Journal of Fluids and Structures 22 (2006) 529–540

JOURNAL OF FLUIDS AND STRUCTURES

www.elsevier.com/locate/jfs

Inverse design of supersonic diffuser with flexible walls using a Genetic Algorithm

S. Ziaei-Rad[a,*], M. Ziaei-Rad[b]

[a]*Mechanical Engineering Department, Isfahan University of Technology, Isfahan, Iran*
[b]*Mechanical Engineering Department, Sharif University of Technology, Tehran, Iran*

Received 20 April 2004; accepted 22 January 2006
Available online 3 April 2006

Abstract

An efficient algorithm for the design optimization of the compressible fluid flow problem through a flexible structure is presented. The methodology has three essential parts: first the behavior of compressible flow in a supersonic diffuser was studied numerically in quasi-one-dimensional form using a flux splitting method. Second, a fully coupled sequential iterative procedure was used to solve the steady state aeroelastic problem of a flexible wall diffuser. Finally, a robust Genetic Algorithm was implemented and used to calculate the optimum shape of the flexible wall diffuser for a prescribed pressure distribution.

Keywords: Compressible flow; Optimization; Genetic Algorithm; Supersonic diffuser

1. Introduction

Aerodynamic design optimization has been an important area of research for many years. Although some of the early work in this area has been limited in applicability because of a lack of computational tools, advances in computational algorithms and computer hardware have recently fostered intense efforts aimed at aerodynamic and multi-disciplinary optimization. Shape optimization is, in fact, one of the most frequently faced problems. In fluid mechanics, the search for optimal aerodynamic shapes dates back to Newton. The search for an axisymmetric body with minimum resistance from the surrounding fluid, during its motion with constant speed parallel to the axis of symmetry, gave rise to the so-called hydrodynamic or aerodynamic shapes.

Perhaps the most widely used optimization techniques are those based on the calculation of the gradients in which a specified objective function is minimized. The gradients of the objective function with respect to the design variables are used to update the design variables in order to systematically reduce the cost function so as to arrive at a local minimum. An important step in this process is the determination of these gradients, which are also referred to as sensitivity derivatives. Several techniques have been investigated for evaluating the sensitivities for aerodynamic applications. A description of these techniques can be found in papers by Baysal and Eleshaky (1991), Huffman et al. (1993), Hou et al. (1994), Narducci et al. (1995) and in the references contained therein.

*Corresponding author. Tel.: +98 311 3915244; fax: +98 311 3912628.
E-mail address: szrad@cc.iut.ac.ir (S. Ziaei-Rad).

0889-9746/$ - see front matter
doi:10.1016/j.jfluidstructs.2006.01.006

Optimization problems with nonsmooth, nondifferentiable, highly nonlinear and many local minima cost functions are commonly encountered in many engineering applications including shape optimization. Conventional gradient-based algorithms are ineffective in these applications due to the problem of local minima or the difficulty in calculating gradients. Optimization methods that require no gradient and can achieve a global optimal solution offer considerable advantages in solving these difficult optimization problems. The need for new search algorithms, which are capable of escaping local optima, has led to the development of nontraditional search and thus stochastic optimization algorithms (Vanderplaats, 1984). The most important stochastic methods are: Genetic Algorithm (GA) (Holland, 1975; Davis, 1991), Simulated Annealing (Kirkpatrick et al., 1983; Corana et al., 1987; Ingber, 1993) and Tabu Search (Glover and Laguna, 1997).

In the mid-70 s, researchers (Hicks and Henne, 1977; Lores et al., 1979) began exploring the use of numerical optimization techniques for the design of aircraft components. However, optimization based on the GA did no get the attention of researchers until recent years. Quagliarella and Della Cioppa (1994) were among the first who used GA to optimize the airfoil, utilizing a potential-based flow solver. Vicini and Quagliarella (1997) used GA for multi-point and multi-objective airfoil applications. Obayashi et al. (1997) applied the algorithm for multi-disciplinary optimization of transonic wings. A method for aerodynamic shape optimization of airfoils and transonic wing using a real number encoding GA was proposed by Holst and Pulliam (2001). For the airfoil they used an Euler equation solver, while a nonlinear potential solver was used for the transonic wing optimization. Examples of multi-design point wing optimization using Euler and Navier–Stokes flow solvers can be found in papers by Sasaki et al. (2000) and Oyama (2000a, b).

Frank and Shubin (1992) compare three optimization-based methods for solving aerodynamic design problems of a 1-D duct flow with rigid walls. Their optimization methods are (i) the black-box method with finite-difference gradients, (ii) a modification where gradients are found by an algorithm based on the implicit function theorem, (iii) an all-at-once method where the flow and design variables are simultaneously altered. They concluded that the third approach is dramatically less expensive than the other approaches. Shubin (1995) presented both design and analysis of the same model with flexible walls. Again he used different gradient-based optimization approaches for calculating the optimum shape of the duct to a prescribed pressure.

Recently, Periaux et al. (2001) proposed a new evolutionary strategy for the multiple objective design optimization of internal aerodynamic shape operating with transonic flow. Their global shape optimization was to find a shape of a nozzle which realizes a prescribed pressure distribution on its boundary for a given flow condition. They used game theory to replace a global optimization problem by a noncooperative game based on Nash equilibrium with several players solving local constrained sub-optimization tasks. The transonic flow simulator uses a full potential solver taking advantage of domain decomposition methods and GAs for the matching of nonlinear local solutions. However, they did not consider wall flexibility.

A disadvantage of the GA approach is expense. In general, the number of function evaluation required for a GA algorithm exceeds the number required by a finite-difference-based gradient optimization (Bock, 1990; Obayashi and Tsukahara, 1997). Recently, Doorly et al. (1999, 2000, 2001), applied a parallel GA for aerodynamic design optimization. They applied the method to a 2-D wing section and discussed the advantage of the method over the sequential GA.

This paper is concerned with design optimization of a supersonic nozzle with flexible walls using a GA. First, the behavior of compressible flow in the supersonic diffuser was calculated numerically using a flux splitting method. Second, a fully coupled sequential iterative procedure was used to solve the steady state aeroelastic problem of the flexible wall diffuser. Finally, a robust GA was implemented and used for the shape optimization of the diffuser.

2. Problem description

A schematic of the physical model on coordinate system is shown in Fig. 1. The physical model consists of an axisymmetric supersonic diffuser with flexible walls. It was assumed that the diffuser has been constrained at both ends (Points A and B in Fig. 1). A steady airstream ($\gamma = 1.4$) passes through the diffuser. For simplicity, only the variation of variables in the streamwise direction is considered.

The profile of the diffuser wall is created either by an algebraic equation or by spline curves. This matter will be discussed in more detail in Section 6.

3. Fluid–solid interaction

For a complete solid–fluid interaction, three distinct gradients are necessary: (i) a flow solver, (ii) a structural analysis code and (iii) a coupling interface code. In this paper, the behavior of compressible flow in a supersonic diffuser was

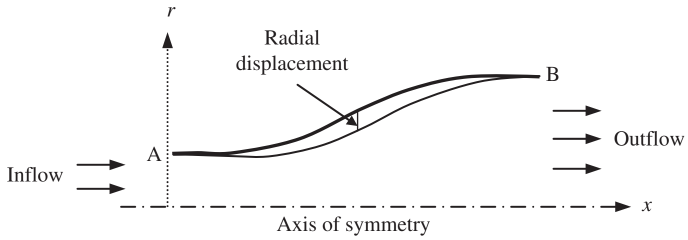


Fig. 1. Structural model of the diffuser.

studied numerically in quasi-one-dimensional form using a flux splitting method. For structural analysis, a finite element approach was used to compute the nodal displacement of the diffuser wall. Finally, a fully coupled sequential iterative procedure was used to solve the steady state aeroelastic problem of a flexible wall diffuser. Each of the above items will be described briefly.

3.1. The flow solver

The governing equations in quasi-one-dimensional flux vector form may be written as

$$\frac{\partial}{\partial t}(SQ)+\frac{\partial E}{\partial x}-H=0, \tag{1}$$

where Q is the vector of conserved variables, E is the inviscid flux vector in the streamwise direction, H denotes the source (load) vector and S is the cross-sectional area. These may be further expressed by the primary variables as

$$Q=\begin{bmatrix}\rho & \rho u & \rho e_t\end{bmatrix}^{\mathrm{T}}, \tag{2}$$

$$E=\begin{bmatrix}\rho u & \rho u^2+p & (\rho e_t+p)u\end{bmatrix}^{\mathrm{T}}, \tag{3}$$

$$H=\begin{bmatrix}0 & p\,\frac{\mathrm{d}S}{\mathrm{d}x} & 0\end{bmatrix}^{\mathrm{T}}. \tag{4}$$

In the above equations, p, ρ and u are nondimensionalized pressure, density and velocity, respectively, and e_t represents the total energy per unit mass, the quantities in the above equations are nondimensionalized using the following scheme:

$$\begin{aligned}&\rho=\frac{\bar{\rho}}{\rho_0},\quad u=\frac{\bar{u}}{u_0},\quad p=\frac{\bar{p}}{\rho_0 u_0^2},\\ &e_t=\frac{\bar{e}_t}{u_0^2},\quad a=\frac{\bar{a}}{u_0},\quad M=\frac{\bar{u}}{\bar{a}},\end{aligned} \tag{5}$$

where $\bar{\rho}$, $\bar{u}$, $\bar{p}$, $\bar{e}_t$ and $\bar{a}$ are dimensional density, velocity, pressure, total energy and sound speed, respectively, and M is the Mach number. The quantities with zero sub-index are the flow characteristics at the nozzle inlet.

To solve the above set of equations, a set of proper boundary conditions is required. At the supersonic inflow boundary, nondimensionalized variables are specified, namely $M=1.5$ (Mach number), $p=1$ and $\rho=1$. At the outlet, the flow is subsonic. Therefore, two variables will be determined from the internal computational cells using a second-order implicit extrapolation. The third one, i.e., the subsonic pressure, should be specified and that is set to $p=2.5$ for all the cases studied in this paper.

In order to accurately capture the shock wave discontinuity, a finite-difference flux vector splitting method has been used as the flow solver. In particular, we use the flux vector splitting method of Steger and Warming (1979). A steady state solution is obtained in a time-asymptotic sense, using an implicit discretization of governing equations. The computational flow field is divided into N finite-difference meshes, where N indicates the spatial intervals used in the streamwise direction. The implicit discretized form of conservation equation may be expressed as

(Steger and Warming, 1979)

$$-\left(\frac{\Delta t}{\Delta x}A^{+}_{i-1}\right)\Delta Q_{i-1}+\left[SI+\frac{\Delta t}{\Delta x}(A^{+}_{i}-A^{-}_{i})-\Delta t\,B_i\right]\Delta Q_i$$
$$+\left(\frac{\Delta t}{\Delta x}A^{-}_{i+1}\right)\Delta Q_{i+1}=-\frac{\Delta t}{\Delta x}(E^{+}_{i}-E^{+}_{i-1}+E^{-}_{i+1}-E^{-}_{i})+\Delta t\,H_i, \quad (6)$$

in which A and B are the derivatives of E and H with respect to Q, Δt and Δx are time and spatial steps, respectively, and I is the unity matrix of order 3. The superscript $+/-$ denotes positive/negative eigenvalues of matrices A and B. Moreover, a backward finite-difference approximation has been used for positive matrices, while a forward one is adopted for negative matrices. The system of linear equations (6) was solved for ΔQ, and then the matrix of the coefficients is updated using $Q^{n+1}=Q^{n}+\Delta Q$ for the next iteration. A steady state solution is said to have been obtained when the variation of properties (e.g., pressure) is very small between two iterations.

Initially the diffuser has been discretized uniformly. Once a converged solution is obtained, the regions with high gradients where more computational cells are needed were determined. A mathematical technique is then used to cluster the grid in high-gradient regions for accelerating the convergence speed (Hoffmann and Chiang, 1998). The convergency is controlled through the following relation:

$$\sum_{i=1}^{N}\left|P_i^{n+1}-P_i^{n}\right|<\varepsilon, \quad (7)$$

where ε is a very small number (e.g., $\varepsilon=10^{-3}$), P_i is pressure distribution over all nodes and superscripts $n+1$ and n denote new and old time steps, respectively. This procedure is repeated several times until a grid-independent solution is achieved.

3.2. Flexible wall model

A finite element approach was used to compute the nodal displacement of the diffuser wall. The wall was modelled by both beam elements and 2-D 4-noded axisymmetric elements. As the results were almost the same, the beam elements are used throughout this study as will be explained later.

The deformed shape of a beam element is described by the transverse displacement and slope of the beam. The beam is subjected to a distributed pressure created by flow passing in the diffuser. Using a standard finite element formulation, the strain energy for an arbitrary beam element can be written as (Rao, 1999)

$$A^{(e)}=\frac{EI}{2}\int_0^{\Delta x}\left(\frac{\mathrm{d}^2v}{\mathrm{d}x^2}\right)^2\mathrm{d}x, \quad (8)$$

where E, I and v are, respectively, the modulus of elasticity, moment of inertia of the beam cross-section and nodal values associated with a beam element. By assuming a Hermitian shape function, the stiffness matrix may be obtained by minimizing the strain energy of the element. Detailed descriptions on this formulation for obtaining the element stiffness and load vector can be found in Rao (1999). Once the element characteristics are found in a common global system, the next step is to construct the overall system equations. This yields

$$[K]\{U\}=\{F\}, \quad (9)$$

where $[K]$ and $\{F\}$ are the stiffness matrix and load vector, respectively. The displacement vector, $\{U\}$, has $3\times N$ components which correspond to longitudinal displacement, lateral displacement and slope at each node of the diffuser wall, where N denotes the number of nodes.

A computer program was written to calculate the nodal displacement of the beam subjected to distributed-pressure calculated from the flow field. In this study, we used the same mesh for both structural and fluid analyses. Because the computational domains share nodes on the common boundary, the compatibility conditions on the interface are naturally satisfied.

3.3. The fluid–solid coupling

The physical models used in treating fluid–structure interaction phenomena vary enormously in their complexity and range of applicability. To understand these phenomena, after modelling both the structure and the fluid, we need to use a process to couple these two parts of the problem in a proper fashion. In this way, two distinct methods for coupled field analysis can be identified; namely, the direct method and the sequential method. In the direct method, the analysis contains all the necessary degrees of freedom. Element matrices and load vectors contain all the necessary terms associated with both fields in this method. Sequential coupling, on the other hand, involves several analyses, each belonging to a different field. In this method, the two fields will be coupled by applying the results of the first analysis as a load vector for the second analysis. In this study, since the coupling is not very strong, we used the latter method which is simpler to apply and needs less computational effort than the former one.

The solution of Eq. (9) would require the structural left-hand side to be solved simultaneously with the solution of flow field on the right-hand side. However, Eq. (9) can be solved sequentially, i.e., solved for the flow field and then the load (pressure) is applied on the structure and the new shape of the wall is found. Using this new shape, the diffuser is discretized and solved again. This procedure will be repeated until the changes in the shape of the wall are smaller than a prescribed tolerance. To summarize, the steps that should be followed to get the final shape of the diffuser are: (i) start with an initial shape of the diffuser wall; (ii) solve the flow equations iteratively; (iii) solve the structural equations and compute the nodal displacements; (iv) update the wall profile; (v) repeat steps 2–4 until a converged solution is obtained.

4. Optimization algorithm

4.1. Genetic algorithm

Overall, one can categorize the optimization methods into two major classes, namely gradient-based methods and global-based methods. However, in many engineering applications conventional gradient-based algorithms are ineffective due to the problem of local minima or the difficulty in calculating gradients. The GA is one of the optimization methods that require no gradient and can achieve a global optimal solution (Holland, 1975; Davis, 1991). GAs are called so because they attempt to use the supposition of evolution as a basic mechanism for improvement, that is, learning/survival of the fittest, in solving a problem. The GAs are computationally simple but powerful and not limited by assumptions about the search space. Following the terminology of true genetic researches, the computational GAs developed by Holland (1975) and his students encode potential solutions into chromosome-like structures and then allow these structures to compete, reproduce and mutate to produce better solutions over time. GAs have been increasingly used in optimization studies over the past decade and have more recently been used in multi-disciplinary optimization.

Many facts control the way that a GA works. A potential solution has first to be encoded along with all of the other potential solutions that form a generation. This population is then fed one at a time to the objective function so that a measure of the performance of each number of the population can be ascertained. Those with better performance then have a higher probability of surviving the tournament selection process to reproduce the next generation. Mutation is allowed to occur and helps preserve genetic diversity. Over time, as generations build on the successes of previous generations, the performance of the entire population increases as the algorithm learns what elite values produce good answers. Poor performers die off and over many generations, the best performances split and recombine with each other to produce even better solutions.

Basic operators used to create successive improved populations include selection, crossover, mutation and interchange. Typically, two designs selected from a population are mated to create child designs. In order to ensure that good designs propagate to the child populations, a higher chance to be selected as parents is given to those designs that are better than the rest of the population. Selection is the part of the algorithm that provides better opportunity to good designs by implementing, for example, a roulette wheel which is divided into slices representing different designs. Those designs with better characteristics are given a proportionally larger slice of the wheel. When the wheel is spun (simulated by using a random number generator between 0 and 1, where the circumference of the wheel is normalized to be 1), those designs that occupy larger slices of the wheel have a better chance to be chosen as parent designs.

Once a pair of parents is selected, the mating of the pair also involves a random process called crossover. For example, by splicing together the left part of the string of one parent with the right part of the string of the other parent, two child strings are generated. The developed GA program can use different crossover relations. The most important ones that can be used in the program are: (a) arithmetic crossover; (b) heuristic crossover; (c) simple crossover.

Mutation is implemented by changing, at random and with small probability, the value of a gene and serves the purposes of avoiding premature loss of diversity in the designs. Since inferior designs may have some good traits that can get lost in the gene pool when these designs are not selected as parents, by introducing occasional mutations, different portions of the design space can be investigated for valuable information. Mutation is a random process where a single bit changes its value from 0 to 1 or 1 to 0. The applied mutation methods in the program are the most popular types: (i) multi-nonuniform mutation; (ii) nonuniform mutation; (iii) uniform mutation.

GAs are global optimizers because of mutation and their general probabilistic nongradient nature. However, in engineering applications, good answers are desired as fast as possible. Whether the answer is the absolute global optimum or not is less of a concern than whether a good answer can be found in the time allotted by management. Besides, for complicated problems, there is no analytical way to determine the global optimum and so arguments over whether the true optimum has been found are academic.

4.2. Optimization strategy

In the inverse problem one must usually determine unknowns in an indirect way. For example, our inverse problem is to find the diffuser geometry, given the flow speed and the prescribed pressure distribution on its surface. In order to carry out any optimization strategy, the following merits need to be carefully defined.

4.2.1. The objective function

The inverse design of diffuser consists of finding the geometric shape whose pressure distribution along the wall $p(x)$, matches a prescribed pressure distribution $\hat{p}(x)$, where x is the Cartesian coordinate measured along the diffuser axis. A discrete objective function, I, may be defined as (Dadone and Grossman, 2000)

$$I = \sum_{i=1}^{N} \left[p_i - \hat{p}_i\right]^2. \tag{10}$$

4.2.2. The design variables

As mentioned before, we wish to find the optimum geometric shape relative to global minimum of objective function. In this study, we used two different definitions (two cases) for the diffuser wall shape.

In the first case, we assumed that the diffuser shape follows the algebraic equation

$$s(x) = a + b\ \tanh(cx - d). \tag{11}$$

There are four unknowns in this relation, but one of them should be computed in terms of three others due to the constant rate of mass flow during the solution procedure. This will be necessary for obtaining a unique answer for the problem. The value of inlet area is fixed at 1.0512, thus using Eq. (11) one can arrives at

$$a = 1.0512 - b\ \tanh(cx - d), \tag{12}$$

therefore, three design variables, i.e., b, c and d, will be found by the optimization algorithm so that the objective function, I, has a minimum value.

In the second case, the diffuser shape profile will be created by a spline (de Boor, 1978)

$$Z(t) = (1-t)^3 Zm_1 + 3(1-t)^2 tZc_1 + 3(1-t)t^2 Zc_2 + t^3 Zm_2, \tag{13}$$

where t is a number between 0 and 1. $Z(t)$ is a complex number

$$Z(t) = x + \mathrm{i}S(x), \tag{14}$$

Zm_1 and Zm_2 are two fixed complex numbers that represent the coordinates of the two ends of the wall curve, and Zc_1, Zc_2 are control point coordinates in the complex plane. Moving the control points in the complex plane will produce different shapes for the diffuser wall. Therefore, the design variables in this case are the coordinates of the control points, i.e., x_{c1}, S_{c1}, x_{c2} and S_{c2}.

4.2.3. The design constraints

Some geometrical constraints are put on the design variables to avoid nonpractical shapes for the diffuser wall. In this way, no divergence will occur in the fluid–structure interaction algorithm.

5. Numerical implementation

Based on the above formulation, a code was developed to calculate the shape of a diffuser with flexible walls in order to have the same pressure distribution along its axis of symmetry as a diffuser with rigid walls. The developed code has four main parts: (i) a flow solver in quasi-one-dimensional form using a flux splitting method with shock capturing capability; (ii) a finite element program for the calculation of nodal displacement of the wall using standard beam theory; (iii) a sequential solid–fluid interaction algorithm that works iteratively to find the final shape of the diffuser; (iv) a robust adaptive real-coded GA algorithm for the design optimization.

5.1. Case 1—diffuser with rigid wall

As a preliminary validation case, a supersonic diffuser with rigid wall was considered. The aim is to demonstrate the ability of the flow solver to capture the shock correctly. The diffuser wall profile was calculated using Eq. (11) where the values of b, c and d are set to 0.347, 0.8 and 4, respectively. The value of a has been computed from Eq. (12). The supersonic inlet Mach number for this test case is set to 1.5 and the subsonic exit pressure is equal to 2.5 (normalized to

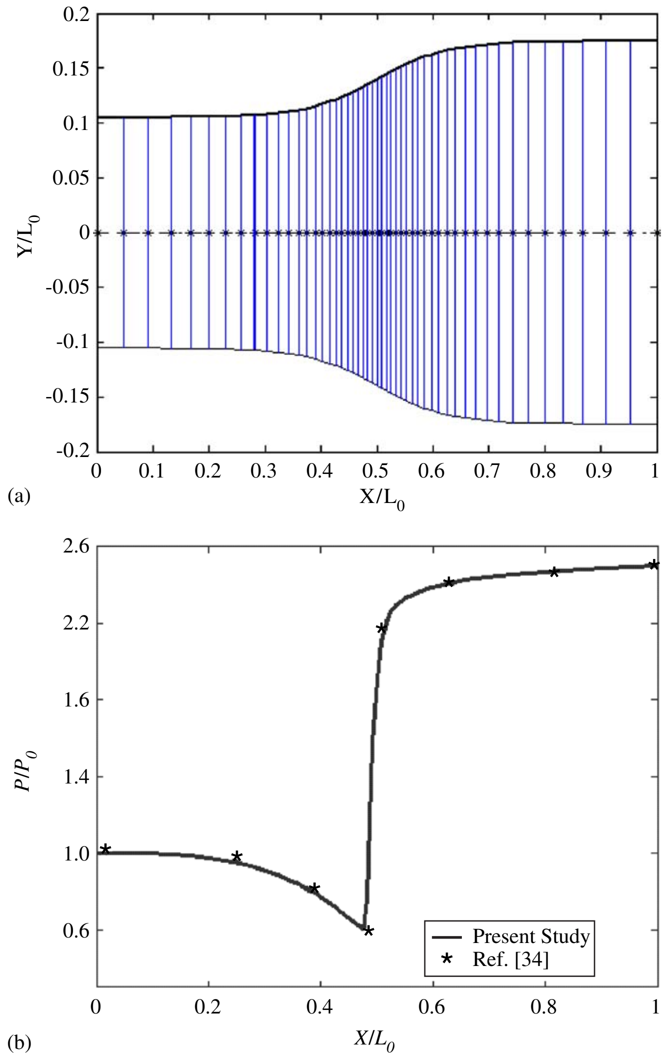


Fig. 2. The diffuser shape and the pressure distribution along the diffuser length: (a) diffuser shape, (b) pressure distribution.

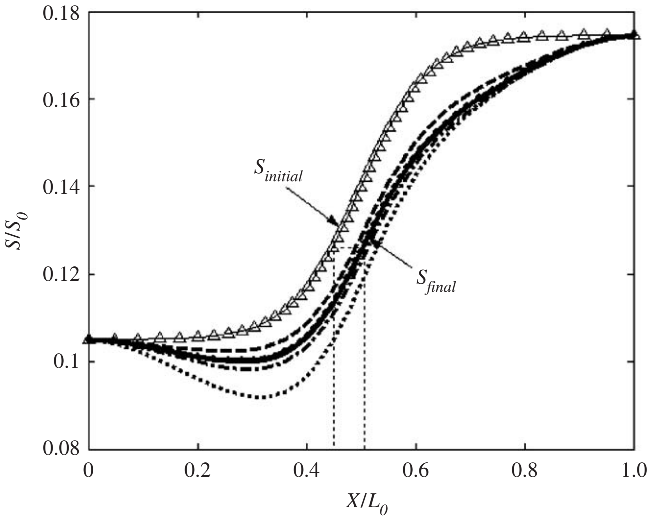


Fig. 3. Convergence towards the final shape of diffuser.

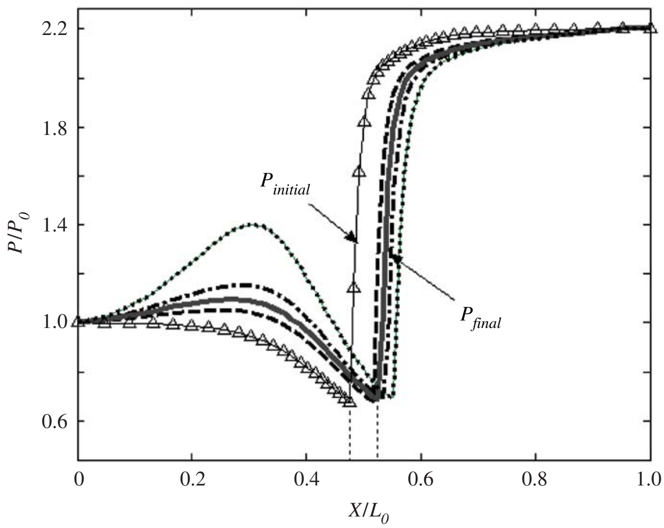


Fig. 4. Convergence towards the final pressure distribution.

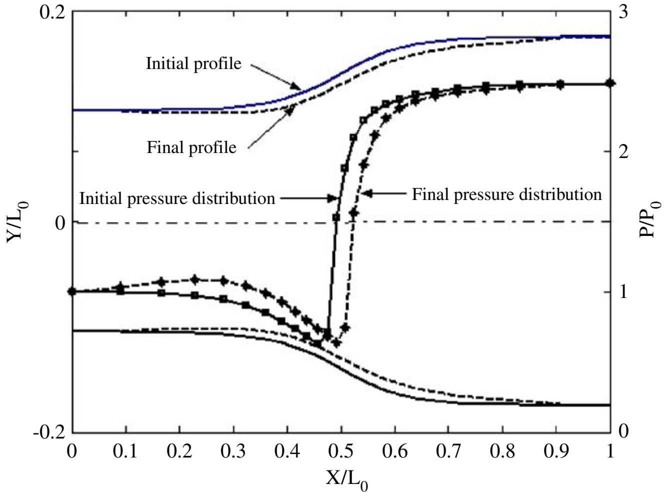


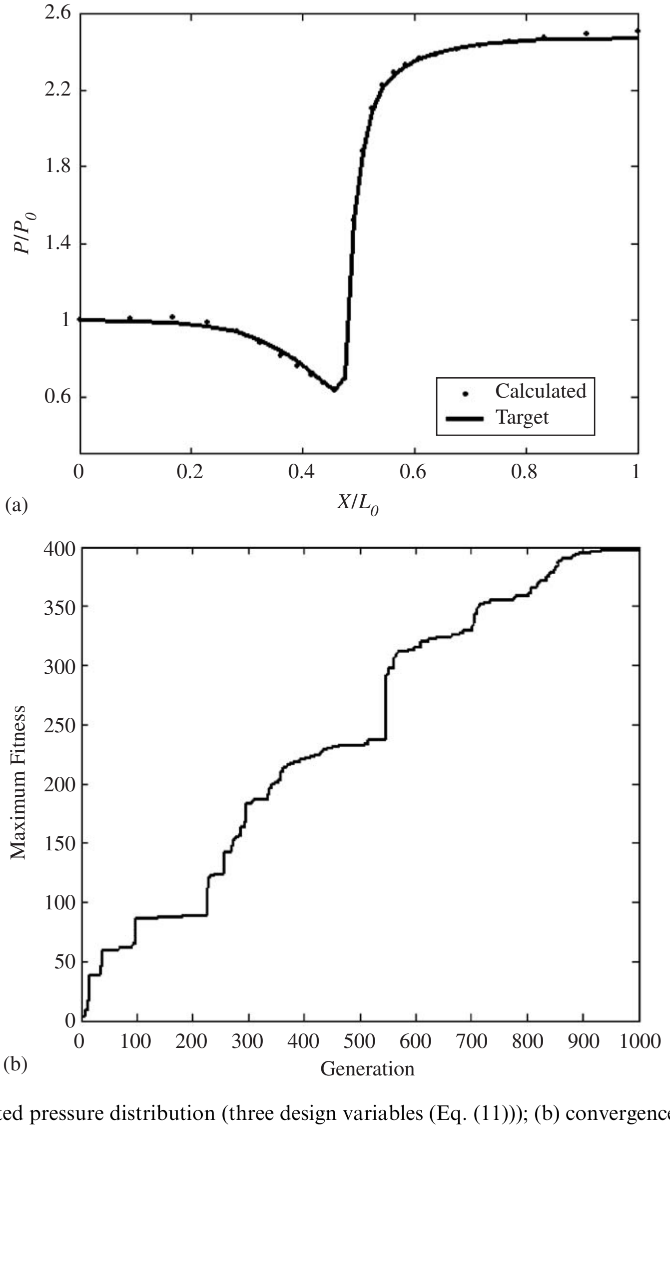
Fig. 5. The initial and final shape and corresponding pressure distribution for the diffuser with rigid and flexible wall (for flexible diffuser, $EI = 1 \times 10^5\,\mathrm{N\,m^2}$).

inlet pressure). The flow was calculated and the shock was captured successfully. Figs. 2(a) and (b) show the diffuser shape and the pressure distribution along its length. As it is clear from the figure, shock is established at $x/L_0 = 0.5$.

5.2. Case 2—diffuser with flexible wall

In the second case, the same diffuser but with flexible walls was considered. All geometrical and fluid properties as well as mesh density were the same as in the previous example. The stiffness of the wall will be controlled through the stiffness of the beam section (i.e., *EI*). A complete fluid–solid interaction should be considered to find out the final shape of the wall and the ultimate position of the shock inside the diffuser. Figs. 3 and 4 show the convergence toward the final solution for both shock position and pressure distribution along the diffuser.

Fig. 5 depicts the final results for the diffuser with rigid and flexible walls. The pressure distribution is also plotted on the same figure for better illustration. As the wall deflects towards the inner side (because the ambient pressure is higher than the inside pressure), the cross-sectional area of the diffuser reduces and, therefore, the shock moves forward towards the diffuser outlet.



(a)

(b)

Fig. 6. (a) Target and calculated pressure distribution (three design variables (Eq. (11))); (b) convergence history for GA (three design variables (Eq. (11))).

5.3. Case 3—inverse design problem (optimization)

The idea of inverse design is to calculate the proper shape of a diffuser with flexible walls, so that its pressure distribution becomes the same as a diffuser with rigid walls. Generally, the design of diffusers is based on the rigid wall assumptions. However, if the flexibility is high, a modification in the profile should be made in order to produce the same pressure distribution as a rigid wall diffuser. Thus, the idea here is to find the coefficients b, c and d in Eq. (11) for the diffuser with flexible wall to have the same pressure distribution as the one shown in Fig. 2(b). In other words, we are searching the variable space to find out the minimum of the quantity I in Eq. (10), in which $\{\hat{p}\}$ is the pressure distribution along the rigid wall diffuser. Any other pressure distribution may also be recommended with respect to the diffuser application by the designer.

The aforementioned case was solved using the GA. The GA was first tuned using heuristic crossover and multi-nonuniform mutation. The number of populations in each generation of GA was set to 100. The crossover rate was set to 90% and the mutation rate was 0.2%. The results for $EI = EI_0 = 1 \times 10^5$, are shown in Figs. 6(a) and (b). Fig. 6(a) shows the target and the calculated pressure distribution. It is clear that the final and the target are in good agreement. The convergence rate for the GA to reach the optimum of the objective function is depicted in Fig. 6(b). The value of the objective function becomes flat after 900 generations. Increasing the number of generations in the GA algorithm

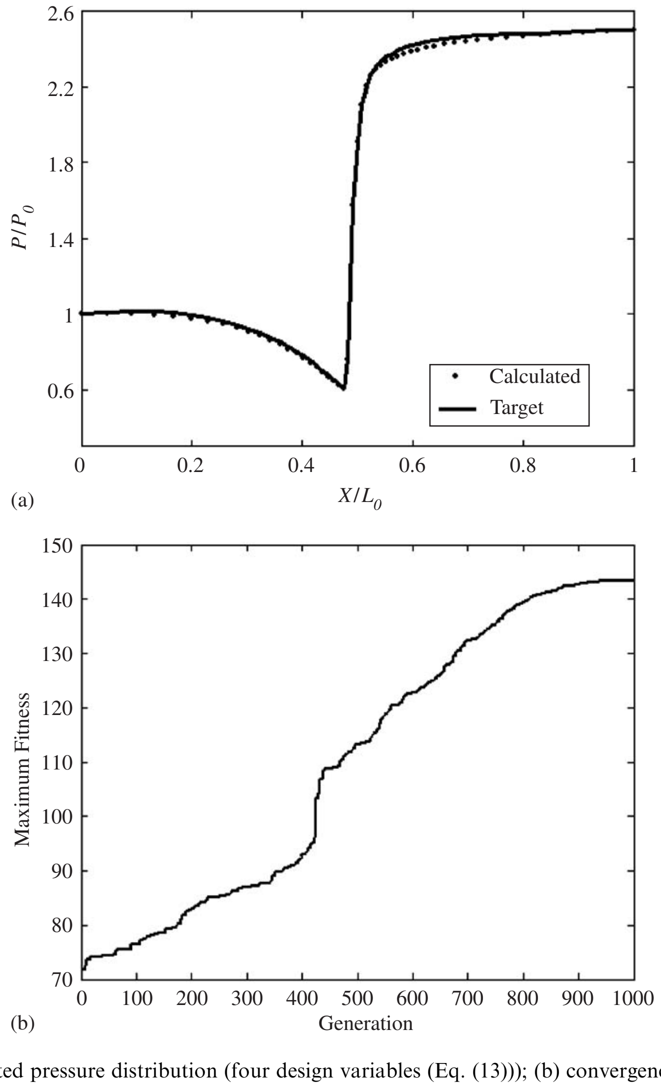


Fig. 7. (a) Target and calculated pressure distribution (four design variables (Eq. (13))); (b) convergence history for GA (four design variables (Eq. (13))).

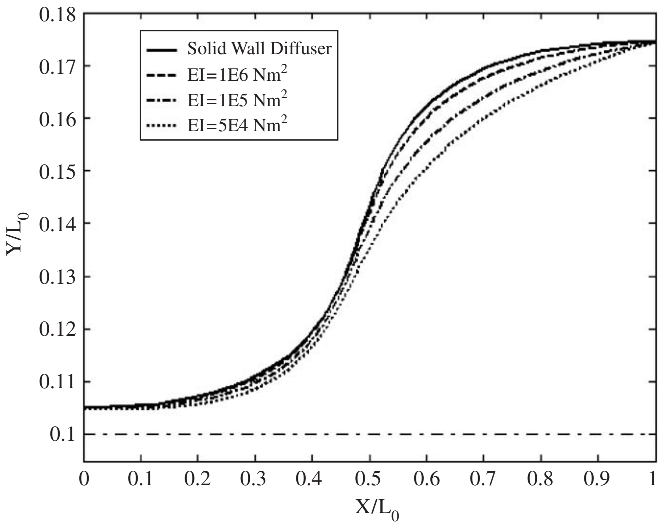


Fig. 8. Effect of wall flexibility on the diffuser shape (four design variables).

may further reduce this value. However, this needs more CPU time and computational effort with little gain in the accuracy of the pressure distribution. Note that for every GA population, a complete fluid–solid interaction should be carried out.

The same problem was solved again using splines, Eq. (13) instead of the algebraic formula in Eq. (11). Figs. 7(a) and (b) depict the final and target pressure distribution and the convergence history of GA. With four design variables for diffuser wall, the GA was executed for 1000 generations. The number of populations in each generation of GA was set to 100. The arithmetic crossover and nonuniform mutation options were used for this case. The crossover and mutation rate were set to 90% and 0.2%, respectively. Fig. 7(a) shows that the calculated and the target pressure distributions are coincident, indicating that the solution has sufficiently converged, particularly in the very sensitive shock transition region, where a derivative-based optimization process will have convergence difficulties with numerical sensitivity derivatives. The optimum values of the design variables found by GA are 7.19, 1.027, 2.206 and 1.602 for x_{c1}, S_{c1}, x_{c2} and S_{c2}, respectively.

5.4. Effect of wall flexibility on the final shape of the diffuser

Fig. 8 shows the final shape of the diffuser for different wall stiffnesses. Again the splines with four variables were used for creating the shape of the diffuser. Only the upper half of the diffuser is shown in this figure for better visualization. The problem was solved using heuristic crossover and multi-nonuniform mutation as GA operators. The population size was set to 60. All cases in this figure have the same pressure distribution along the diffuser axis of symmetry. However, due to the various stiffnesses of the wall, the diffuser outer shapes are different. By decreasing the stiffness of the wall (more flexible structure), the deviation from the original shape of the diffuser (diffuser with solid wall) was increased.

6. Conclusions

The genetic algorithm (GA) is well suited to designing complete aerodynamic modules. Unlike gradient-based optimization approaches, GAs find good designs by learning their own design lessons, without having to be given a starting solution or sensitivity derivatives. Even for large structural deformations, GA can easily handle continuous and discrete variables and work very well without divergence.

References

Baysal, O., Eleshaky, M.E., 1991. Aerodynamic sensitivity analysis methods for the compressible Euler equations. ASME Journal of Fluids Engineering 113, 681–688.

Bock, K., 1990. Aerodynamic design by optimization. Paper 20, AGARD CP-463.

Corana, A., Marchesi, M., Martini, C., Ridella, S., 1987. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. ACM Transaction Mathematical Software 3 (13), 262–280.

Dadone, A., Grossman, B., 2000. Progressive optimization of inverse fluid dynamic design problems. Journal of Computers & Fluids 29, 1–32.

Davis, L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold, Amsterdam.

de Boor, C., 1978. A Practical Guide to Splines. Springer, New York.

Doorly, D., Peiro, J., 2000. Supervised parallel genetic algorithm in aerodynamic optimization. AIAA paper 00-1245.

Doorly, D., Peiro, J., Spooner, S., 1999. Design optimization using distributed evolutionary methods. In; 37th Aerospace Science Meeting, AIAA Paper 99-0111.

Doorly, D., Peiro, J., Oesterla, J., 2001. Optimization of aerodynamic and coupled aerodynamic–structural design using parallel genetic algorithm. AIAA Paper 01-413.

Frank, P.D., Shubin, G.R., 1992. A comparison of optimization-based approaches for a model computation aerodynamics design problem. Journal of Computational Physics 98, 74–89.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publication, USA.

Hicks, R.M., Henne, P.A., 1977. Wing design by numerical optimization. AIAA Paper 77-1247.

Hoffmann, K.A., Chiang, S.T., 1998. Computational Fluid Dynamics for Engineers. Wiley, New York.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, USA.

Holst, T.L., Pulliam, T.H., 2001. Aerodynamic shape optimization using a real-number-encoded genetic algorithm. AIAA Paper 2473.

Hou, GJ.W., Taylor, A.C., Korivi, V.M., 1994. Discrete shape sensitivity equations for aerodynamic problems. International Journal for Numerical Methods in Engineering 37, 2251–2266.

Huffman, W.P., Melvin, R.G., Young, D.P., Johnson, F.T., Bussoletti, J.E., Bieterman, M.B., Hilmes, C.L., 1993. Practical design and optimization in computational fluid dynamics. In: 24th AIAA fluid Dynamics Conference, AIAA Paper 93-3111.

Ingber, L., 1993. Simulated annealing: practice versus theory. Mathematical and Computer Modelling 11, 29–57.

Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Lores, M.E., Smith, P.R., Hicks, R.M., 1979. Supercritical wing design using numerical optimization and comparisons with experiment. AIAA Paper 79-0065.

Narducci, R.P., Grossman, B., Haftka, R.T., 1995. Design sensitivity algorithms for an inverse design problem involving a shock wave. Inverse Problems in Engineering 2 (12), 49–83.

Obayashi, S., Tsukahara, T., 1997. Comparison of optimization algorithms for aerodynamic shape design. AIAA Journal 35, 1413–1415.

Obayashi, S., Yamaguchi, Y., Nakamura, T., 1997. Multiobjective genetic algorithm for multidisciplinary design of transonic wing platform. Journal of Aircraft 34, 690–693.

Oyama, A., 2000a. Multidisciplinary optimization of transonic wing design based on evolutionary algorithms coupled with CFD solver. In: European Congress of Computational Methods in applied Science and Engineering, ECCOMAS2000, Barcelona, Spain, September 11–14, pp. 1215–1223.

Oyama, A., 2000b. Wing design using evolutionary algorithms, PhD Thesis, Deptartment of Aeronautics and Space Engineering, Tohoku University, Japan.

Periaux, J., Chen, H.Q., Mantel, B., Sefrioui, M., Sui, H.T., 2001. Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. Finite Elements in Analysis and Design 37, 417–429.

Quagliarella, D., Della Cioppa, A., 1994. Genetic algorithm applied to the aerodynamic design of transonic airfoil. AIAA Paper 94-1896-CP.

Rao, S.S., 1999. The Finite Element Method in Engineering, third ed. Wiley, New York.

Sasaki, D., Obayashi, S., Sawada, K., Himeno, R., 2000. Multiobjective aerodynamic optimization of supersonic wings using Navier–Stokes equations. In: European Congress of Computational Methods in applied Science and Engineering, ECCOMAS2000, Barcelona, Spain, September 11–14, pp. 234–245.

Shubin, G.R., 1995. Application of alternative multidisciplinary optimization formulation to a model problem for static aeroelesticity. Journal of Computational Physics 118, 73–85.

Steger, J.L., Warming, R.F., 1979. Flux vector splitting of the inviscid gas dynamic equations with application to finite difference methods. NASA Report No. TM-78605.

Vanderplaats, G.N., 1984. Numerical Optimization Techniques for Engineering Design. McGraw-Hill Book Company, New York.

Vicini, A., Quagliarella, D., 1997. Inverse and direct airfoil design using a multiobjective genetic algorithm. AIAA Journal 35, 1499–1505.