hand side to be h-non-smooth on the given level. That is, each control variable is being updated on the appropriate grid.

For ψ_j with a singularity a slightly different approach is taken. Here the effect on the solution is smooth away from the vicinity of non-smoothness of ψ_j. This follows form basic theories of elliptic partial differential equations [ADN]. Thus, if the observation operator C is supported far enough from the singularities of ψ_j, even coarse grids can compute accurately the effect on Cx, which is the important quantity in updating the u_j. However, when the singularities are close to the support of C this is not the case and some refinement has to be done locally in order to account for the local non-smoothness. In practice, a few points around the singularity need to be relaxed, further points are relaxed on coarser grids. The finest grid which needs to be involved in the process with a given singularity is determined from the accuracy achieved in $C\tilde{x}$ which can be estimated using the quantities

$$\|C^h\tilde{x}^h - C^h\tilde{x}^{2h}\| \tag{4.16}$$

and the knowledge about the accuracy of the discretization involved. The later, if not known, can be computed from

$$\frac{\|C^h\tilde{x}^h - C^h\tilde{x}^{2h}\|}{\|C^h\tilde{x}^{2h} - C^h\tilde{x}^{4h}\|} \tag{4.17}$$

In any case, the work performed in the calculation of $\tilde{x}^h, \tilde{p}^h$ should not involve more than one multigrid cycle, done as an FMG cycle used with local refinements. The exact formulation of the algorithm for updating the control variables in the most general case will be presented elsewhere. We denote the process of updating the control variables in the direction $\tilde{u}$ by

$$(x,p,u) \leftarrow MinRel(x,p,u,\tilde{u}) \tag{4.18}$$

Having defined the relaxation and the coarsening process the full algorithm can now be described.

4.3 Unistep Algorithms

Let $\mathcal{V}_k, \mathcal{W}_k, \mathcal{U}_k$ and $\mathcal{H}_k$ be Hilbert spaces approximating $\mathcal{V}, \mathcal{W}, \mathcal{U}$ and $\mathcal{H}$, respectively. Let A_k, B_k and C_k be operators as follows $A_k : \mathcal{V}_k \rightarrow \mathcal{W}_k$, $B^k : \mathcal{U}_k \rightarrow \mathcal{W}_k$, $C_k : \mathcal{V}_k \rightarrow \mathcal{H}_k$, approximating the operators A,B,C, respectively. Assume also that interpolation and restriction operators $P_x^k, P_p^k, P_u^k, R_x^k, R_p^k, R_u^k, \bar{R}_x^k, \bar{R}_p^k, \bar{R}_u^k$ are given as $R_x^k : \mathcal{V}_{k+1} \rightarrow \mathcal{V}_k$, $R_p^k : \mathcal{W}_{k+1} \rightarrow \mathcal{W}_k$, $R_u^k : \mathcal{U}_{k+1} \rightarrow \mathcal{U}_k$, $\bar{R}_x^k : \mathcal{V}_{k+1} \rightarrow \mathcal{V}_k$, $\bar{R}_p^k : \mathcal{W}_{k+1} \rightarrow \mathcal{W}_k$, $\bar{R}_u^k : \mathcal{U}_{k+1} \rightarrow \mathcal{U}_k$, $P_x^k : \mathcal{V}_k \rightarrow \mathcal{V}_{k+1}$, $P_p^k : \mathcal{W}_k \rightarrow \mathcal{W}_{k+1}$, $P_u^k : \mathcal{U}_k \rightarrow \mathcal{U}_{k+1}$. Also assume the the qxq matrices D_k (defined in section 4.2) are given.

On all levels minimization problems as follows are given,

$$\begin{array}{c} \min_{u^k \in \mathcal{U}_{ad}^k} < C_k x^k, x^k > -2 < x^k, g_p^k > -2 < u^k, g_u^k > \\ A_k x^k = B_k u^k + g_x^k \end{array} \tag{4.19}$$

Observe that this is equivalent to having the following systems of equations on all levels

$$\begin{array}{c} A_k x^k = B_k u^k + g_x^k \\ A_k^* p^k + C_k^* C_k x^k = g_p^k \\ B_k^* p^k = g_u^k. \end{array} \tag{4.20}$$

Let (x^k, p^k, u^k) be an approximate solution of (4.19). We define next an algorithm for improving it, denoted by

$$(x^k, p^k, u^k) \leftarrow MG(x^k, p^k, u^k, g_x^k, g_p^k, g_u^k) \tag{4.21}$$

If $k = 1$ **Then**

1. relax the first two equation in (4.20) until convergence.
2. iterate until convergence:

   2a. perform the cycle

   $(x^k, p^k, u^k) \leftarrow MinRel(x^k, p^k, u^k, D_k(g_u^k - B_k^* p^k))$

2b. relax the first two equations in (4.20).

Else

1. perform the following ν_1 times

   1a. relax the first two equation in (4.20)

   1b. perform the cycle

   $$(x^k, p^k, u^k) \leftarrow MinRel(x^k, p^k, u^k, D_k(g_u^k - B_k^* p^k))$$

2. Let $k = k - 1$, and

$$\begin{aligned} g_x^k &= R_x^k \ (g_x^{k+1} - A_{k+1} x^{k+1} + B_{k+1} u^{k+1}) \ + A_k R_x^k x^{k+1} \\ g_p^k &= R_p^k \ (g_p^{k+1} - A_{k+1}^* p^{k+1} + C_{k+1}^* C_{k+1} u^{k+1}) + A_k R_p^k p^{k+1} \\ g_u^k &= R_p^k \ (g_u^{k+1} - B_{k+1}^* p^{k+1}) \ + B_k R_p^k p^{k+1} \end{aligned}$$

3. perform γ times the cycle

   $$(x^k, p^k, u^k) \leftarrow MG(x^k, p^k, u^k, f_x^k, f_p^k, f_u^k)$$

4. correct fine grid solutions

$$\begin{aligned} x^{k+1} &\leftarrow x^{k+1} + P_x^k (x^k - R_x^k \ x^{k+1}) \\ p^{k+1} &\leftarrow p^{k+1} + P_p^k (p^k - R_p^k \ p^{k+1}) \\ u^{k+1} &\leftarrow u^{k+1} + P_u^k (u^k - R_u^k \ u^{k+1}) \end{aligned}$$

5. perform the following ν_1 times

   5a. relax the first two equation in (4.20)

   5b. perform the cycle

   $$(x^k, p^k, u^k) \leftarrow MinRel(x^k, p^k, u^k, D_k(g_u^k - B_k^* p^k))$$

End

In order to obtain full efficiency the algorithm starts at the coarsest level, where each time a refinement is done followed by a fixed number of MG cycles. This is the N-FMG algorithm which is defined next.

1. Solve (4.19) for $k = 1$

2. $k = k + 1$, $x^k = \Pi_x^{k-1} x^{k-1}$, $p^k = \Pi_p^{k-1} p^{k-1}$, $u^k = P_u^{k-1} u^{k-1}$.

3. Define g_x^k, g_p^k, g_u^k.

4. Perform N time the cycle

   $$(x^k, p^k, u^k) \leftarrow MG(x^k, p^k, u^k, f_x^k, f_p^k, f_u^k)$$

5. If $k = M$ stop, else goto 2.

At the end of this algorithm an approximate solution to the minimization problem on level $k = M$ is given.

5 Numerical Examples

Numerical experiments were conducted with scalar elliptic problems governed by the Laplacian in the unit square with Dirichlet boundary conditions. That is, using the notation of section 2,

$$\begin{array}{cc} \Omega = \{(x,y) : 0 \le x,y \le 1\} & \\ \mathcal{V} = \mathcal{W} = \{\phi \in L_2(\Omega) : \phi|_{\partial\Omega} = 0\} & \\ A = \triangle & \\ \mathcal{H} = L_2(\Gamma) \qquad \Gamma = \{(x,0) : 0 \le x \le 1\} & \\ C = \partial/\partial n|_\Gamma & \end{array} \tag{5.1}$$

The control space was $I\!R^q$. Different cases were considered for $(\psi_j \quad j = 1,\ldots,q)$, in the definition of B.

Uniform grids were used in the discretization with the standard 5-point formula for the Laplacian, namely,

$$\begin{array}{c} \Omega^h = \{(i/N, j/N) : 0 \le i,j \le N\} \\ \Gamma^h = \{(i/N, 0) : 0 \le i \le N\} \\ (A^h X^h)_{i,j} = \frac{1}{h^2}(X^h_{i-1,j} + X^h_{i+1,j} + X^h_{i,j-1} + X^h_{i,j+1} - 4X^h_{i,j}) \end{array} \tag{5.2}$$

where Nh = 1, h being the discretization parameter. The observation operator was discretized using a 3-point formula,

$$(C^h X^h)_{i,0} = \frac{1}{h}(1.5X^h_{i,0} - 2X^h_{i,1} + .5X^h_{i,2}) \tag{5.3}$$

and the functions ψ^h_j were discretized by simple injection.

With this discretization the discrete solution is expected to have second order accuracy. That is,

$$\begin{array}{c} \|X^h - X\| \le \beta_x h^2 \\ \|u^h - u\| \le \beta_u h^2 \end{array} \tag{5.4}$$

where β_x, β_u are constant independent of h (they depend on high order derivatives of the state variable). It is enough to solve the discrete problem to an accuracy satisfying

$$\begin{array}{c} \|\tilde{X}^h - X^h\| \ll \|X^h - X\| \\ \|\tilde{u}^h - u^h\| \ll \|u^h - u\| \end{array} \tag{5.5}$$

Here the tilde quantities represent the current numerical approximations to the solution of the discretized problem. The quantities $\|X^h - X\|, \|u^h - u\|$ are called discretization errors.

In all the examples reported below the algorithm described in section 5 was used with the following parameters: $\nu_1 = 2, \nu_2 = 1$. The restriction operators for both the residuals and the full solutions $R^k_x, R^k_p, \bar{R}^k_x, \bar{R}^k_p$ were the 9-point full weighting, that is

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{5.6}$$

The interpolation operators P_x^{k+1}, P_p^{k+1} were the standard bi-linear interpolation, Π_x^k, Π_p^k the bi-cubic interpolation and $\bar{R}_u^k = R_u = I$, the identity operator. The mesh size on level k was $2^{-(k+1)}$, k=1 being the coarsest level. Results for 3-FMG-V(2,1) cycle are given. In each table results for the L_2 residuals r_x, r_p, r_u of the state, costate and the control equations are given. Their exact definition is given by

$$\|r_x\|_k^2 = \sum_{i,j=1}^{N_k} ((g_x^k)_{i,j} + (B_k u^k)_{i,j} - (A_k x^k)_{i,j})^2 h_k^2 \tag{5.7}$$

$$\|r_p\|_k^2 = \sum_{i,j=1}^{N_k} ((g_p^k)_{i,j} - (C_k^* C_k x^k)_{i,j} - (A_k^* p^k)_{i,j})^2 h_k^2 \tag{5.8}$$

$$\|r_u\|_k^2 = \sum_{l=1}^{q} ((g_u^k)_l - (B_k^* p^k)_l)^2 \tag{5.9}$$

where the scaling of h_k^2 in these definitions is used so that these norms approximate the continuous norms and so residuals on different levels can be compared.

1. Distributed Control

In the distributed control case we have worked with the following problem

$$\min_u \int_\Gamma (\frac{\partial x}{\partial n} - d)^2 d\sigma \tag{5.10}$$

$$\begin{gathered} \triangle x = \sum_{i=1}^{q} u_i \psi_i \\ x|_{\partial\Omega} = 0. \\ \psi_j = sin(n_j \pi \xi_1) sin(m_j \pi \xi_2) \qquad n_j, m_j \text{ integers,} \end{gathered} \tag{5.11}$$

Table 1 shows the results for a one-dimensional control case, that is, $q = 1$ where $m_1 = n_1 = 1$, i.e. a smooth case. This is the simplest of all cases and is given here mainly for reference. Residuals of the state, costate and control equations are given separately. The error in the control (relative to the true differential error) is given as well. Observe that a 1-FMG-V cycle gives solutions to the levels of discretization errors on fine levels, as $\|u - u_{exact}\|$ reaches its minimum in essentially one cycle. The $O(h^2)$ behavior of the error is clear from the results, reflecting the order of the scheme used.

Table 2 shows the results of a similar experiment in which the control is three dimensional ($q = 3$) with $n_1 = n_3 = 1, m_1 = m_2 = 1, n_2 = m_3 = 2$. The behavior here is similar to the previous case. The $O(h^2)$ is evident also here.

2. Boundary Control

The boundary control case problem was

$$\min_u \int_\Gamma (\frac{\partial x}{\partial n} - d)^2 d\sigma \tag{5.12}$$

$$\begin{aligned} \triangle x &= 0 \\ x|_{\partial\Omega} &= \sum_{i=1}^{q} u_i \psi_i \\ \psi_j &= sin(n_j \pi \xi_1)|_{\partial\Omega} \quad n_j \text{ integer,} \end{aligned} \tag{5.13}$$

Tables 3 and 4 shows results for this problem. The first with $q = 1, n_1 = 1$, the other with $q = 2, n_1 = 1, n_2 = 2$. Also in this case the results are basically optimal. a solution to the levels of discretization errors are obtained in 1-FMG-V(2,1). The $O(h^2)$ convergence toward the differential solution is evident in both cases. Observe that the initial residuals on finer levels start are smaller by a factor which is close to four. This is typical to examples with smooth solution and proper FMG interpolation operators Π_x, Π_p.

3. Pointwise Control

The pointwise control case problem was

$$\min_u \int_\Gamma (\frac{\partial x}{\partial n} - d)^2 d\sigma \tag{5.14}$$

$$\begin{aligned} \triangle x &= \sum_{i=1}^{q} u_i \psi_i \\ x|_{\partial\Omega} &= 0. \\ \psi_j &= \delta(\xi - \xi_j), \xi_j \in \Omega \end{aligned} \tag{5.15}$$

Tables 5,6 and 7 show results for non-smooth control. Results are given for one two and three delta functions, i.e., $q = 1, 2, 3$. The location of the delta function is given in each table. Here the solution is less smooth than before. Still the results show essentially the same behavior as in the smooth case. Observe that the initial residuals on the different levels are of the same order, reflecting the non-smoothness of the solution. This behavior cannot be improved by using a higher interpolation in the refinement stage of the FMG algorithm, although local relaxation in the vicinity of the singularities can improve the results we have not experimented with such ideas here.

In some of the experiments it can be observed that one of the three residuals shown is increased in some of the cycles. This is reasonable since the three residuals are coupled and it is only their sum which goes down in each cycle. The results presented by all tables clearly demonstrate the effectiveness of the method developed here which aimed at the full optimization problem, therefore leading to one-shot solution of these problems. 1-FMG-V cycle is basically enough to reach below the levels of discretization errors.

References

[ADN] S. Agmon, A. Douglis, L. Nirenberg: Estimates Near The Boundary for Solutions of Elliptic Partial Differential Equations Satisfying General Boundary Conditions. I. Comm. Pur. Appl. Math. Vol xii, 623-727 (1959) II. Comm. Pur. Appl. Math. Vol xvii, 35-92 (1964).

[B1] A. Brandt: Multilevel Adaptive Solutions to Boundary Value Problems, Math. Comp. 31 (1977), pp 333-390.

[B2] A. Brandt: Multigrid Techniques: 1984 Guide, with applications to Fluid Dynamics. Available as GMD studien Nr. 85, GMD-AIW, Postfach 1240, D-5205, St. Augustin1, W. Germany, 1984.

[BT] A. Brandt and S. Ta'asan : Multigrid method for slightly indefinite and nearly singular problems. *Multigrid Methods II*, Lecture Notes in Mathematics 1228, Springer Verlag, 1985.

[T1] S. Ta'asan : Multigrid methods for stability Problems. J. of Sci. Comp. Vol 3. No. 3 1988.

[T2] S. Ta'asan : Multigrid methods for locating singularities in bifurcation problems. SIAM J. Stat. and Sci. Comp. Vol 1 No. 1 1990.

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.337e−06	.121e−05	.397e−09	.8166
2	1	.591e−02	.121e+00	.415e−03	.589e−01
	2	.104e−03	.609e−02	.975e−04	.612e−01
	3	.216e−03	.373e−02	.485e−04	.568e−01
	4	.592e−04	.996e−03	.183e−04	.581e−01
	5	.250e−04	.446e−03	.737e−05	.576e−01
3	1	.860e−03	.522e−01	.319e−04	.174e−01
	2	.648e−03	.292e−02	.209e−06	.149e−01
	3	.121e−04	.136e−03	.755e−07	.147e−01
4	1	.125e−03	.357e−01	.434e−05	.329e−02
	2	.324e−04	.132e−02	.371e−06	.305e−02
	3	.407e−04	.683e−04	.294e−07	.301e−02

Table 1: $q = 1, n_1 = 1, m_1 = 1$

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.246e−06	.110e−05	.374e−08	.253e+00
2	1	.887e−02	.207e+00	.654e−03	.174e+00
	2	.378e−02	.592e−01	.200e−03	.117e+00
	3	.169e−02	.258e−01	.103e−03	.994e−01
	4	.237e−03	.359e−02	.271e−04	.100e+00
	5	.905e−04	.130e−02	.908e−05	.102e+00
3	1	.246e−02	.110e+00	.772e−04	.427e−01
	2	.418e−03	114e−01	.222e−04	.317e−01
	3	.113e−03	.294e−02	.755e−05	.293e−01
4	1	.380e−03	.618e−01	.835e−05	.858e−02
	2	.595e−04	.316e−02	.110e−05	.630e−02
	3	.383e−04	.167e−03	.166e−06	.618e−02

Table 2: $q = 3, n_1 = n_3 = 1, m_1 = m_2 = 1, n_2 = m_3 = 2$

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.631e−06	.566e−07	.000e+00	.179e+00
2	1	.466e−01	.122e+01	.126e−02	.212e−01
	2	.159e−01	.379e+00	.123e−03	.617e−01
	3	.521e−02	.126e+00	.142e−04	.477e−01
	4	.177e−02	.426e−01	.161e−05	.524e−01
	5	.597e−03	.144e−01	.183e−06	.509e−01
3	1	.826e−02	.480e+00	.431e−07	.139e−01
	2	.184e−03	.146e−01	.332e−10	.133e−01
	3	.668e−04	.641e−03	.200e−10	.133e−01
4	1	.158e−02	.264e+00	.689e−06	.225e−02
	2	.338e−03	.199e−01	.285e−08	.279e−02
	3	.273e−03	.253e−02	.895e−11	.275e−02

Table 3: $q = 1, n_1 = 1$

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.631e−06	.800e−06	.296e−13	.334e+00
2	1	.354e+00	.786e+01	.288e−01	.149e−01
	2	.206e+00	.388e+01	.896e−02	.150e+00
	3	.120e+00	.229e+01	.307e−02	.754e−01
	4	.706e−01	.134e+01	.106e−02	.117e+00
	5	.415e−01	.792e+00	.367e−03	.930e−01
3	1	.523e−01	.250e+01	.924e−03	.392e−01
	2	.106e−01	.526e+00	.446e−04	.298e−01
	3	.212e−01	.104e+00	.168e−05	.281e−01
4	1	.120e−01	.126e+01	.394e−05	.528e−02
	2	.628e−03	.668e−01	.958e−08	.600e−02
	3	.274e−03	.466e−02	.313e−10	.594e−02

Table 4: $q = 2, n_1 = 1, n_2 = 2$.

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.477e−06	.853e−07	.279e−08	.370e+00
2	1	.485e+00	.172e+01	.363e−02	.536e−01
	2	.395e−01	.167e+00	.332e−03	.863e−01
	3	.320e−01	.310e−01	.132e−03	.572e−01
	4	.931e−02	.131e−01	.447e−04	.656e−01
	5	.364e−02	.445e−02	.166e−04	.624e−01
3	1	.341e+00	.791e−01	.115e−03	.114e−01
	2	.861e−02	.277e−01	.118e−04	.128e−01
	3	.248e−02	.510e−02	.258e−05	.138e−01
4	1	.574e+00	.332e−01	.364e−05	.208e−02
	2	.132e−01	.112e−02	.770e−07	.272e−02
	3	.587e−03	.200e−03	.883e−07	.275e−02

Table 5: $q = 1, \xi_1 = (.5, .5)$

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.211e−05	.921e−05	.347e−06	.864e+00
2	1	.619e+00	.437e+01	.363e−01	.390e+00
	2	.236e+00	.223e+01	.256e−01	.181e+00
	3	.147e+00	.144e+01	.166e−01	.536e−01
	4	.927e−01	.921e+00	.105e−01	.425e−01
	5	.598e−01	.591e+00	.677e−02	.909e−01
3	1	.391e+00	.389e+01	.148e−01	.537e−01
	2	.389e−01	.200e+00	.160e−02	.373e−01
	3	.138e−01	.336e−01	.523e−03	.312e−01
4	1	.660e+00	.361e+00	.587e−03	.673e−02
	2	.168e−01	.187e−01	.405e−04	.511e−02
	3	.708e−03	.120e−02	.116e−06	.509e−02

Table 6: $q = 2, \xi_1 = (.75, .5), \xi_2 = (.5, .5)$

level	cycle no.	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
1	5	.131e−05	.953e−05	.334e−05	.115e+01
2	1	.816e+00	.447e+01	.477e−01	.531e+00
	2	.293e+00	.245e+01	.319e−01	.273e+00
	3	.215e+00	.177e+01	.243e−02	.848e−01
	4	.131e+00	.119e+01	.162e−01	.345e−01
	5	.110e+00	.883e+00	.131e−01	.127e+00
3	1	.396e+00	.412e+01	.155e−01	.793e−01
	2	.431e−01	.214e+00	.236e−02	.640e−01
	3	.244e−01	.476e−01	.103e−02	.531e−01
4	1	.683e+00	.373e+00	.132e−02	.168e−01
	2	.402e−01	.341e−01	.258e−03	.880e−02
	3	.452e−02	.533e−02	.309e−04	.788e−02

Table 7: $q = 3, \xi_1 = (.75, .5), \xi_2 = (.5, .5)\xi_3 = (.25, .5)$

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-187497 ICASE Report No. 91-2		

4. Title and Subtitle	5. Report Date
"ONE SHOT" METHODS FOR OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS I: FINITE DIMENSIONAL CONTROL	January 1991
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Shlomo Ta'asan	91-2
	10. Work Unit No.
	505-90-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Submitted to SIAM Journal on Optimization

Final Report

16. Abstract

This paper discusses the efficient numerical treatment of optimal control problems governed by elliptic PDE's and systems of elliptic PDE's, where the control is finite dimensional. Distributed control as well as boundary control cases are discussed. The main characteristic of the new methods is that they are designed to solve the full optimization problem directly, rather than accelerating a descent method by an efficient multigrid solver for the equations involved. The methods use the adjoint state in order to achieve efficient smoother and a robust coarsening strategy. The main idea is the treatment of the control variables on appropriate scales, i.e., control variables that correspond to smooth functions are solved for on coarse grids depending on the smoothness of these functions. Solution of the control problems is achieved with the cost of solving the constraint equations about two to three times (by a multigrid solver). Numerical examples demonstrate the effectiveness of the method proposed in distributed control case, pointwise control and boundary control problems.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
unistep methods, distributed parameter systems, multigrid, control	64 - Numerical Analysis Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	22	A03