Comm. Korean Math. Soc. **13** (1998), No. 1, pp. 195–209

ITERATION METHOD FOR CONSTRAINED OPTIMIZATION PROBLEMS GOVERNED BY PDE

HYUNG-CHUN LEE

ABSTRACT. In this paper we present a new iteration method for solving optimization problems governed by partial differential equations. We generalize the existing methods such as simple gradient methods and pseudo-time methods to get an efficient iteration method. Numerical tests show that the convergence of the new iteration method is much faster than those of the pseudo-time methods especially when the parameter σ in the cost functional is small.

1. Introduction

In the last few years there has been an increased interest in mathematical analyses and computation of optimal control problems. In the computational aspect, optimal control problems lead to large scale computations. This interest is a direct result of improvement in computer technology. Among many control problems, the most interesting problems are those which involve complex governing equations such as the Euler, Navier-Stokes, Boussinesque, Acoustic wave and Maxwell's.

The mathematical problem is formulated as a constrained optimization problem which can sometimes be viewed as the following control problem.

$$\min_g \mathcal{J}(g, u(g)), \tag{1.1}$$

$$N(g, u) = 0, \tag{1.2}$$

Received July 3, 1997. Revised November 22, 1997.

1991 Mathematics Subject Classification: 49A22, 49B22, 65N30.

Key words and phrases: Iteration method, gradient method, pseudo-time method, constrained optimization.

This work was supported in part by BSRI 96-1441 and '96 Ajou University research fund.

where $N(g,\cdot)$ is, in general, a nonlinear partial differential operator defined on a Hilbert space H of functions defined on a domain $\Omega \subset R^N$, $N = 2$ or 3. The control variable g is assumed to be in another Hilbert space $\mathcal{U}$, for example, H^*, the dual space of H, for distributed control problems, or functions defined on the boundary $\partial\Omega$, or part of it, for boundary control problems. $\mathcal{J}(g, u(g))$ is the cost functional defined on a Hilbert space $\mathcal{U} \times H$. In many applications, $\mathcal{J}(g, u(g))$ may be defined by

$$\mathcal{J}(g, u(g)) = \sigma||g||_{\mathcal{U}}^m + ||u||_H^m, \tag{1.3}$$

where $||\cdot||_{\mathcal{U}}$ and $||\cdot||_H$ are norms on the Hilbert spaces $\mathcal{U}$ and H, respectively, and m is a positive number.

The most existing algorithms use gradient information including conjugate gradient for reaching the minimum, possibly together with preconditioners for accelerating convergence [4]. Efficient gradient calculation can be done using the adjoint method. In this approach, each optimization step requires the solution of the state and the adjoint state equations, and an efficient implementation has been achieved by using multigrid methods for both equations. Some other methods are proposed in the literature. For example, the one shot method proposed in [5] for control problems, also uses the adjoint method together with multigrid acceleration for state and adjoint state, but also includes an acceleration of the minimization process. This method can be viewed as a preconditioning of the gradient descent method where the condition number is independent of the grid size, and is of order 1. Later in [1, 7] and the references therein, the method was applied to various finite dimensional cases and extended to the infinte dimensional cases.

The necessity of using multigrid algorithms in the one shot methods is certainly disadvantage since in many engineering applications the underlying solvers do not use multigrid methods. This drawback has led to inquires in other directions.

The solution when using the adjoint method is an intersection point of three hypersurfaces describing the state equation, the adjoint equation and the optimality condition equation (together forming the necessary conditions of the minimization problem), see Fig. 1. The adjoint method can be viewed as marching on the intersection of the hypersurfaces corresponding to state and adjoint state variables in the direction of the intersection with the control hypersurfaces(Path A). Ta'asan [6] introduced another way to reach the intersection point of the three hypersurfaces which is

so called the pseudo-time method. Pseudo-time method is the one that marches on the optimal control hypersurface(Path B). We introduce a new iteration method which is a generalized simple gradient and pseudo-time method(Path C). The main idea of the new iteration scheme comes from the fact that the condition numbers of the matrices coressponding to state and adjoint state are much better than that of the matrix in pseudo-time method.

In section 2 we state the simple gradient method. In section 3 we introduce and compare the new iteration method with pseudo-time method for constrained optimization problems. In the final section, we give some numerical experiments to show the efficiency of our new method.

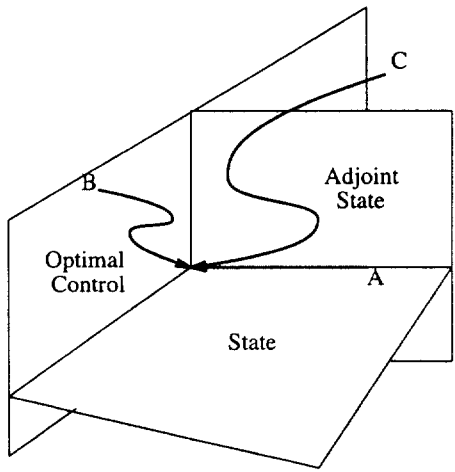

FIGURE 1. Hypersurface for state, adjoint state and optimal control

2. The simple gradient method using the adjoint equations

We consider the problem of minimizing the cost functional $\mathcal{J}(g,u)$ subject to

$$N(u,g) = 0 \quad \text{in } \Omega. \tag{2.1}$$

The first order necessary conditions are derived by introducing Lagrange multiplier and considering the augmented functional

$$\mathcal{L}(u,g,\lambda) = \mathcal{J}(g,u) + \int_\Omega N(g,u)\lambda \, d\Omega \tag{2.2}$$

where u, g and λ are the state variable, the control variable and the Lagrange multiplier(or adjoint variable), respectively.

Calculating the variation of the functional $\mathcal{L}$ with respect to the variation of the functions λ, u and g, respectively, and integrating by parts, we get

$$N(g,u) = 0 \qquad \text{State equation,} \tag{2.3}$$

$$N_u^*\lambda + \mathcal{J}_u = 0 \qquad \text{Adjoint Equation,} \tag{2.4}$$

$$N_g^*\lambda + \mathcal{J}_g = 0 \qquad \text{Optimality Condition,} \tag{2.5}$$

where $N_u^*\lambda$ and $N_g^*\lambda$ are the Fréchet derivatives of the operator N^*, which is the adjoint operator of N, with respect to u and g at λ, respectively. $\mathcal{J}_u$ and $\mathcal{J}_g$ are also the Fréchet derivatives of the functional $\mathcal{J}$ with respect to u and g, respectively. The above optimality system (2.3) - (2.5) consists of three groups of equations: the state equation for (g,u), the adjoint state equation for (λ,u), and the optimality condition for (g,u,λ). We assume the existence of solutions for both the state and the adjoint equations.

A simple gradient method for solving (2.3)-(2.5) may be described as follows:

1) choose $g^{(0)}$;

2) for $n = 0, 1, 2, ...,$ set $g^{(n+1)} = g^{(n)} - \rho_n R \dfrac{d\mathcal{J}(g^{(n)}, u(g^{(n)}))}{dg}$,

where R is the Riesz map from $\mathcal{H}^*$ to $\mathcal{H}$ and ρ_n is a sequence of positive step lengths. This algorithm converges if the second order Fréchet derivative of the cost functional is positive definite and $0 < \rho_* \leq \rho_n \leq \rho^* < \frac{2m}{M^2}$, where ρ_* and ρ^* are some positive real numbers and $M > m > 0$ are the bounds for the second order Fréchet derivative of the cost functional in a neighborhood of an optimal solution(see, e.g.,[2, 4]). The gradient $d\mathcal{J}(g,u(g))/dg$ may be computed

$$\frac{d\mathcal{J}(g,u(g))}{dg} = N_g^*\lambda(u,g) + \mathcal{J}_g(g,u(g)), \tag{2.6}$$

where $u(g)$ and $\lambda(u,g)$ are the solution of the state and adjoint equations, respectively. The quantity $-d\mathcal{J}(g,u(g))/dg$ can serve as a minimization

direction (steepest descent). Thus, by introducing the pseudo-time variable the gradient method can be viewed as the following evolution process.

$$N(g,u) = 0, \tag{2.7}$$

$$N_u^*\lambda + \mathcal{J}_u = 0, \tag{2.8}$$

$$\frac{dg}{dt} + N_g^*\lambda + \mathcal{J}_g = 0, \tag{2.9}$$

where dg/dt represents the derivative of g with respect to the pseudo-time variable introduced into the problem. The actual iteration method is obtained by replacing dg/dt with $(g^{(n+1)} - g^{(n)})/\delta t$ for a sufficiently small δt.

A gradient method with variable step lengths $\{\rho_n\}$ is given as follows.

1) choose an initial guess $g^{(0)}$;

2) for each $n \geq 1$,
solve for $u^{(n)}$ from $N(g^{(n-1)}, u^{(n)}) = 0$;
solve for $\lambda^{(n)}$ from $N_u^*\lambda^{(n)} + \mathcal{J}_u(g^{(n-1)}, u^{(n)}) = 0$
and solve for $g^{(n)}$ from $g^{(n)} = g^{(n-1)} - \rho_n(N_g^*\lambda + \mathcal{J}_g(g^{(n-1)}, u^{(n)}) = 0$.

3. Iteration method

In [6], Ta'asan suggested the new approach so called "Pseudo-time methods" which are actually iteration methods such as the Jacobi relaxation method. As we know, the solution of the minimization problem is the intersection point of the hypersurfaces defined by the state, the adjoint state, and the optimality condition equations (see Fig. 1). The pseudo-time methods can be viewed as marching on the hypersurface defined by the optimality condition equation. The construction of algorithms that march along the optimal control hypersurface and converge to the minimum of the optimization problem can be done for a wide class of problems governed by PDE. The pseudo-time approach is to look at iterative methods for the solution of the state and adjoint equations as a stable approximation to the evolution equations governed by the constrained PDE.

The construction of the method is done in two steps. In the first the stationary PDE is embedded into an evolution PDE for which the solution evolves in the optimal control hypersurfaces, and an energy estimate

ensuring convergence is derived. The second step involves a stable and consistent discretization of the pseudo-time dependent problem.

In case that the linearlized operator N_u is coersive, the optimality condition equation can be solved for the optimal control variables by keeping the state and adjoint state variables fixed. One can view one of the pseudo-time methods as an approximation to the following time dependent problem

$$\frac{d}{dt}u + N(g,u) = 0, \tag{3.1}$$

$$\frac{d}{dt}\lambda + N_u^*\lambda + \mathcal{J}_u = 0, \tag{3.2}$$

$$N_g^*\lambda + \mathcal{J}_g = 0, \tag{3.3}$$

where the last equation is essentially an extra boundary condition for the optimal control variables.

In this method, a technical difficulty which needs some explanation is related to the problem of staying on the design hypersurfaces [3]. In addition, since the state equation is nonlinear in general, we have to use some iterative methods to find a solution to the state equation. This is the main reason for developing the new approach.

We here combine the simple gradient methods and pseudo-time methods. Introducing the pseudo-time variable in the state, the adjoint state and the optimality condition, a new iteration method can be viewed as the following evolution process.

$$\frac{d}{dt}u + N(g,u) = 0, \tag{3.4}$$

$$\frac{d}{dt}\lambda + N_u^*\lambda + \mathcal{J}_u = 0, \tag{3.5}$$

$$\frac{d}{dt}g + N_g^*\lambda + \mathcal{J}_g = 0. \tag{3.6}$$

The actual iteration method is obtained by replacing dg/dt with $(g^{(n+1)} - g^{(n)})/\delta t$ for sufficiently small δt.

The algorithm for the new iteration method with variable step lengths ρ_n is given as follows:

1) choose an initial guess $g^{(0)}$;

for each $n \geq 1$,

2) solve for $u^{(n)}$ from $\frac{d}{dt}u^{(n)} + N(g^{(n-1)}, u^{(n)}) = 0$ a few steps;

3) solve for $\lambda^{(n)}$ from $\frac{d}{dt}\lambda^{(n)} + N_u^*\lambda^{(n)} + \mathcal{J}_u(g^{(n-1)}, u^{(n)}) = 0$ a few steps;
4) and solve for $g^{(n)}$ from $g^{(n)} = g^{(n-1)} - \rho_n(N_g^*\lambda + \mathcal{J}_g(g^{(n-1)}, u^{(n)}) = 0$.

Choosing $\rho_n = 1$ for $n = 1, 2, ...$ in the above algorithm, we get the pseudo-time method, and by solving the second and the third steps by direct method, we recover the simple gradient method. In this sense, we have a generalized simple gradient and pseudo-time method.

4. Numerical Results

In this section, we demonstrate the effectiveness of the new iteration method for an optimization problem governed by the Helmhortz equation. Let $\Omega = \{(x, y)|0 < x < 1, 0 < y < 1\}$. The constrained minimization problem is given by

$$\min_g \frac{1}{2}\int_\Omega (u - u_{ad})^2\, dx + \frac{1}{2}\sigma \int_\Omega g^2 dx \tag{4.1}$$

subject to

$$-\Delta u + u = g \quad \text{in } \Omega, \tag{4.2}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{4.3}$$

where u_{ad} is the desired state. To get the necessary conditions, we introduce the Lagrange multiplier λ and consider the augmented functional

(4.4)

$$\mathcal{L}(u, g, \lambda) = \frac{1}{2}\int_\Omega (u - u_{ad})^2\, dx + \frac{1}{2}\sigma \int_\Omega g^2 dx + \int_\Omega (-\Delta u + u - g)\lambda\, d\Omega.$$

Calculating the variation of the augmented functional $\mathcal{L}$ with respect to the variation of the functions λ, u and g, respectively, and integrating by parts, we obtain the following necessary conditions which are the Euler equations and boundary conditions;

$$\begin{aligned} -\Delta u + u &= g \quad \text{in} \quad \Omega, \\ u &= 0 \quad \text{on} \quad \partial\Omega, \\ -\Delta\lambda + \lambda - u &= -u_{ad} \quad \text{in} \quad \Omega, \\ \lambda &= 0 \quad \text{on} \quad \partial\Omega, \\ \sigma g - \lambda &= 0 \quad \text{in} \quad \Omega, \end{aligned} \tag{4.5}$$

which is equivalent to

$$
\begin{aligned}
-\Delta u + u - \frac{\lambda}{\sigma} &= 0 \quad \text{in} \quad \Omega, \\
u &= 0 \quad \text{on} \quad \partial\Omega, \\
-\Delta\lambda + \lambda - u &= -u_{ad} \quad \text{in} \quad \Omega, \\
\lambda &= 0 \quad \text{on} \quad \partial\Omega.
\end{aligned} \tag{4.6}
$$

First, we consider the pseudo-time embedding

$$
\begin{aligned}
\frac{d}{dt}u &= -\Delta u + u - g \quad \text{in} \quad \Omega, \\
u &= 0 \quad \text{on} \quad \partial\Omega, \\
\frac{d}{dt}\lambda &= -\Delta\lambda + \lambda - u + u_{ad} \quad \text{in} \quad \Omega, \\
\lambda &= 0 \quad \text{on} \quad \partial\Omega, \\
\sigma g - \lambda &= 0 \quad \text{in} \quad \Omega,
\end{aligned} \tag{4.7}
$$

Second, the system (4.7) is decoupled via the simple gradient method. Thus, the new-iteration method is given as follows:

1) choose an initial $g^{(0)}$;

for each $n \geq 1$,

2) solve for $u^{(n)}$ from

$$\frac{d}{dt}u^{(n)} = -\Delta u^{(n)} + u^{(n)} - g^{(n)} \quad \text{in } \Omega,$$

$$u^{(n)} = 0 \quad \text{on } \partial\Omega;$$

3) solve for $\lambda^{(n)}$ from

$$\frac{d}{dt}\lambda^{(n)} = -\Delta\lambda^{(n)} + \lambda^{(n)} - u^{(n)} + u_{ad} \quad \text{in } \Omega,$$

$$\lambda^{(n)} = 0 \quad \text{on } \partial\Omega;$$

4) and solve for $g^{(n+1)}$ from

$$g^{(n+1)} = g^{(n)} - \rho_n\left(g^{(n)} - \frac{\lambda^{(n)}}{\sigma}\right) \quad \text{in } \Omega.$$

In our simulation, we set $u_{ad} = \sin\pi x \sin\pi y$. The computations were performed on a 16×16 grid. The results for the direct method are shown in Fig. 2 and the values of cost functionals obtained are in Table 1.

σ	$\int_\Omega \lvert u-u_{ad}\rvert^2 d\Omega$	$\sigma \int_\Omega \lvert g\rvert^2 d\Omega$	$\mathcal{J}(u,g)$
10^{-2}	0.1579	1.053×10^{-4}	0.1580
10^{-3}	0.0195	1.2993×10^{-4}	0.0196
10^{-5}	3.7181×10^{-6}	2.4808×10^{-6}	6.1988×10^{-6}

TABLE 1. The values of cost functionals

Now we fix the parameter $\sigma = 10^{-3}$. We performed various numerical simulations for different σ's. In the pseudo-time method the smaller σ, the more iterations are needed. In the new iteration method, the number of iterations is independent of σ. However, we need to adjust the parameter ρ in the complicated manner to get the convergence for small σ(see [4]). In this paper, we report only the case of $\sigma = 10^{-3}$. In Fig. 3, Fig. 4 and Fig. 5, we performed 5, 10 and 15 Jacobi relaxation iterations for the state equation and the adjoint equation in each gradient step, respectively. From the computational results one can see that the convergence behavior depends on the gradient step length ρ. In our example, we obtained the stable and good convergence when $\rho = 0.2$ for various Jacobi relaxation iteration steps. The present rate of convergence could be improved by adjusting the values ρ and the number of Jacobi relaxation iterations in each gradient step. Fig. 6 shows the cost functional versus the number of iterations for fixed $\rho = 0.2$ and various numbers of Jacobi relaxation iterations. In our example, we obtained the stable and good convergence when the number of Jacobi relaxation iterations is 10 in each gradient step. Note that to obtain better convergence one may choose variable number of Jacobi relaxation iterations in each gradient step.

In Fig. 7, we present the convergence histories for the pseudo-time and the new iteration method in the logarithm of residuals. For the jacob relaxation parameter $rjac = 0.5$, we obtain the same rate of convergence in the full equations and state equation. For $rjac = 0.96$, we obtain about 3 times faster rate of convergence than those of the case $rjac = 0.5$ for state equation but the pseudo-time method was unstable. Fig. 8 shows that the convergence histories for the pseudo-time method and the new iteration method in the logarithm of functionals for the case that $\sigma = 10^{-3}$, $\rho = 0.2$ and the iteration steps are 10. In the new iteration

method, 19.785 seconds of cpu time were needed while 44.263 seconds of cpu were needed in the pseudo time method for 300 iteration steps. After 300 iteration steps, the functionals converged to the "exact" value of the functional 0.0196 in Table 1 with the approximation error 10^{-5}. In Fig. 9, we present the convergence histories for the pseudo-time and the new iteration method in the logarithm of absolute errors $E^{(n)}$ which is defined by

$$E^{(n)} = \left| \mathcal{J}(g^{(n)}, u^{(n)}) - \mathcal{J}(\hat{g}, \hat{u}) \right|, \tag{4.8}$$

where the "exact" optimal solution $(\hat{g}, \hat{u})$ is obtained by the direct method. All computations in this paper were carried out on the SUN UltraSparc 2 workstation at Ajou University.

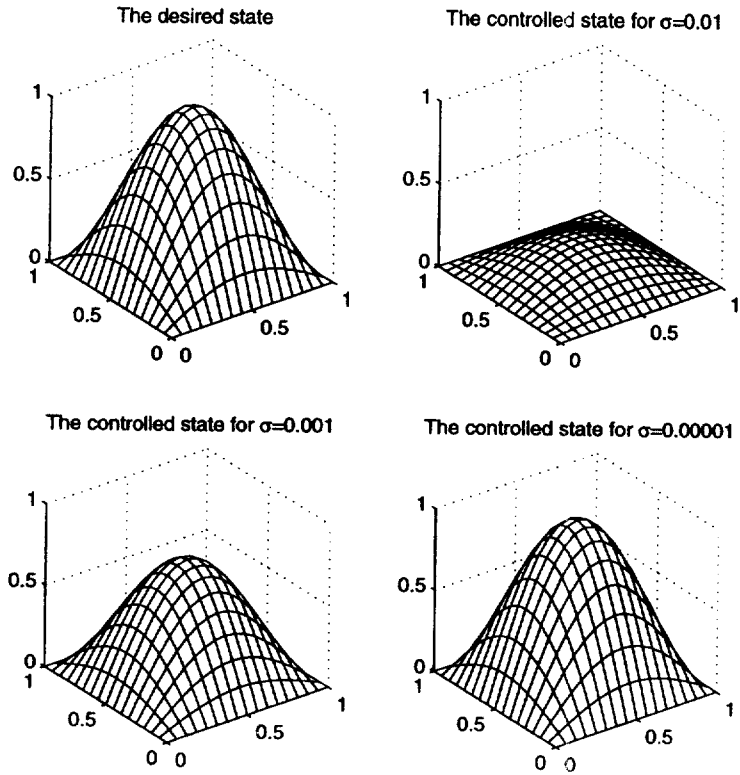

FIGURE 2. Desired state and controlled states.

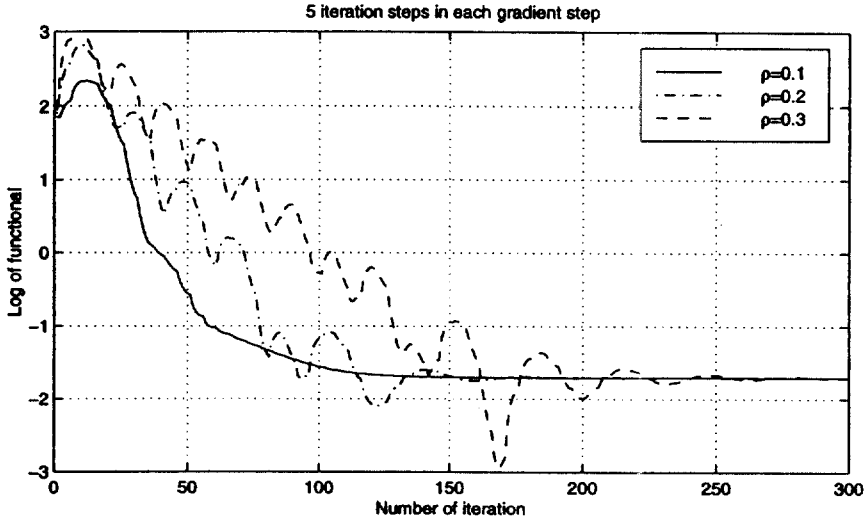

FIGURE 3. Logarithm of the functional versus the number of iterations.

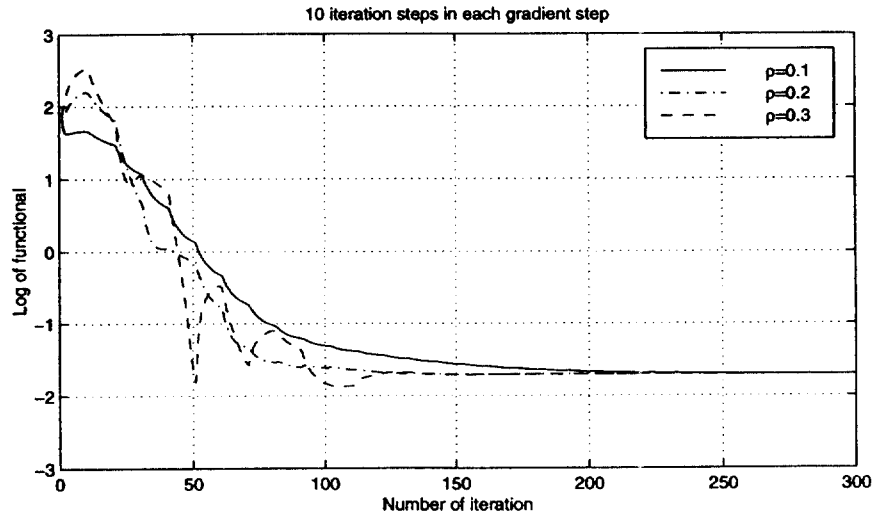

FIGURE 4. Logarithm of the functional versus the number of iterations.

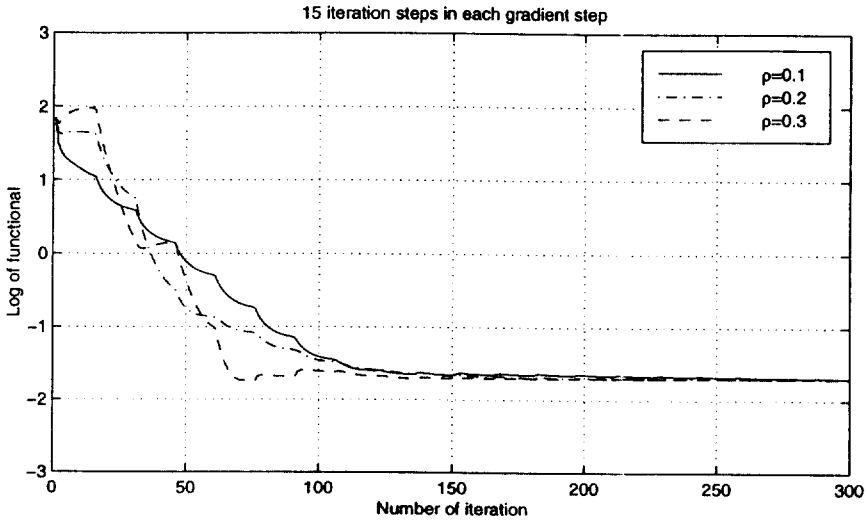

FIGURE 5. Logarithm of the functional versus the number of iterations.

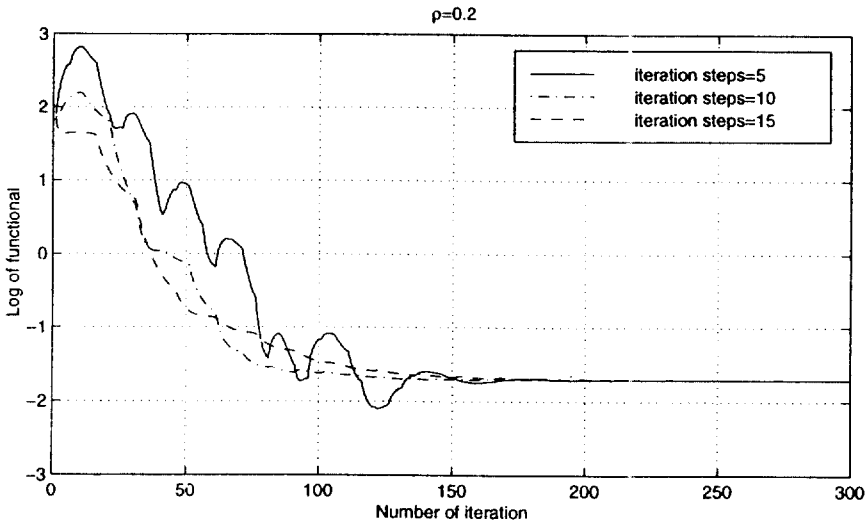

FIGURE 6. Logarithm of the functional versus the number of iterations.

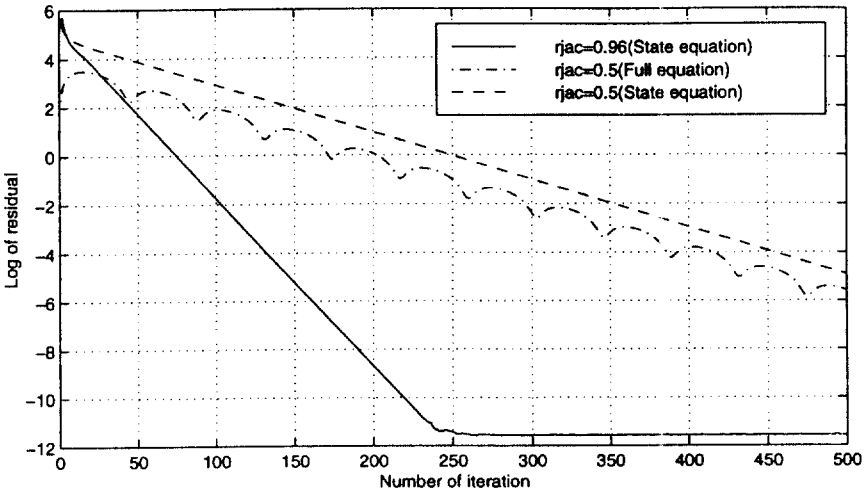

FIGURE 7. Convergence history for the pseudo-time and new iteration methods. Logarithm of residuals.

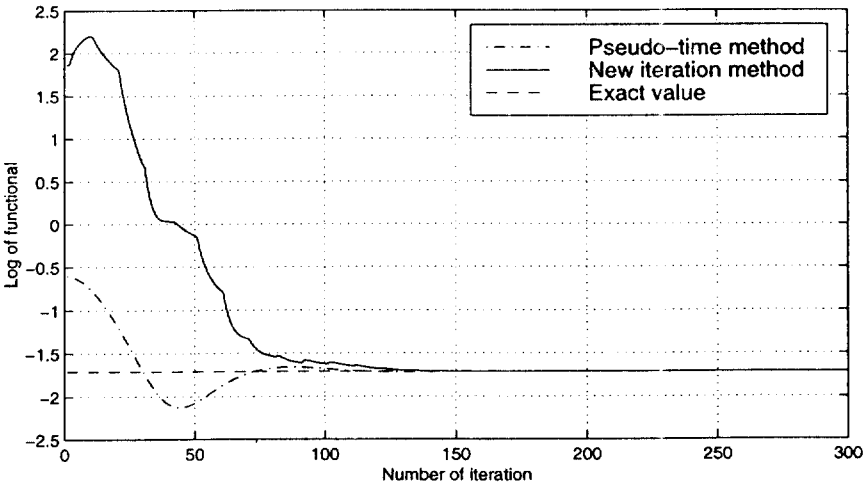

FIGURE 8. Convergence history for the pseudo-time and new iteration methods. Logarithm of functionals.

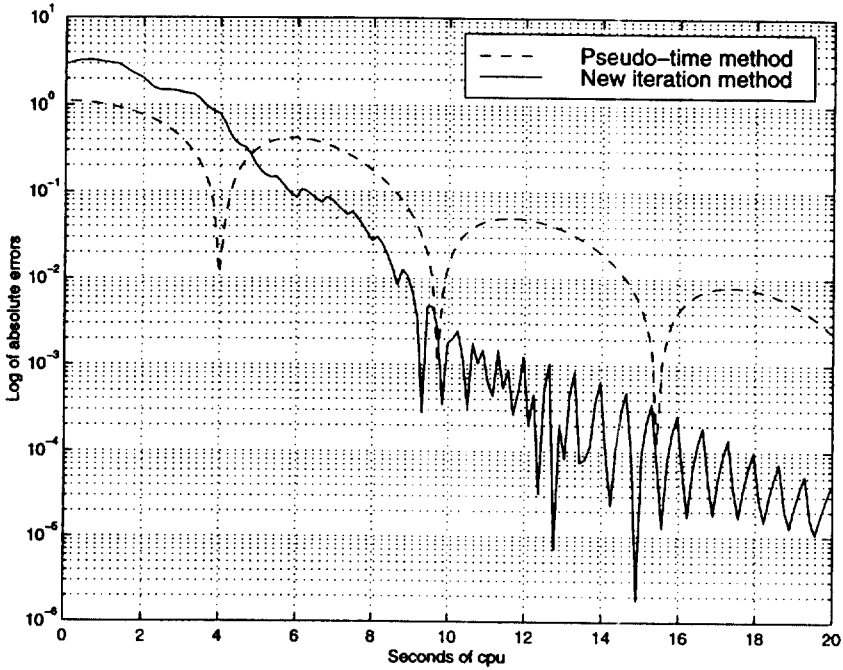

FIGURE 9. Convergence history for the pseudo-time and new iteration methods. Logarithm of absolute errors.

References

[1] Arian and S. Ta'asan, Multigrid one shot methods for optimal control problems : Infinte Dimensional Control. ICASE Report No. 94-52 (1994).

[2] P. Ciarlet, Introduction to numerical linear algebra and optimization, Cambridge (1989).

[3] A. Iollo, G. Kuruvila and S. Ta'asan, Pseudo-time methods for optimal shape design using the Euler equations, ICASE Report No. 95-59 (1995).

[4] Max D. Gunzburger and Hyung C. Lee, Analysis, approximation, and computation of a coupled solid/fluid temperature control problem, Compt. Methods Appl. Mech. Engng. **118** (1994), 133-152.

[5] S. Ta'asan, One-shot method for optimal control of distributed parameter systems I: Finte dimensional control, ICASE Report No. 92-2 (1991).

[6] S. Ta'asan, Pseudo-time methods for constrained optimization problems governed by PDE, ICASE Report No. 95-32 (1995).

[7] S. Ta'asan, M. Salas and G. Kuruvila, Aerodynamics Design and optimization in one shot, In 30th Aerospace Sciences Meeting and Exhibit, AIAA 92-005 (1992).

Department of Mathematics
Ajou University
Suwon 442-749, Korea
E-mail: hclee@minmax.ajou.ac.kr