Shape optimization with computational fluid dynamic simulation

M. El-Sayed, T. Sun & J. Berry
Department of Mechanical Engineering,
Kettering University, Flint, MI, USA.

Abstract

Computational Fluid Dynamics (CFD) has proven to be useful in supporting design in many industrial applications, where fluid flow is important. CFD analyses alone are quite useful in predicting the flow pattern for a given flow design. However, these results typically provide limited insight into changes that are required to truly optimize the design.

This paper deals with the shape optimization of fluid flows using numerical optimization techniques. By linking a commercial optimization code to a developed CFD code, a CFD optimization tool was developed. To study the applicability of the developed tool and its ability to produce results with reasonable CPU time, the shape optimization of an S duct with circular cross sections was studied with one, two, and four design variables. The developed optimization approach along with the optimization and CPU results are discussed.

1 Introduction

Computer Fluid Dynamics (CFD) has proven to be useful in supporting design in many industrial applications, where fluid flow is important. CFD analyses alone are quite useful in predicting the flow pattern for a given flow channel design. However, these results typically provide limited insight into changes that are required to truly optimize the design.

Transactions on the Built Environment vol 67, © 2003 WIT Press, www.witpress.com, ISSN 1743-3509

Coupling of CFD analyses with numerical optimization since the early 1970's has mostly focused on aerospace design problems [1-3]. One automotive application is the work by Choi and Kim [4], who coupled CFD simulations with numerical optimization techniques to design a low noise automotive cooling fan.

The high computational expense of real-world flow simulations demands that the optimal design be computed with relatively few CFD solutions. To accomplish this, various optimization algorithms may be used. In this paper, a CFD-based optimization methodology is developed by the integration of three software programs. The numerical optimization code *VisualDOC*, the CFD computational code *Chaos* and the surface geometry generation code *Prime* are combined to produce the CFD shape optimization capability [5]. The optimization capability is tested for its applicability and CPU time using an S-shaped outlet duct.

2 Analysis and optimization

In CFD design, the initial design is parameterized with respect to input values that may be adjusted to obtain better flow conditions. In general, these design variables may include any quantity that is used as input to the CFD analysis. Once the design is parameterized and initial design variable values are chosen, the optimization process begins.

In this work, the implementation of the CFD-based optimization methodology is achieved by the integration of three major software programs: *VisualDOC*, numerical optimization software, is used to solve the constrained and unconstrained optimization problems. *Prime* is used to produce the surface geometry of an intake port for the set of design variables supplied by the optimization program, and to show the result from the CFD code. *Chaos* is used to do the CFD calculation. The design optimization process of Figure 1 is achieved by linking the above tools through the use of a main C computer program. The main program calls Chaos and Prime to evaluate the objective function. *VisualDOC* then computer new design variable values while searching for the optimum. Some additional computer programs are developed to facilitate the communication between these tools. *VisualDOC* works with Third-party programs. Figure 1, shows the overall optimization strategy. The generation of *VisualDOC* optimized design data follows the following steps:

- Start with a set of initial design parameters.
- Generate analysis data first by using *CHAOS*. Analyze the problem using this analysis data, and make the analysis output available.
- Make the input file for *VisualDOC* by using the data from the analysis output.
- Call *VisualDOC* to get the output file for this iteration.
- Modify the analysis input by using the output file of *VisualDOC*.
- Run third-party program to get the analysis output again, until the whole process converges.

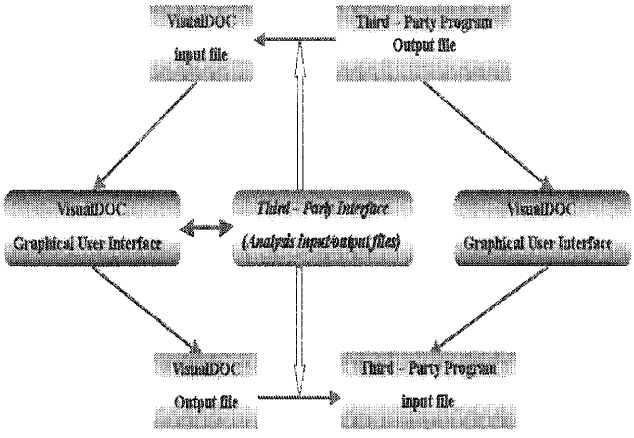


Figure 1

3 Test cases

An S – shape duct with circular cross sections was used as a test case. The path – line curve in Figure 2, is defined in the $X-Z$ plane and is specified to be normal to its inlet and outlet cross sections at (X_3,Z_3) and (X_0,Z_0), respectively. The upper and lower limits on the design variable are chosen so that they account for both packaging constraints and robust construction of the geometry. Four design variables can be used to define the design space. Variables x_1, x_2 are used to control the center-line path and R_1, R_2 are used to control the cross section for upper and lower sections. The upper and lower limits on the design variable are chosen so that they account for both packaging constraints and robust construction of the geometry. In Table 1, the upper and lower limits of the design variable are specified.

The steady airflow in the S-shape duct is calculated by using CFD code, *CHAOS 2.0*. The model is divided into three regions: the main S-shape duct and the inlet and outlet extensions. The extensions are used to allow the flow to develop at the inlet and recover at the outlet so that the flow in the S-shape region is not affected by the applied boundary conditions. The mesh has 3200 elements, 13211 nodes and 1442 constraints. At the outlet, a zero gradient boundary condition is applied to the remainder of the outlet parameters.

The main design goal is to find an optimum shape for the S – shape duct so that the pressure losses due to both skin friction and turning of the flow in the bends are minimized. Three test cases of shape optimization of the duct were conducted subject to constraints representing the limitations on the packaging

space as well as other geometric restrictions imposed to ensure geometric robustness.

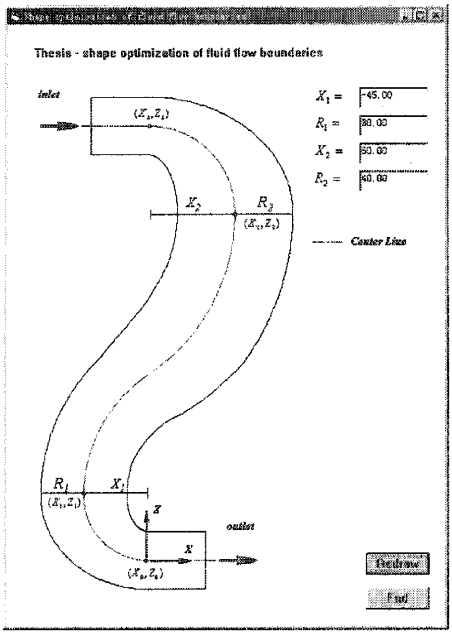


Figure 2

Design Variable	Lower Limit	Upper Limit
$X_1(mm)$	-100.00	-20.00
$R_1(mm)$	20.00	50.00
$X_2(mm)$	20.00	100.00
$R_2(mm)$	20.00	50.00

Table 1

3.1 One design variable

For this case with x_1 is the design variable, the optimization problem was formulated as:

Minimize: $\Delta P = P_{in} - P_{out}$

Subject to: $G_1 = |X_1| - 64.61 < 0$

$G_2 = 40.39 - |X_1| < 0$

Sequential Quadratic Programming (SQP) algorithm of DOT [6] was used to solve the constrained optimization problem. The constraint function G_1 represents a packaging constraint while the function G_2 represents design variable relationship that should be maintained to guarantee a robust construction of the CAD surfaces. The minimization of ΔP was also subject to the side constraints given in Table 1.

The optimization calculations were performed on an SUN workstation. The optimization process took 8 hours of CPU time to converge. A total of 3 SQP optimization design iterations were required and 21 function evaluations were performed during the optimization process. Figure 3, shows the history of the objective function ΔP, which is decreased by 46.41% and the history of the design variable. Figure 4, shows the pressure contours for the initial and final designs.

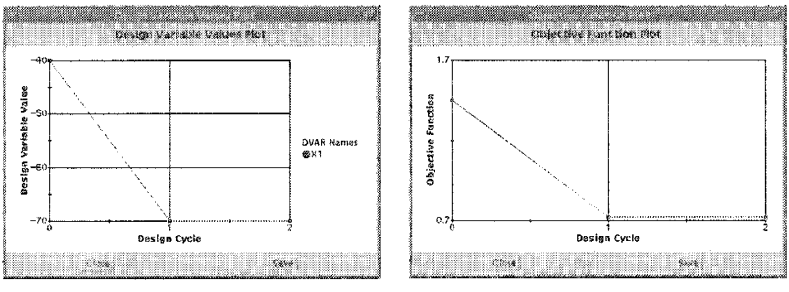


Figure 3

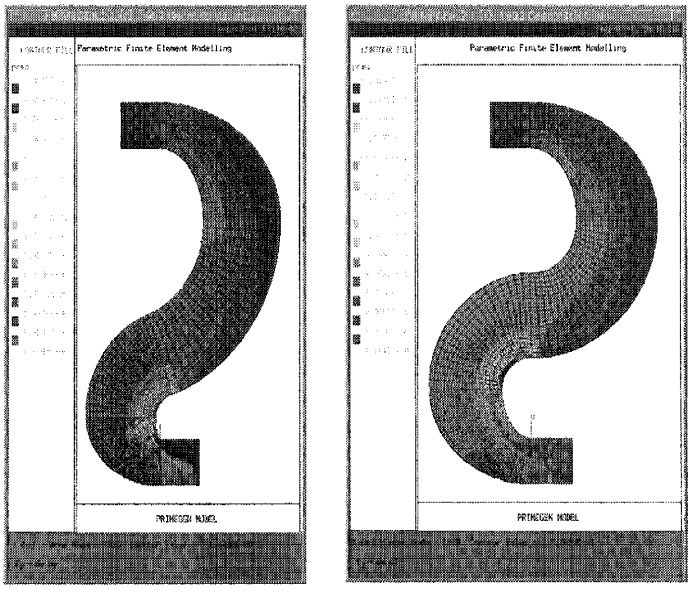


Figure 4

3.2 Two design variables

For this case with x_1 and x_2 are the design variables, the optimization problem was formulated as:

Minimize: $\Delta P = P_{in} - P_{out}$

Subject to: $G_1 = |X_1| - 64.61 < 0$

$G_2 = 40.39 - |X_1| < 0$

$G_3 = 44.43 - |X_2| < 0$

Sequential Quadratic Programming (SQP) algorithm of DOT is used to solve the constrained optimization problem. The constraint function G_1 represents a packaging constraint. Functions G_2 and G_3 represent design variable relationship that should be maintained in order to guarantee a robust construction of CAD surfaces. The minimization of ΔP is also subject to the side constraints given in Table 1, which define the lower and upper limits of the design variable. The optimization calculations were performed on an SUN workstation. The optimization process took 28.75 hours of CPU time to converge. A total of 5 SQP optimization design iterations were required and 69 function calls were performed during the optimization process. Figure 5, shows the history of the

Transactions on the Built Environment vol 67, © 2003 WIT Press, www.witpress.com, ISSN 1743-3509

objective function ΔP decreasing by 41.43% and the history of the design variables. Figure 6, shows the pressure contours for the initial and final designs.

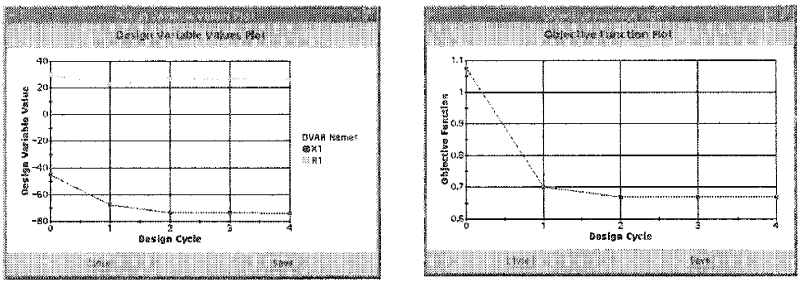


Figure 5

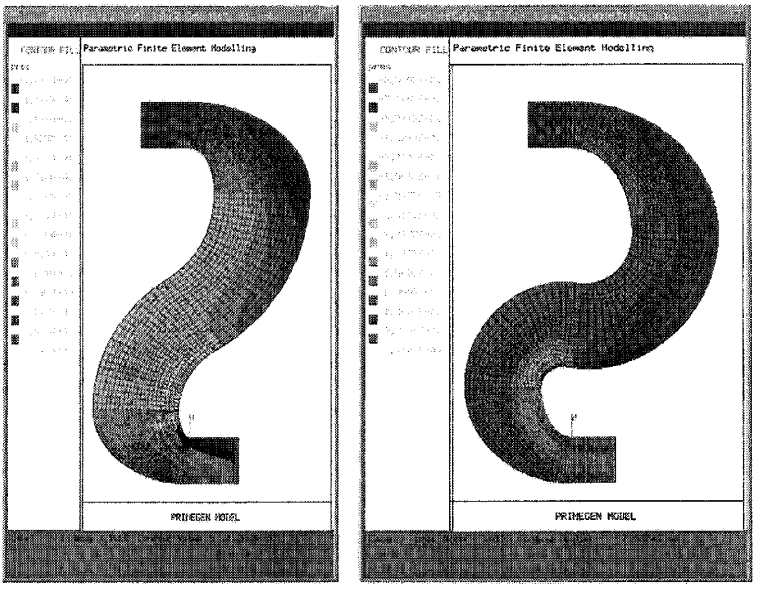


Figure 6

3.3 Four design variables

With four design variables, the optimization problem was formulated as follows:

Minimize: $\Delta P = P_{in} - P_{out}$

Subject to:

$$G_1 = |X_1| + |R_1| - 100 < 0$$

$$G_2 = |R_1| - |X_1| + 5 < 0$$

$$G_3 = |R_2| - |X_2| + 5 < 0$$

Sequential Quadratic Programming (SQP) algorithm of DOT was used to solve the constrained optimization problem. G_1 is packaging constraint while G_2 and G_3 represent design variable relationship that should be maintained in order to guarantee a robust construction of CAD surfaces. The minimization of ΔP is also subject to the side constraints given in Table 1, which define the lower and upper limits of the design variable.

The optimization calculations were performed on an SUN workstation. The optimization process took 72.5 hours of CPU time to converge. A total of 8 SQP optimization design iterations were required and 174 function calls were performed during the optimization process. Figure 7 shows the history of the objective function ΔP decreasing by 41.61% and the history of the design variables. the constraint functions and the design variables are summarized in Table 1. . Figure 8, shows the pressure contours for the initial and final designs.

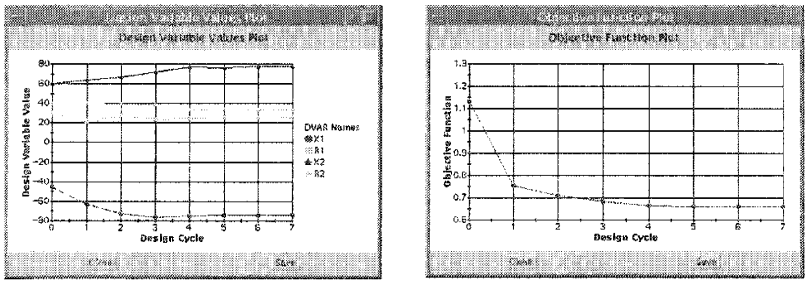


Figure 7

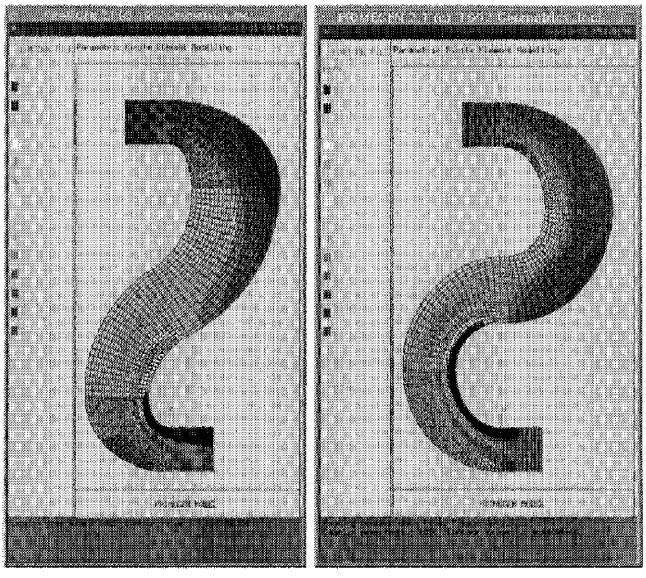


Figure 8

4 Conclusions

The main objective of this work was to develop a shape optimization tool for fluid flows using numerical optimization techniques. By linking a commercial optimization code to an in-house developed CFD code, a CFD optimization code was developed. To examine the applicability of the developed code S-shape duct optimization was conducted with one, two, and four design variables. The achieved optimization results show improvement in the objective function Δp for each optimization case. The CPU time on the SUN workstation in general increased with number of design variables from 8 hours for single design variable to 72.5 hours for four variables. These results show that is possible to use CFD and numerical optimization to achieve optimum shapes for fluid flow problems. Additional work, however, is needed to improve the computational efficiency and reduce CPU time of CFD simulation.

Transactions on the Built Environment vol 67, © 2003 WIT Press, www.witpress.com, ISSN 1743-3509

References

[1] Hicks R., and Vanderplaats G., *Design of low-speed airfoil by numerical optimization*, SAE paper 750524, 1975.
[2] Hicks R., and Vanderplaats G., *Airfoil section drag reduction at transonic speeds by numerical optimization*, SAE paper 760477, 1976.
[3] Hicks R., and Vanderplaats G., *Application of numerical optimization to the design of supercritical airfoil without drag creep*, SAE paper 770440,1977.
[4] Choi J., Kim K. and Chung D., *Numerical optimization for design optimization of automotive cooling fan,* SAE paper number 970935, 1997.
[5] Stone, T. A., Santos, J. and Haug, E. J., *An Interactive Pre-Processor for Structural Design Sensitivity Analysis and Optimization*, Computers & Structures, 34, 3, pp. 375-383,1990.
[6] Vanderplaats G., *Numerical optimization techniques for engineering design*, McGraw-Hill, Inc. 1984.