드래그 감소를 위한 유체의 최적 엑티브 제어 및 최적화 알고리즘의 개발(3)

- 트루 뉴턴법을 위한 정식화 개발 및 유체의 3차원 최적 엑티브 제어

Optimal Active-Control & Development of Optimization Algorithm for Reduction of Drag in Flow Problems(3)

- Construction of the Formulation for True Newton Method and Application to Viscous Drag Reduction of Three-Dimensional Flow

박 재 형†
Bark, Jai-Hyeong

(논문접수일 : 2007년 8월 3일 ; 심사종료일 : 2007년 11월 5일)

요 지

저자는 기존의 연구에서 대용량-비선형성을 가지는 유체의 최적화를 수행하기 위해 몇 가지 강력한 방법들을 제시한 바 있다. 즉, 최적화 과정에서 수렴성을 높이기 위해 step by step기법을 사용하였고, 또한 수렴속도를 높이기 위하여 최적화 이터레이션 과정에서 얻어지는 민감도정보를 이용하여 시스템 평형방정식의 해석을 위한 좋은 초기치를 제공하는 방법과, 평형방정식을 구속조건으로 사용하는 동시기법(simultaneous technique)에서 착안하여 해석과 최적화 수렴 판정치를 조작하는 방법을 제시한 바 있다. 그러나 그들 기법은 기본적으로 유사뉴턴법에 기본을 두고 있다. 현재까지 최적화에서 SQP기법을 사용할 때는 정확한 헤시안 매트릭스의 유도가 매우 까다롭고 힘들기 때문에 유사뉴턴법을 사용하고 있는 실정이다. 그러나 3차원 문제와 같이 더욱 큰 용량의 문제를 위해서는 진정한 의미에서의 뉴턴법, 트루 뉴턴법(true Newton method)을 사용할 필요가 있다. 본 연구에서는 트루 뉴턴법을 사용하기 위해 헤시안 매트릭스의 정확치를 얻는 과정을 유도하고 이를 기본으로 트루 뉴턴법을 이용한 최적화 루틴을 만들었다. 그리고 이를 3차원 문제에 적용하여 그 효과를 검증하였다.

핵심용어 : 트루 뉴턴법, 헤시안 매트릭스의 정확치, SQP 기법, 삼차원 Navier-Stokes 유체, 드래그

Abstract

We have developed several methods for the optimization problem having large-scale and highly nonlinear system. First, step by step method in optimization process was employed to improve the convergence. In addition, techniques of furnishing good initial guesses for analysis using sensitivity information acquired from optimization iteration, and of manipulating analysis/optimization convergency criterion motivated from simultaneous technique were used. We applied them to flow control problem and verified their efficiency and robustness. However, they are based on quasi-Newton method that approximate the Hessian matrix using exact first derivatives. However solution of the Navier-Stokes equations are very cost, so we want to improve the efficiency of the optimization algorithm as much as possible. Thus we develop a true Newton method that uses exact Hessian matrix. And we apply that to the three-dimensional problem of flow around a sphere. This problem is certainly intractable with existing methods for optimal flow control. However, we can attack such problems with the methods that we developed previously and true Newton method.

Keywords : *true Newton method, exact Hessian matrix, SQP method, three-dimensional Navier-Stokes flow, drag*

† 책임저자, 정회원 · 목원대학교 건축학부 교수
Tel: 016-464-6250 ; Fax: 042-822-5260
E-mail: jhbark@mokwon.ac.kr

• 이 논문에 대한 토론을 2008년 2월 29일까지 본 학회에 보내주시면 2008년 4월호에 그 결과를 게재하겠습니다.

1. Introduction

We have developed several methods for the optimization problem having large-scale and highly nonlinear system(Bark, 2007). First, step by step method in optimization process was employed to improve the convergence. In addition, techniques of furnishing good initial guesses for analysis using sensitivity information acquired from optimization iteration, and of manipulating analysis/optimization convergency criterion motivated from simultaneous technique were used. We applied them to flow control problem and verified their efficiency and robustness.

However, they are based on quasi-Newton method that approximate the Hessian matrix using exact first derivatives. However solution of the Navier-Stokes equations are very cost, so we want to improve the efficiency of the optimization algorithm as much as possible. Thus we develop a true Newton method that uses exact Hessian matrix. And we apply that to the three-dimensional problem of flow around a sphere. This problem is certainly intractable with existing methods for optimal flow control. However, we can attack such problems with the methods that we developed previously and true Newton method.

2. Problem definition

We consider the three-dimensional problem of flow around a sphere immersed in a stream of fluid. Our final aim is to reduce the drag force on the body by controlling the velocities on the surface of the body. To solve this problem, we use true Newton method for optimization with the several techniques that we developed previously.

As an objective function, we will use the rate of dissipation of energy due to viscosity, which is equivalent to the drag force on the body, in the case of an external incompressible Navier-Stokes flow.

Therefore, our optimization problem to be solved can be expressed as:

$$\textit{minimize} \quad 2\mu\int_{\Omega}[\mathbf{D(u)}:\mathbf{D(u)}]d\Omega \qquad (1)$$

$$\textit{subject to} \quad -\mu\Delta\mathbf{u}+\rho(\mathbf{u}\cdot\nabla)\mathbf{u}-\frac{1}{\epsilon}\nabla(\nabla\cdot\mathbf{u})=\mathbf{0} \qquad (2)$$

where μ is the dynamic viscosity, ρ is the density, and $\mathbf{u}$ is the flow velocity, ε is the penalty parameter. and $\mathbf{D(u)}=(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})/2$. The symbol : represents the scalar product of two tensors.

3. Optimization

Our optimization problem was expressed equation (1) and (2). However, it is not straightforward to apply SQP methods to flow optimization problems. Therefore we pursued a decomposition of the problem into the state space and the control space, i.e. solve the equation (2), at each optimization iteration for the state variables(u) given values of the control variables(b). Thus, we have eliminated the state equations, i.e. equation (2), from the constraint set, and we have eliminated the state variables from the set of optimization variables.

Therefore our optimization problem can be simply expressed as:

$$\textit{minimize} \quad 2\mu\int_{\Omega}[\mathbf{D(u(b))}:\mathbf{D(u(b))}]d\Omega \qquad (3)$$

$$\text{where,} \quad \mathbf{D(u(b))}=(\nabla\mathbf{u(b)}+\nabla\mathbf{u}^{\mathrm{T}}\mathbf{(b)})/2$$

Now, the dimension of the optimization problem is now greatly reduced, and the constraints are eliminated. As a result of this decomposition, the state variables become an implicit function of the control variables, the implicit function being the flow solution itself. The implicit dependence of state variables on control variables requires a sensitivity analysis, to find the derivatives of the objective function with respect to the control variables.

3.1 First-order sensitivity analysis(Bark, 2007)

Here, we will show how to obtain the gradient of objective function and constraints with respect to the control variables, taking into account the implicit dependence of the state variables on the controls through the discrete Navier-Stokes equation.

Let F denote either the objective or a constraint function. Here, F depends explicitly on the control variables $\mathbf{b}$ as well as implicitly on them through the state variables $\mathbf{u}$:

$$F = F(\mathbf{u}(\mathbf{b}), \mathbf{b}) \tag{4}$$

The relationship between $\mathbf{u}$ and $\mathbf{b}$ is dictated by the discrete form of the Navier-Stokes equations, which we refer to as the state equations:

$$\mathbf{h}(\mathbf{u}(\mathbf{b}), \mathbf{b}) = 0 \tag{5}$$

The total derivative of the objective with respect to the control variables can be found by applying the chain rule,

$$\frac{DF}{D\mathbf{b}} = \frac{\partial F}{\partial \mathbf{b}} + \frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{d\mathbf{b}} \tag{6}$$

Here $\partial F/\partial \mathbf{b}$ and $\partial F/\partial \mathbf{u}$ can be readily found from the expression for the discrete form of the objective function. The only unknown is $d\mathbf{u}/d\mathbf{b}$, but this vector can be found by applying the implicit function theorem for the state equations, yielding

$$\frac{D\mathbf{h}}{D\mathbf{b}} = \frac{\partial \mathbf{h}}{\partial \mathbf{b}} + \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{d\mathbf{b}} = 0 \tag{7}$$

Thus, the sensitivity of the velocity field with respect to control variables can be found by solving the linear system

$$\frac{\partial \mathbf{h}}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{d\mathbf{b}} = -\frac{\partial \mathbf{h}}{\partial \mathbf{b}} \tag{8}$$

This is a linear system with coefficient matrix that is just the Jacobian of the state equations with respect to the state variables, and thus is a component of any Newton solver. Therefore, the same linear solver that is at the heart of the flow simulation is used for sensitivity analysis, with the exception of the right-hand side vectors.

The matrix of these vectors, $\partial \mathbf{h}/\partial \mathbf{b}$, can be readily found analytically. It is seen that this first-order sensitivity analysis requires the solution of a system of equations for as many right-hand sides as there are control variables.

3.2 Construction of the exact Hessian matrix for true Newton method

In this section, we will present the necessary expressions for constructing the exact Hessian matrix of the object to develop the true Newton method.

Let's consider the total derivative of the objective function with respect to the control bariable b_i:

$$\frac{DF}{Db_i} = \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_i} = g_i \tag{9}$$

The objective function is dependent on the state variables(u) and the control variables(b). However the gradient of the objective function is dependent on $d\mathbf{u}/d\mathbf{b}$ as well as the state variables(u) and the control variables(b). Therefore the second derivative of the objective function with respect to the control variables b_i can be expressed as:

$$\frac{Dg_i}{Db_j} = \frac{\partial g_i}{\partial b_j} + \frac{\partial g_i}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_j} + \frac{\partial g_i}{\partial \left(\frac{d\mathbf{u}}{db_i}\right)} \cdot \frac{d\left(\frac{d\mathbf{u}}{db_i}\right)}{db_j} \tag{10}$$

Therefore the Hessian matrix can be obtained by:

$$\begin{aligned}
\mathbf{H}_{ij} = \frac{D^2F}{Db_j^2} &= \frac{\partial}{\partial b_j}\left(\frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_i}\right) \\
&\quad + \frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_i}\right) \cdot \frac{d\mathbf{u}}{db_j} \\
&\quad + \frac{\partial}{\partial\left(\frac{d\mathbf{u}}{db_i}\right)}\left(\frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_i}\right) \cdot \frac{d^2\mathbf{u}}{db_j db_j} \\
&= \frac{\partial^2 F}{\partial b_i \partial b_j} + \frac{\partial d}{\partial b_j}\left(\frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_i}\right) \\
&\quad + \frac{\partial}{\partial d\mathbf{u}}\left(\frac{\partial F}{\partial b_i}\right) \cdot \frac{d\mathbf{u}}{db_j} + \frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d^2\mathbf{u}}{db_i db_j} \\
&\quad + \frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial F}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{db_i}\right) \cdot \frac{d\mathbf{u}}{db_j} \\
&= \frac{\partial^2 F}{\partial b_i \partial b_j} + \frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial F}{\partial b_j}\right) \cdot \frac{d\mathbf{u}}{db_i} + \frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial F}{\partial b_i}\right) \cdot \frac{d\mathbf{u}}{db_j}
\end{aligned}$$

$$+\frac{\partial^2 F}{\partial \mathbf{u}^2}\cdot\frac{d\mathbf{u}}{db_i}\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial F}{\partial \mathbf{u}}\cdot\frac{d^2\mathbf{u}}{db_i db_j} \quad (11)$$

Here $\partial^2F/\partial b_i\partial b_j$, $\partial F/\partial \mathbf{u}$, $\partial/\partial\mathbf{u}(\partial F/\partial b_i)$, $\partial/\partial\mathbf{u}(\partial F/\partial b_j)$, and $\partial F/\partial\mathbf{u}$ can be easily found analytically. Furthermore, $d\mathbf{u}/db_i$, $d\mathbf{u}/db_j$ are available from first-order sensitivity analysis. The remaining unknown is $d^2\mathbf{u}/db_idb_j$. This is derived below;

Let us reconsider the implicit function theorem for the state equations with respect to the control variables b_i:

$$\frac{D\mathbf{h}}{Db_i}=\frac{\partial \mathbf{h}}{\partial b_i}+\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_i}=0=\mathbf{t}_i \quad (12)$$

The state equations are dependent on the state variables(u) and the control variables(b). However the gradient of the these equations is dependent on not only the state variables(u) and the control variables(b) but also on $d\mathbf{u}/d\mathrm{b}$. Therefore the second derivative of the state equations with respect to the control variables b_j can be expressed as:

$$\frac{Dt_i}{Db_j}=\frac{\partial t_i}{\partial b_j}+\frac{\partial t_i}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial t_i}{\partial\left(\frac{d\mathbf{u}}{db_i}\right)}\cdot\frac{d\left(\frac{d\mathbf{u}}{db_i}\right)}{db_j}=0 \quad (13)$$

Taking the derivative of equation (12) with respect to b_j, and using equation (13), gives:

$$\begin{aligned}\frac{D^2\mathbf{h}}{Db_iDb_j}&=\frac{\partial}{\partial b_j}\left(\frac{\partial \mathbf{h}}{\partial b_i}+\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_i}\right)\\&+\frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{h}}{\partial b_i}+\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_i}\right)\cdot\frac{d\mathbf{u}}{db_j}\\&+\frac{\partial d}{\partial\left(\frac{d\mathbf{u}}{db_i}\right)}\left(\frac{\partial \mathbf{h}}{\partial b_i}+\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_i}\right)\cdot\frac{d^2\mathbf{u}}{db_{idb_j}}\\&=\frac{\partial^2\mathbf{h}}{\partial b_idb_j}+\frac{\partial}{\partial b_j}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_i}\right)+\frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{h}}{\partial b_i}\right)\cdot\frac{d\mathbf{u}}{db_j}\\&+\frac{\partial}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d\mathbf{u}}{db_i}\right)\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d^2\mathbf{u}}{db_idb_j}\\&=\frac{\partial^2\mathbf{h}}{\partial b_i\partial b_j}+\frac{\partial}{\partial b_i}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial}{\partial b_j}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)\cdot\frac{d\mathbf{u}}{db_i}\\&+\frac{\partial^2\mathbf{h}}{\partial \mathbf{u}^2}\cdot\frac{d\mathbf{u}}{db_i}\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\cdot\frac{d^2\mathbf{u}}{db_idb_j}=0\end{aligned} \quad (14)$$

Therefore,

$$\frac{d^2\mathbf{u}}{db_idb_j}=-\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)^{-1}\left[\frac{\partial^2\mathbf{h}}{\partial b_i\partial b_j}+\frac{\partial^2\mathbf{h}}{\partial \mathbf{u}^2}\cdot\frac{d\mathbf{u}}{db_i}\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial}{\partial b_i}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)\cdot\frac{d\mathbf{u}}{db_j}+\frac{\partial}{\partial b_j}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)\cdot\frac{d\mathbf{u}}{db_i}\right] \quad (15)$$

Here $\partial\mathbf{h}/\partial\mathbf{u}(=\mathbf{J})$ is the Jacobian of the state equations, and $\partial^2\mathbf{h}/\partial b_i\partial b_j$, $\partial/\partial b_i(\partial\mathbf{h}/\partial\mathbf{u})$, $\partial/\partial b_j(\partial\mathbf{h}/\partial\mathbf{u})$ and $\partial^2\mathbf{h}/\partial\mathbf{u}^2$ can be readily found analytically. Furthermore, we already know $d\mathbf{u}/db_i$, $d\mathbf{u}/db_j$ from the first-order sensitivity analysis. Therefore all quantities on the right-hand side of (equation) are known, and $d^2\mathbf{u}/db_idb_j$ can be readily computed. Formally, this requires the solution of a system of equations for m^2 right-hand sides (where m is the number of control variables), all having the same coefficient matrix, namely the Jacobian J.

However, note that:

$$\frac{d^2\mathbf{u}}{db_idb_j}=\frac{d^2\mathbf{u}}{db_jdb_i} \quad (16)$$

so that the Hessian matrix is symmetric. Thus, only $m(m+1)/2$ right-hand side solutions are required. It would appear that only one *LU* factorization would be required. However, this *LU* factorization was computed for the first-order sensitivity expressions. Therefore, no *LU* factorizations are required, and (if a direct method is being used) second-order sensitivity analysis comes at essentially no additional cost beyond analysis and first-order sensitivity.

With the availability of exact curvature information, we have a true Newton method for optimization and thus we can expect quadratic convergence (to go along with the quadratic convergence for the analysis problem). Although the expressions for second-order sensitivity are quite cumbersome, we shall see in the next section that they are more efficient and they permit us to solve three dimensional problems, and are thus well-worth the effort.

3.3 SQP method for solving the optimization problem using the true Newton method

To solve the nonlinear programming problems, the sequential quadratic programming(SQP) method is generally regarded as the best method. This method is based on the iterative formulation and solution of quadratic programming subproblems, obtains subproblems by using a quadratic approximation of the Lagrangian. That is:

$$minize \quad \frac{1}{2}\mathbf{p}_k^T \mathbf{B}(\mathbf{b}_k, \lambda_k)\mathbf{p}_k + \nabla f(\mathbf{b}_k)^T \mathbf{p}_k$$

where B_k is a positive definite approximation of the Hessian of the Lagrangian function. b_k represents the current iterate points. Let p_k be the solution of the subproblem. ∇f_k the gradient of the objective function, i.e. $\nabla f_k = DF/D\mathbf{b}_k$. A line search is used to find a new point b_{k+1}, where

$$b_{k+1} = b_k + \alpha p_k \quad \alpha \in (0, 1]$$

such that a merit function will have a lower function value at the new point. The augmented Lagrange function is used here as the merit function. When optimality is not achieved, B_k is updated according to the BFGS formula.

However, we constructed formulations of the exact Hessian matrix for the objective function in this study. Therefore, we can use the true Newton method now.

Following is summary of the our optimization process using true Newton method:

1) do analysis (obtain u_k knowing b_k solving equation (2))
2) do first-order sensitivity analysis to get.
3) get the exact Hessian matrix H_k.
4) check convergence criterion : if $\|DF/D\mathbf{b}_k\| \le \gamma$, then terminate; otherwise go to step 5
5) find p_k by solving

$$\mathbf{H}_k \mathbf{p}_k = -\frac{DF}{D\mathbf{b}_k}$$

6) $\mathbf{b}_{k+1} = \mathbf{b}_k + \mathbf{p}_k$
7) go to step 1

4. Numerical Examples

4.1 Comparison between quasi-Newton and true Newton method by an infinite cylinder-2D

In the previous paper, we compared our proposed methods based on quasi-Newton ideas(Bark, 2007). In this section, we will compare them with the true Newton method, that uses the exact Hessian information with the same example(an infinite cylinder-2D).

First, we check the convergence rates to make sure that we are obtaining the rates predicted by theory. We tested this for Reynolds number 50 problem. Table 1 shows the error in objective as a function of iteration number. F^k implies the value of the objective function at iteration k, and F^* the optimal value of the objective function. The Methods took respectively 22 and 17 optimization iterations. From the table we can see the quadratic convergence in the Newton method beginning at iteration 13. This is reflected in a doubling of the number of correct digits in the objective function. On the other hand, the quasi-Newton method does not exhibit quadratic convergence-the number of correct digits is not doubling at each iteration.

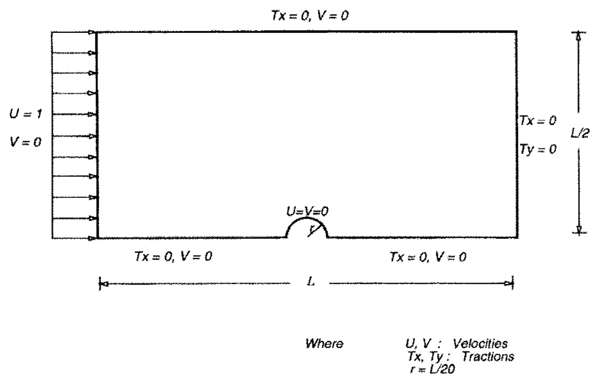

Fig. 1 Flow around cylinder-2D

Table 1 Convergence of objective function

iteration	$\|\|F^k-F^*\|\|$	
	quasi-Newton	true Newton
1	1.054819945559	1.054819945559
2	1.001993115835	102.14488238395
3	1.069074860480	36.821083070181
4	0.624116592811	8.871273994225
5	0.771807215336	2.180216446137
6	0.176850308049	0.352447331178
7	0.104132535399	0.065997521112
8	0.062021852623	0.853941779573
9	0.054375192028	0.224662374922
10	0.044056495017	0.101465531176
11	0.034139295156	0.075240322190
12	0.027133988604	0.067200974037
13	0.017465180067	0.029023064229
14	0.010363979280	0.004379458242
15	0.005358094972	0.000091362669
16	0.004469789884	0.000000079406
17	0.003188945188	0.000000000000
18	0.001266680895	
19	0.000226711696	
20	0.000011248955	
21	0.000000447843	
22	0.000000000000	

Table 2 Number of *LU* factorizations (*Re*=500)

	by quasi-Newton	by true Newton
NA-SQP	401	276
T1-NA-SQP	318	230
T2-NA-SQP	261	139
T3-NA-SQP	111	74

Table 3 Total CPU time (*Re*=500)

	by quasi-Newton	by true Newton
NA-SQP	84.7	68.1
T1-NA-SQP	67.2	61.1
T2-NA-SQP	55.0	40.1
T3-NA-SQP	23.4	19.8

We also applied *NA-SQP*~*T3-NA-SQP* in Newton form for *Re*=500 problem by step size *Re*=50.

Tables 2 and Table 3 show the number of *LU* factorizations, and CPU time of these methods. We see an improvement over quasi-Newton methods of about 40~50% in the number of *LU* factorizations. The improvement over quasi-Newton is about 20% in total CPU time.

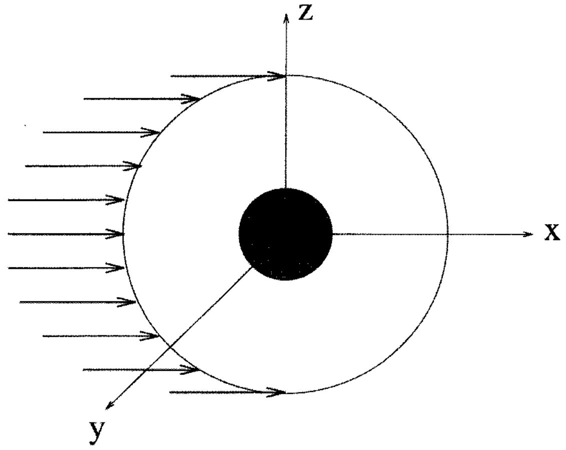

Fig. 2 flow around sphere

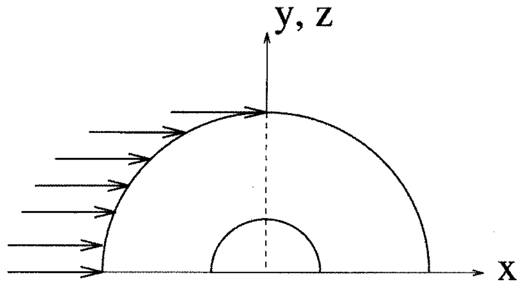

Fig. 3 x-y (x-z) plane

4.2 Flow around a sphere-3D

In this section, we solve the three-dimensional problem of flow around a sphere (Figure 2). This problem is certainly intractable with existing methods for optimal flow control. The methods of this study make it possible to attack such problems. Here we model the flow with the steady state Navier-Stokes equations. Flow around a sphere is generally considered steady-state until Reynolds number 130. Therefore we will use *Re*=130 for our problem.

We solve only a quarter of the flow domain by exploiting symmetry about midplanes. A problem description is shown in Figure 3. Figure 3 shows the x-y or x-z plane. In the x-y plane, the velocities in the z direction (w) are zero, and in the x-z plane, the velocities in the y direction (v) is zero. Figure 4 shows the y-z section.

The mesh in the x-y plane (or x-z plane) is given in Figure 5. This mesh appears to be course, but we found this is sufficient for *Re*=130. The volume

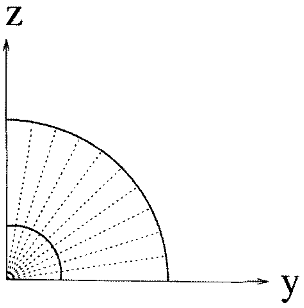

Fig. 4 y-z plane

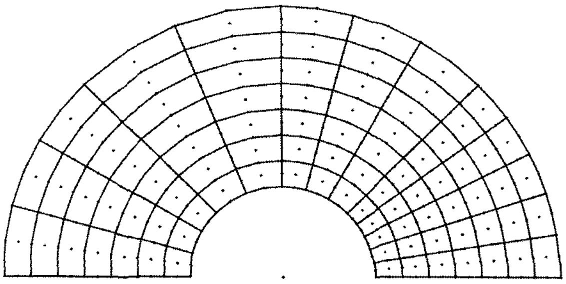

Fig. 5 mesh

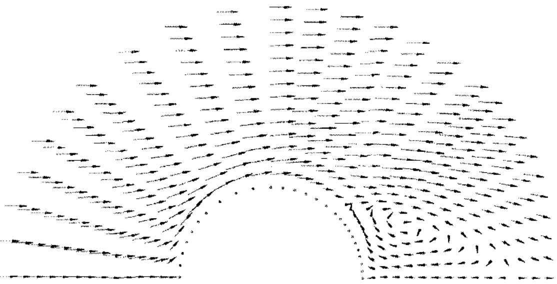

Fig. 6 velocity vector without any control

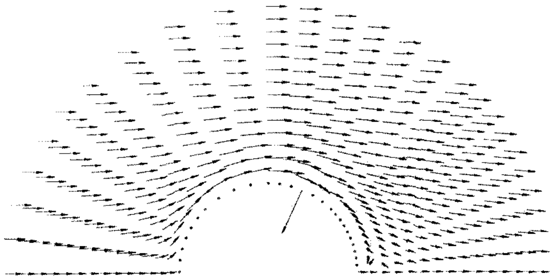

Fig. 7 Optimal velocity vectors on plane having holes-case 1

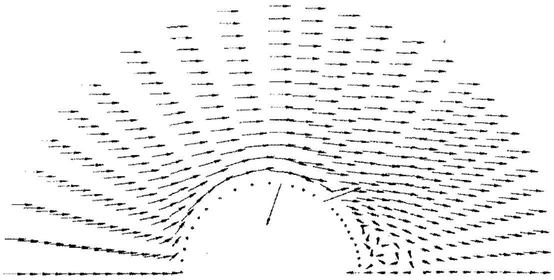

Fig. 8 Optimal velocity vectors on plane having holes-case 2

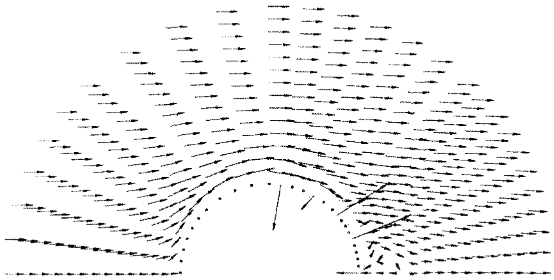

Fig. 9 Optimal velocity vectors on plane having holes-case 3

mesh is obtained by rotating the x-y plane about the x axis, in 9°s increments. This gives us a total of 11 planes. Altogether 4155 nodes and 455 iso-parametric triquadratic 27-nodes prism elements. For Gauss-Legendre numerical integration, a 3×3×3 scheme is used.

Figure 6 shows the velocity vectors around the sphere without any control. The same flow pattern occurs on any plane in the x direction, because of the axisymmetry of the problem. The flow separation and recirculation behind the cylinder are evident from the figures.

To apply boundary velocity controls, we choose six planes by selecting every other planes from x-y through x-z plane. Three cases are tested. The cases correspond to one, three, and five holes on each plane. The total number of control variables is 17, 33, and 65. The locations of the holes are evident from Figures 7~9.

Figures 7~9 show the optimal velocity vectors on planes having holes, for each of the three cases. Figures 10~12 show the optimal velocity vectors on planes that have no hole. From the figures, Case 1(a single hole on each plane) appears to have a velocity field closest to potential flow. However, from Table 4, we see that it has the highest optimal objective function value. Case 3 has the lowest objective, with Case 2 between. This result is expected, as we expect better performance from a large number of suction/injection holes. However, the flowfields are not as esthetically pleasing as their two dimensional counterparts. However, the solutions computed are at least locally optimal with

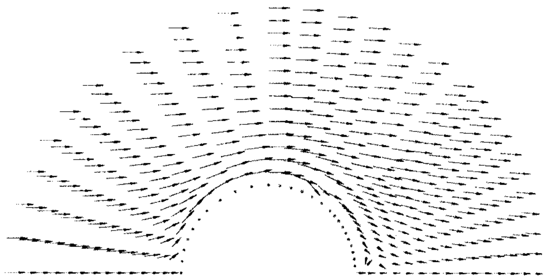

Fig. 10 Optimal velocity vectors on plane do not having holes-case 1

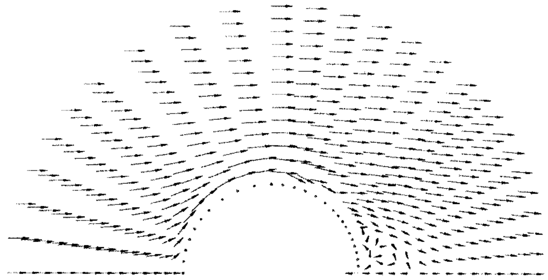

Fig. 11 Optimal velocity vectors on plane do not having holes-case 2

respect to the dissipation function.

5. Conclusion

We considered the three-dimensional problem of flow around a sphere immersed in a stream of fluid. Our final goal was to reduce the drag force on the body by controlling the velocities by suction and injection on the surface of the body. To solve this kind of problems, we have proposed several powerful methods. They are based on quasi-Newton method. In this study, we developed a true Newton method that uses exact Hessian matrix with application to the three-dimensional problem of flow around a sphere. This problem is certainly intractable with existing methods for optimal flow control. We are unaware of any other attempt to solve a three-dimensional problem. However, we attacked such problems with the true Newton method including the feature of *T3-NA-SQP*. Although the expressions for second-order sensitivity are quite cumbersome, they are more efficient and they permit us to solve three dimensional problems, and are thus well-worth the effort.

In the process of solving three-dimensional problem, we encountered a limit of about 13000 state variables, due to memory. Beyond this size, where undoubtedly most industrial-scale flow control problems life, iterative solvers will be required(of course, we may increase memory more).

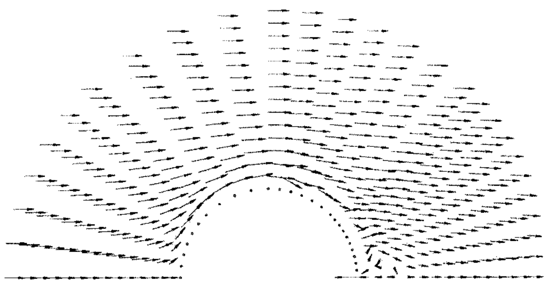

Fig. 12 Optimal velocity vectors on plane do not having holes-case 3

Table 4 Optimal objective function values

	Case 1	Case 2	Case 3	No control
Objective	1.7299	1.5513	1.4300	5.7755

References

박재형 (2007) 드래그 감소를 위한 유체의 최적 엑티브 제어 및 최적화 알고리즘의 개발 (1)-대용량, 비선형 유체의 최적화 문제를 위한 알고리즘 및 테크닉의 개발, 한국전산구조공학회논문집, 20(5), pp.661~670.

박재형 (2007) 드래그 감소를 위한 유체의 최적 엑티브 제어 및 최적화 알고리즘의 개발 (2)-개발된 기법의 검증 및 드래그 감소를 위한 유체의 최적 액티브 제어, 한국전산구조공학회논문집, 20(5), pp.671~680.

박재형, 이리형 (1996) 구조체의 위상학적 최적화를 위한 비선형 프로그래밍, 한국전산구조공학회논문집, 9(3), pp.163~178.

박재형, 홍순조 (2002) Navier-Stokes 유체의 최적 제어, 한국전산구조공학회논문집, 15(4), pp.661~674.

박재형, 홍순조 (2002) Navier-Stokes 유체의 최적제어를 위한 SQP기법의 개발, 한국전산구조공학회논문집, 15(4), pp.675~691.

박재형, 홍순조, 이리형 (1996) Multiple- loading condition을 고려한 구조체의 위상학적 최적화, 한국전산구조공학회논문집, 9(3), pp.179~186.

Batchelor, G.K. (1967) *An introduction to Fluid Dynamics*, Cambridge Univ. Press, Cambridge, UK.

Biegler, L.T., Nocedal, J., Schmid, C. (1995) A reduced Hessian method for large-scale constrained optimization, *SIAM. J. Optim.* 5, 314.

Cramer, E.J., Dennis, J.E., Frank, P.D., Lewis, R.M., Shubin, G.R. (1994) Problem formulation for multidisciplinary optimization, *SIAM J. Optim.* 4, 754.

Davis, T.A. (1993) *Users' Guide for the Unsymmetric Pattern Multifrontal Package (UMFPACK)*, Tech. Rep. TR-93-020, CIS Dept., University of Florida, Gainesville, FL.

Davis, T.A., Duff, I.S. (1993) *An Unsymmetic Pattern Multifrontal Method for Sparse LU Factorization*, Tech. Rep. TR-93-018, CIS Dep., University of Florida, Gainesville, FL.

Farthat, C., Chen, P.S. (1994) Tailoring domain decomposition methods for efficient parallel coarse grid solution and for system with many right hand sides, in Domain Decomposition Methods in Science and Engineering, *Contemporary Mathematics, Vol. 180, American Mathematical Society, Providence*, RI.

Gabay, D. (1982) Reduced *Quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Math, Programming Study 16, 18.

Gad-el-Hak, M. (1989) Flow control, *Appl. Mech.* Rev. 42, 261.

Ghattas, O., Orozco, C.E. (1997) A parallel reduced Hessian SQP method for shape optimization, in Multidisciplinary Design Optimization, *State-of-the-Art, edited by N. Alexandrov and M. Hussaini*, p.133, SIAM, Philadelphia.

Gill, P.E., Murray, W., Wright, M.H. (1981) *Practical Optimization*, Academic Press, New York

Gunzburger, M.D. (1989), *Finite Element Methods for Viscous Incompressible Flow*, Academic Press, San Diego.

Gunzburger, M.D. (1995) Flow Control, *IMA Volumes in Mathematics and Its Applications*, 68, Springer-Verlag, Berlin/New York.

Gunzburger, M.D., Hou, L.S., Svobodny, T.P. (1993) *Optimal control and optimization of viscous, incompressible flows, in Incompressible Computational Fluid Dynamics*, edited by M.D. Gunzburger and R.A. Nicolaides, Cambridge Univ. Press, Cambridge, UK, 1993.

Haftka, R.T., Gurdal, Z., Kamat, M.P. (1990) *Elements of Structural Optimization*, Kluwer Academic, Dordercht/Norwall, MA, 1990.

Moin, P., Bewley, T. (1994) Feedback control of turbulence, App. Mech. Rev. 47,s3,

Prandtl, L., Tietjens,, O.G. (1934) *Applied Hydro-and Aeromechanics*, Dover, New York, p.81, 1934.

Xie, Y. (1991) Reduced Hessian Algorithms for solving Large-Scale Equality Constrained Optimization Problem, Ph. D. thesis, University of Cololado, Boulder, Department of Computer Science.