Inverse design of 2-D subsonic ducts using flexible string algorithm

Article *in* Inverse Problems in Science and Engineering · December 2009

DOI: 10.1080/17415970903047451

4 authors, including:

Mahdi Nili-Ahmadabadi
Isfahan University of Technology
87 PUBLICATIONS **661** CITATIONS

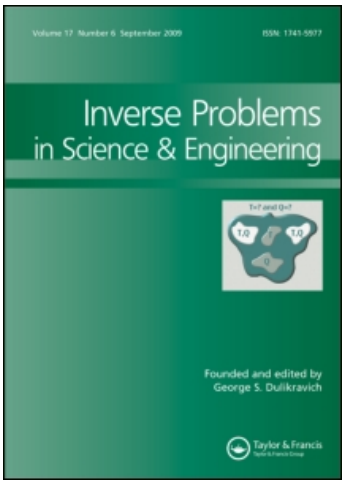
SEE PROFILE

Ali Hajilouy Benisi
Sharif University of Technology
53 PUBLICATIONS **596** CITATIONS

SEE PROFILE

Farhad Ghadak
Imam Hossein University
13 PUBLICATIONS **177** CITATIONS

SEE PROFILE



Inverse Problems in Science and Engineering
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713643452

Inverse design of 2-D subsonic ducts using flexible string algorithm

Mahdi Nili-Ahmadabadi [a]; Mohammad Durali [b]; Ali Hajilouy-Benisi [a]; Farhad Ghadak [c]
[a] School of Mechanical Engineering, Center of Excellence in Energy Conversion, Sharif University of Technology, Tehran, Iran [b] School of Mechanical Engineering, Center of Excellence in Design, Robotics and Automation, Sharif University of Technology, Tehran, Iran [c] Department of Aerospace Engineering, Imam Hossein University, Tehran, Iran

First Published:December2009

To cite this Article Nili-Ahmadabadi, Mahdi, Durali, Mohammad, Hajilouy-Benisi, Ali and Ghadak, Farhad(2009)'Inverse design of 2-D subsonic ducts using flexible string algorithm',Inverse Problems in Science and Engineering,17:8,1037 — 1057
To link to this Article: DOI: 10.1080/17415970903047451
URL: http://dx.doi.org/10.1080/17415970903047451

Inverse Problems in Science and Engineering
Vol. 17, No. 8, December 2009, 1037–1057

Taylor & Francis
Taylor & Francis Group

Inverse design of 2-D subsonic ducts using flexible string algorithm

Mahdi Nili-Ahmadabadi[a]*, Mohammad Durali[b], Ali Hajilouy-Benisi[a] and Farhad Ghadak[c]

[a]*School of Mechanical Engineering, Center of Excellence in Energy Conversion, Sharif University of Technology, P.O. Box 11155-9567, Tehran, Iran;* [b]*School of Mechanical Engineering, Center of Excellence in Design, Robotics and Automation, Sharif University of Technology, P.O. Box 11155-9567, Tehran, Iran;* [c]*Department of Aerospace Engineering, Imam Hossein University, Tehran, Iran*

(*Received 6 April 2008; final version received 17 May 2009*)

The duct inverse design in fluid flow problems usually involves finding the wall shape associated with a prescribed distribution of wall pressure or velocity. In this investigation, an iterative inverse design method for 2-D subsonic ducts is presented. In the proposed method, the duct walls shape is changed under a novel algorithm based on the deformation of a virtual flexible string in flow. The deformation of the string due to the local flow conditions resulting from changes in wall geometry is observed until the target shape satisfying the prescribed wall's pressure distribution is reached. The flow field at each step is analysed using Euler equations and the advection upstream splitting method method. Some validation test cases and a design example are presented here which show the robustness and flexibility of the method in handling complex geometries. The method is a physical and quick converging approach and can efficiently utilize commercial flow analysis software.

Keywords: inverse design; 2-D subsonic duct; flexible string algorithm

Nomenclature

A	linear acceleration ($\mathrm{m\,s^{-2}}$)
F	force vector (N)
N	number of shape modifications and number of links
p_i	pressure calculated at cell centre near the wall
Q	sum of the conserved flow variables such as $\rho, \rho u, \rho v, \rho e$ in Euler equations
t	time (s)
x	X position of joints (m), x coordinate
y	Y position of joints (m), y coordinate
w	width of duct (m)
Δ	difference
Δs	link length (m)

*Corresponding author. Email: nili@mech.sharif.edu

ISSN 1741–5977 print/ISSN 1741–5985 online

DOI: 10.1080/17415970903047451
http://www.informaworld.com

Δp_i^*	pressure difference applied to each link (Pa)
Δp_i	difference between TPD and CPD at each link
α	angular acceleration (rad s^{-2})
θ	link angle (deg)
ρ	mass per unit length (kg m^{-1})
ω	angular velocity (rad s^{-1})

Subscripts

i	links index
j	joints index
max	maximum
Tar	target
x	X component
y	Y component

Superscript

I.G.	initial guess
$j1$	starting point of each link
$j2$	end point of each link
n	iteration number at analysis code
$t+\Delta t$	related to updated geometry
t	related to current geometry

1. Introduction

The design of hardware involving fluid flow or heat transfer such as intakes, manifolds, duct reducers, compressor, turbine blades, etc., is defined as the shape determination of the solid elements so that the flow or heat transfer rate is optimal in some sense. Often, both the computational fluid dynamics (CFD) and design algorithms are involved in solving an optimal shape design problem. The limitations and computational cost of the design techniques are challenging problems for present time computational technology.

One of the optimal shape design methods is the surface shape design (SSD). The SSD in fluid flow problems usually involves finding a shape associated with a prescribed distribution of surface pressure or velocity. It should be noted that the solution of a SSD problem is not generally an optimum solution in a mathematical sense. It just means that the solution satisfies a target pressure distribution (TPD) which resembles a nearly optimum performance.

There are basically two different algorithms for solving SSD problems: decoupled (iterative) and coupled (direct or non-iterative) techniques. In the coupled solution approach, an alternative formulation of the problem is used in which the surface coordinates appear (explicitly or implicitly) as dependent variables. In other words, coupled methods tend to find the unknown part of the boundary and the flow-field unknowns simultaneously in a (theoretically) single-pass or one-shot approach.

On the other hand, the traditional fully coupled approaches transform the flow equations to a computational domain in which the unknown coordinates appear as dependent variables. Stanitz [1–3] solved two- and three-dimensional (3-D) potential flow duct design problems using stream and potential functions as independent variables.

Zanetti [4] considered two-dimensional and axi-symmetric Euler equations and mapped the physical domain to a fixed computational region.

A novel direct shape design method was proposed by Ashrafizadeh *et al*. [5]. They basically showed that a fully coupled formulation of the SSD problem could be solved in the physical domain using a simple extension of commonly used CFD algorithms. Since the proposed direct design method does not need any transformation to or from a computational domain, it is applicable, in principle, to any flow model in 2- or 3-D domains. Ghadak [6] extended the application of this method to the design of ducts carrying flows governed by the non-linear coupled Euler equations.

The iterative shape design approach relies on repeated shape modifications such that each iteration consists of flow solution followed by a geometry updating scheme. In other words, a series of sequential problems are solved in which the surface shape is altered between iterations so that the desired TPD is finally achieved.

Iterative methods, such as optimization techniques, have been by far the most widely used methods to solve the practical SSD problems. The traditional iterative methods used for the SSD problems are often based on trial and error or optimization algorithms. The trial and error process is very time-consuming and computationally expensive and hence needs designer experience to reach minimum costs. Optimization methods [7,8] are commonly used to automate the geometry modification in each iteration cycle. In such methods, an objective function (e.g. the difference between a current surface pressure and the target surface pressure [9]) is minimized and subjected to the flow constraints which have to be satisfied. Although the iterative methods are general and powerful, they are often computationally costly and mathematically complex. These methods can utilize the analysis methods for the flow-field solution as a black-box.

Other iterative methods presented so far use physical algorithms instead of mathematical algorithms to automate the geometry modification in each iteration cycle. These methods are easier and quicker than the other iterative methods. One of these physical algorithms is governed by a transpiration model, in which one can assume that the wall is porous and hence the mass can be fictitiously injected through the wall in such a way that the new wall satisfies the slip boundary condition. Aiming the removal of nonzero normal velocity on the boundary, a geometry update determined by applying either the transpiration model based on mass flux conservation [10–15] or the streamline model based on alignment with the streamlines [16] must be adopted.

An alternative algorithm is based on the residual-correction approach. In this method, the key problem is to relate the calculated differences between the actual pressure distribution on the current estimate of the geometry and the TPD (the residual) to required changes in the geometry. Obviously, the art in developing a residual-correction method is to find an optimum state between the computational effort (for determining the required geometry correction) and the number of iterations needed to obtain a converged solution. This geometry correction may be estimated by means of a simple correction rule, making use of the relations between geometry changes and pressure differences known from linearized flow theory.

The residual-correction decoupled solution methods try to utilize the analysis methods as a black-box. Barger and Brooks [17] presented a streamline curvature method in which they considered the possibility of relating a local change in surface curvature to a change in local velocity. Since then, a large number of methods have been developed following that concept. For example, in Garbedian-McFadden (GM) design method (elastic membrane concept) presented in [18], an auxiliary partial differential equation was solved iteratively

in the computational plane, together with the fluid flow equation, to achieve the given TPDs. In the GM method, the surface pressure distribution, or the flow velocity on the surface (i.e. just outside the boundary layer) depends on the surface ordinate, the surface slope and the surface second derivative. Linear combination of these three parameters is related to the change in velocity squared on the surface. Malone *et al.* [19] solved GM equation directly in a physical domain, rather than in the computational domain. A modified GM (MGM) design algorithm was presented by Malone *et al.* [20] that removed some limitations of the original GM method. Subsequent refinements and modifications made MGM applicable to design problems based on the full potential [21], the Euler [22] and the Navier–Stokes equations [23,24]. Takanashi [25] developed a 3-D wing design method based on this approach. In [26], Takanashi's method is linked with a Navier–Stokes solver to design transonic airfoils. Two major problems with the classical MGM approach were its slow convergence at leading and trailing edges of the airfoil, and its slower convergence in conjunction with the flow-field analysis codes of increasing non-linearity [27]. In order to alleviate these problems, Dulikravich [27] developed a mathematical formulation for MGM in shape inverse design using an analytical Fourier series solution. Also, he showed that 3-D subsonic wings can be reliably designed using a panel code, a Euler solver or a turbulent Navier–Stokes solver.

The main idea behind decoupling the flow and the geometry solutions in inverse design in most cases is to take maximum advantages of the available analysis methods. Another advantage of decoupled solution methods is the fact that, in general, the constraints can be implemented much more easily in a separate geometry updating process than in a complete system of equations for flow as well as geometry variables.

This research deals with an inverse method for designing ducts in 2-D configurations. The design goal is to obtain a duct so as to provide a given pressure distribution along its solid boundaries. The new feature of the method consists of considering the duct wall as a flexible string having mass. The difference between TPD and current pressure distribution (CPD) at each modification step is applied to the string as an actual force that accelerates and moves the string. Having achieved to the target shape, the difference between TPD and CPD vanishes and finally the string deformation is stopped automatically. Solving the string kinematic equations together with the flow equations at each modification step updates the duct shape so as to achieve the TPD.

Also, the research describes the method in detail and its effectiveness is verified through some numerical examples. Then, an analysis concerning the convergence rate and the stability of the method is provided.

In spite of the other residual correction methods with mathematical base that use flow equation for inverse design problems, the flexible string algorithm (FSA) turns the inverse design problem into a fluid–solid interaction scheme with physical sense that uses pressure to deform the flexible wall. Despite mathematical base methods that require arbitrary choice of parameters, in FSA method, the choice of string mass controlling solution stability (as will be explained later), can wisely be chosen to result in faster solution convergence. Therefore, higher calculation efficiency and time savings will be expected.

2. Fundamentals of the method

A 2-D flow field is assumed as shown in Figure 1. If a flexible string is fixed at point A in the flow, pressure applied to the sides of the string will deform it to lay on a

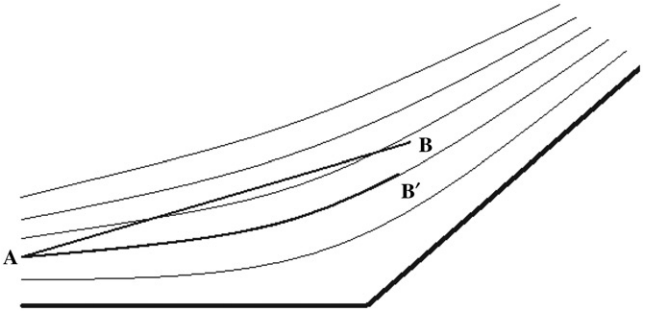


Figure 1. The string deformation in a 2-D flow.

streamline passed through the point A (A–B′ curve). This phenomenon occurs because it is assumed that no mass flux can pass across the string. For duct inverse design, the duct wall is considered as a flexible string whose outer surface is exposed to TPD and whose wetted surface is exposed to pressure resulting from passing flow through the duct.

Throughout the modification process to achieve the target wall geometry, the unknown contour duct wall, like flexible strings, is assumed to have fixed starting point and free end point. The wall geometry is modified by pressure difference between TPD and CPD. When target wall shape is obtained, this pressure difference logically vanishes. Similar to the other iterative methods, at each modification step, the flow field is analysed using Euler equations solution by means of advection upstream splitting method (AUSM) [28].

3. Mathematical approach

3.1. *Governing equations and boundary conditions for string*

To derive the string kinematic relations, the string is approximated by a chain with 'n' links of equal lengths with joints bearing no moment. Supposing uniform mass distribution along each link, the mass centre is located at its mid point. The free body diagram of an arbitrary link of the chain is shown at Figure 2. Assuming 2-D motion of the chain for each link, three kinematic relations can be derived as:

(1) Moment equation about mass centre of an arbitrary link:

$$\frac{1}{2}\left(F_{xi}^{j2}+F_{xi}^{j1}\right)\Delta s_i \sin\theta_i - \frac{1}{2}\left(F_{yi}^{j2}+F_{yi}^{j1}\right)\Delta s_i \cos\theta_i = \frac{1}{12}\rho_i(\Delta s_i)^3\alpha_i \tag{1}$$

(2) Newton's second law in x direction:

$$F_{xi}^{j1} - F_{xi}^{j2} - p_i w \Delta s_i \sin\theta_i = \rho_i \Delta s_i a_{xi} \tag{2}$$

(3) Newton's second law in y direction:

$$F_{yi}^{j1} - F_{yi}^{j2} + p_i w \Delta s_i \cos\theta_i = \rho_i \Delta s_i a_{yi} \tag{3}$$

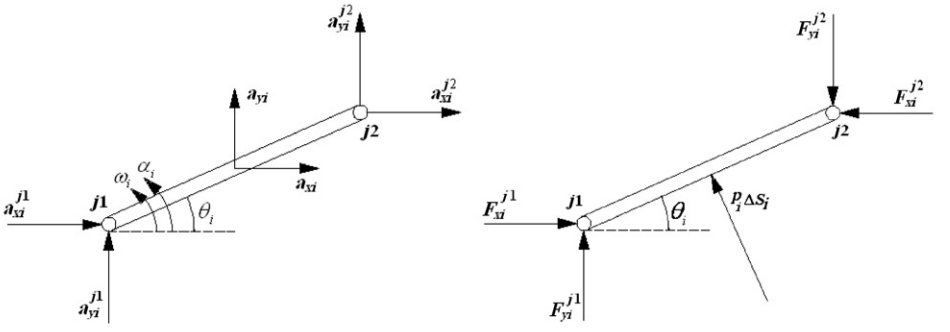


Figure 2. Free body diagram of an arbitrary link '*i*' of the chain, (a) kinematics and (b) forces.

Relations between the linear accelerations of the first joint of each link and its mass centre are as follows:

$$\rho_i \Delta s_i a_{xi} = \rho_i \Delta s_i \left(a_{xi}^{j1} - \frac{1}{2}\Delta s_i \sin\theta_i \alpha_i - \frac{1}{2}\Delta s_i \cos\theta_i \omega_i^2 \right) \tag{4}$$

$$\rho_i \Delta s_i a_{yi} = \rho_i \Delta s_i \left(a_{yi}^{j1} + \frac{1}{2}\Delta s_i \cos\theta_i \alpha_i - \frac{1}{2}\Delta s_i \sin\theta_i \omega_i^2 \right) \tag{5}$$

Equations (6) and (7) indicate the relations between the linear accelerations of two consecutive joints of each link:

$$a_{xi}^{j2} = a_{xi}^{j1} - \Delta s_i \sin\theta_i \alpha_i - \Delta s_i \cos\theta_i \omega_i^2 \tag{6}$$

$$a_{yi}^{j2} = a_{yi}^{j1} + \Delta s_i \cos\theta_i \alpha_i - \Delta s_i \sin\theta_i \omega_i^2 \tag{7}$$

Furthermore, the consistency equations at each joint are as follows:

$$a_{xi}^{j2} = a_{xi+1}^{j1}\ ,\ a_{yi}^{j2} = a_{yi+1}^{j1}\ ,\ F_{xi}^{j2} = -F_{xi+1}^{j1}\ ,\ F_{yi}^{j2} = -F_{yi+1}^{j1} \tag{8}$$

The boundary conditions of the chain's (string's) governing equations include fixed starting and free end points. Therefore, zero acceleration and force are attributed to the starting and end points, respectively.

$$a_{x1}^{j1} = a_{y1}^{j1} = 0 \tag{9}$$

$$F_{xn}^{j2} = F_{yn}^{j2} = 0 \tag{10}$$

To study the chain kinematics, it is enough to calculate angular acceleration of each link (α_i) and there is no need to calculate the other unknowns such as the forces and linear accelerations components. Having eliminated the forces and linear accelerations and solving the linear system of equations, the angular accelerations are calculated exactly. Then, the angular velocity (ω_i) and the angle change of each link ($\triangle\theta_i$) are obtained as follows:

$$\omega_i^{t+\Delta t} = \omega_i^t + \alpha_i \Delta t \tag{11}$$

$$\Delta\theta_i = 1/2\alpha_i \Delta t^2 + \omega_i \Delta t \tag{12}$$

Starting from the first link towards the end one, the new positions of joints $(j+1)$ are obtained by adding the angle change of each link with respect to the calculated position of the previous joint (j).

The solution starts with an initial guess such that the duct's main characteristics (e.g. length and inlet or outlet area) are known and fixed. In this method, one of the joints coordinates (say x) is fixed.

$$x_j^{t+\Delta t} = x_j^t \tag{13}$$

$$y_{j+1}^{t+\Delta t} = y_j^{t+\Delta t} + \Delta s_i \sin(\theta_i + \Delta\theta_i) \tag{14}$$

3.2. *Applying the pressure difference to the string*

In subsonic flow regimes, the boundary conditions are: duct inlet Mach number and outlet pressure. Therefore, at each shape modification step, the outlet pressure remains constant, while the inlet pressure changes according to the pressure in the first interior cell. As we require the string starting point to remain stationary, $\Delta p_{\text{first-joint}}$ must be zero. Thus, the pressure difference applied to the string at any other point must be gauged with respect to the pressure difference at the first joint. This is shown in Equation (15):

$$\Delta p_j = (p_{\text{Tar}(j)} - p_j) - [p_{\text{Tar}(1)} - p_1] \;::\; j = 1, n+1 \;::\; p_{n+1} = p_{\text{Tar}(n+1)} = P_{\text{back}} \tag{15}$$

The basis of the method requires that for internal subsonic flows, CPD is applied to the outer side of the string and TPD is applied to the inner side.

3.3. *FSA design procedure*

Figure 3 shows how the string equations are typically incorporated into existing flow solution procedures. The computed pressure surfaces are normally obtained from partially converged numerical solutions of the flow equations. During the iterative design procedure, as the CPD approaches the TPD, the force applied to the string gradually vanishes and at the final steps the subsequent solutions of the string equations yield no changes in the duct surface coordinates.

4. Validation cases

For validation of the proposed method, a given configuration such as a Michael nozzle, S-shaped nozzle or a straight duct with a bump is analysed to obtain the solid wall pressure distribution. Then, these pressure distributions are considered as our TPD for the SSD problems.

In all test cases studied here, computations were performed using a Pentium IV PC computer with 2.0 GHz CPU Core 2 Due and 2 GB RAM. The iterations were stopped after the residuals defined as: $\sum \left|\Delta p_j / \Delta p_j^{\text{I.G.}}\right|$ were reduced by three orders of magnitude. After each geometry modification, the analysis code is run until the residuals were reduced by 0.5 orders of magnitude. The residual of the analysis code is defined as the change in conserved flow variables ($\sum \left|(Q^{n+1} - Q^n)/Q^{\text{I.G.}}\right|$) and is normalized by the residual of the first iteration. The chosen residuals seem to be enough for convergence. The difference

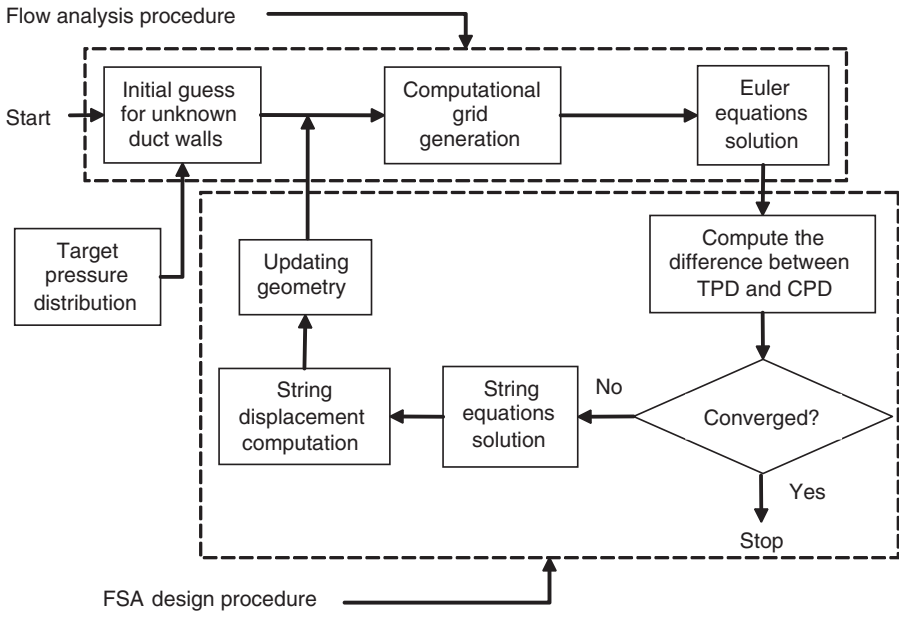


Figure 3. Implementation of the inverse design algorithm.

Table 1. Grid, number of iterations and CPU time for a test case.

		Number of iterations		CPU time (s)
Test case	Grid	Number of shape modifications	Max iterations in analysis code for each shape modification	Total
Michael nozzle	20×10	48	23	5.2
	30×15	62	56	23.4
	40×20	81	75	68.8
	60×30	120	112	514.3

between the final and the target shape are hardly noticeable with these numbers. Any further increase in residuals will add to computation time and bears no valuable benefit.

Here, the robustness of the design method is considered as the ability to use every initial guesses in each arbitrary computational grid. An ideal robust method should work well regardless the resolution of the computational grid and the initial guess. To show the robustness of the method, CPU time, the iteration number of inner loop (analysis code) and outer loop (design algorithm) for nozzle Michael test case, are shown in Table 1.

4.1. *A trivial test case*

To validate the results of the proposed design method, a 28×10 grid is used to design a straight duct with constant cross-sectional area with the inlet Mach number of 0.8.

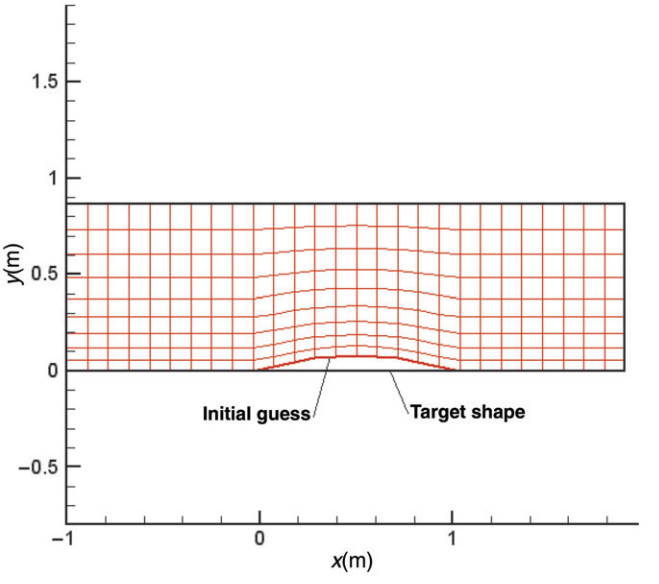


Figure 4. Geometry of bumped and straight duct.

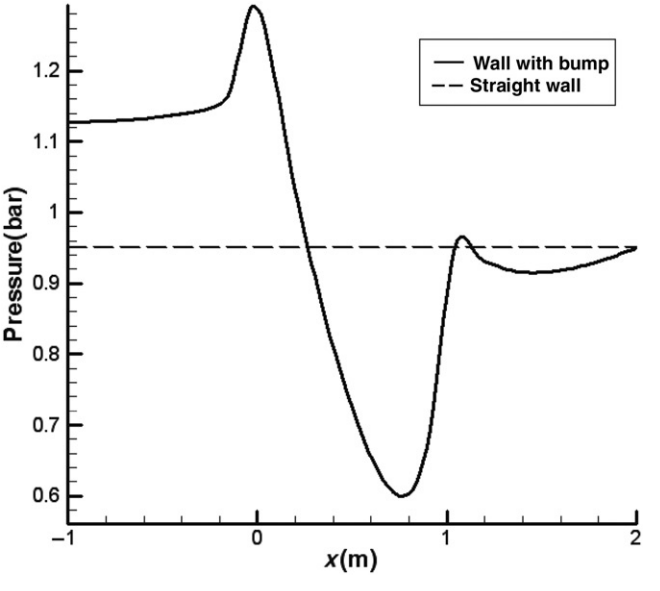


Figure 5. Pressure distribution along the bottom surface of bumped and straight duct.

In this example, both the TPD and its corresponding shape are exactly known in advance. In other words, everyone knows that for a straight duct the pressure distributions on the upper and lower walls are similar and constant.

Let us start with an intentionally distorted shape (Figure 4) as the initial guess and impose a constant pressure distribution as the TPD (Figure 5). The boundary pressure

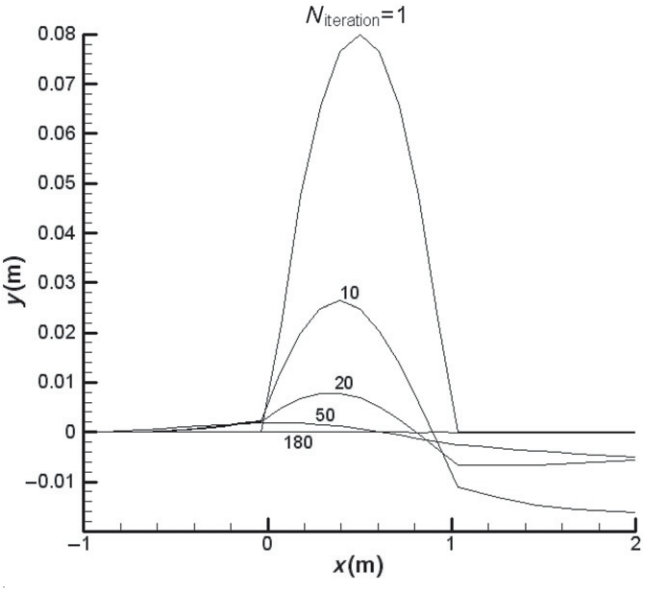


Figure 6. Shape modification process of the string from the bumped wall to the straight wall.

corresponding to the initial shape is shown in Figure 5. Although the deviation of the initial boundary pressure distribution from the TPD is relatively high, the method solves the design problem easily in 180 shape modifications and achieves the desired shape (Figure 6). At the initial iteration cycles, the string motion is large due to the large difference between CPD and TPD. By approaching the target shape, the difference between CPD and TPD is decreased and causes the slow motion of the string in the subsequent iterations.

The reverse of this problem (a rectangular channel as the initial guess and using the bumped wall pressure as the target) with the same pervious condition was also investigated, for which the number of iterations was found to be 90. Figure 7 shows the evolution of the string deformation which shows a gradual rising of the maximum point and falling of the end part of the string.

4.2. *Michael nozzle*

The second case is Michael nozzle which is extensively used in subsonic wind tunnels. To verify the capability of the proposed method, the flow field of the Michael nozzle is analysed and then the wall pressure distribution is determined. The method should converge to this shape from an initial arbitrary shape if the goal is set to be the calculated (desired) pressure distribution. Starting from a straight converging wall, the design program algorithm is converged only after 75 modification steps. The wall pressure distributions of the initial and final shape are shown in Figure 8(b). Figure 9 shows the modification evolution from initial guess to the target shape. The inlet Mach number has been set to 0.4 and a 30×15 grid has been used for this analysis.

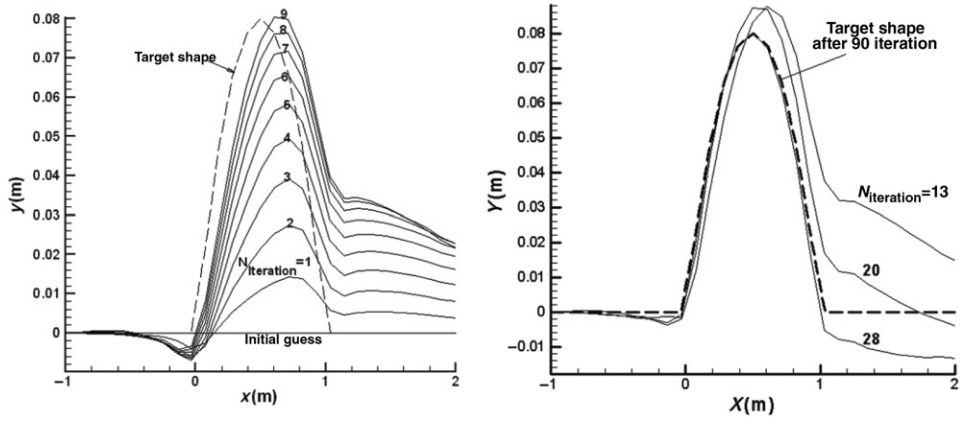


Figure 7. Shape modification process of the string from straight wall to the bumped wall.

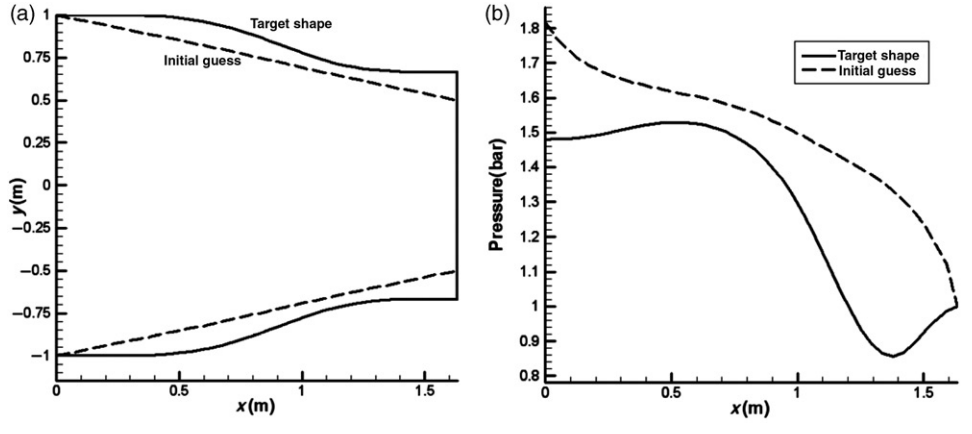


Figure 8. (a) Geometries of Michael nozzle and initial guess. (b) Initial and target wall pressure distributions.

4.3. *Effect of initial guess*

One of the outstanding capabilities of inverse design methods is its full independency from the initial guess. Keeping this reason in mind, a diffuser with straight walls can be considered as the initial guess for Michael nozzle. As shown in Figure 10, in spite of the presence of high difference between initial guess and target shape, the initial guess is converged to the target shape only after 130 iterations.

4.4. *Design of Michael nozzle with constant outlet area*

In the duct design problems, the duct inlet area is usually assumed to be constant and known but the outlet area is not, since the starting and end points of the string are fixed and free, respectively. However, there are some cases in which the duct outlet area or an

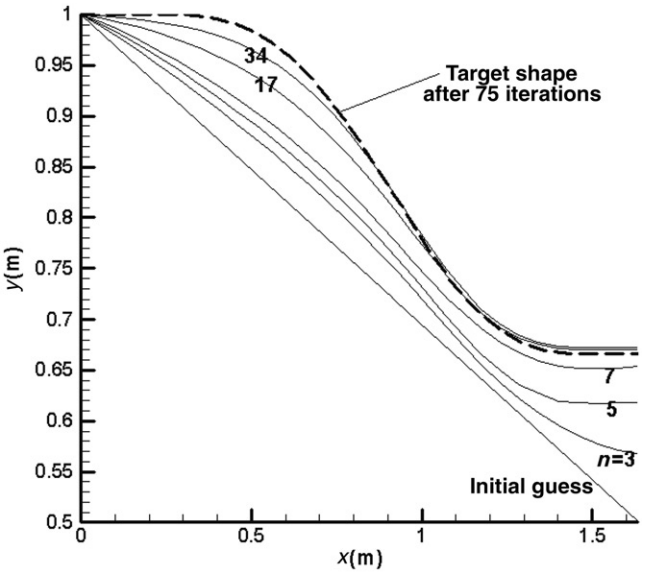


Figure 9. Shape modification process from a straight converging nozzle to the Michael nozzle.

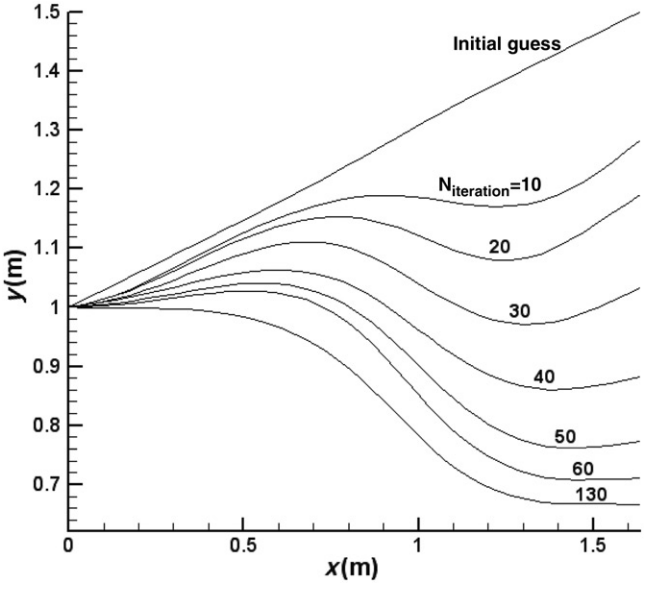


Figure 10. Shape modification process from the straight diffuser to the Michael nozzle.

intermediate sectional area is given as known. The method is also used for those problems. To test this capability, a straight duct with constant sectional area equal to the outlet area is considered as the initial guess. Because of the fixed outlet area or fixed string end point, in contrast to Equation (16), the pressure difference applying to the string is gauged with respect to the outlet pressure. As shown in Figure 11, the program is converged to the target shape after 150 geometry modifications.

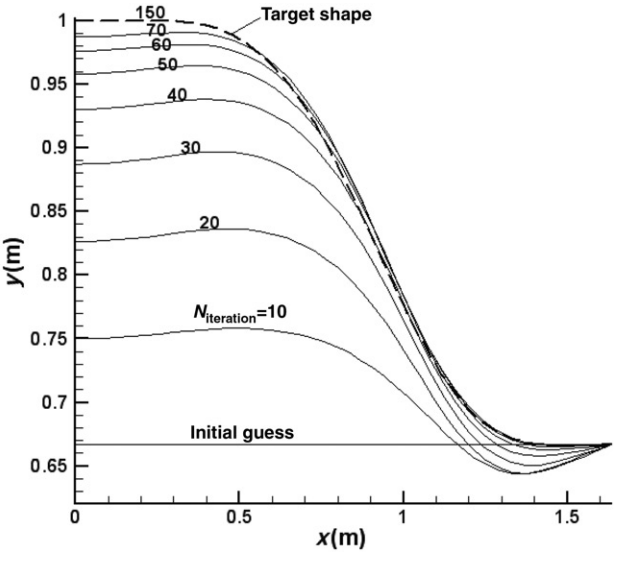


Figure 11. Shape modification process from the straight duct to the Michael nozzle.

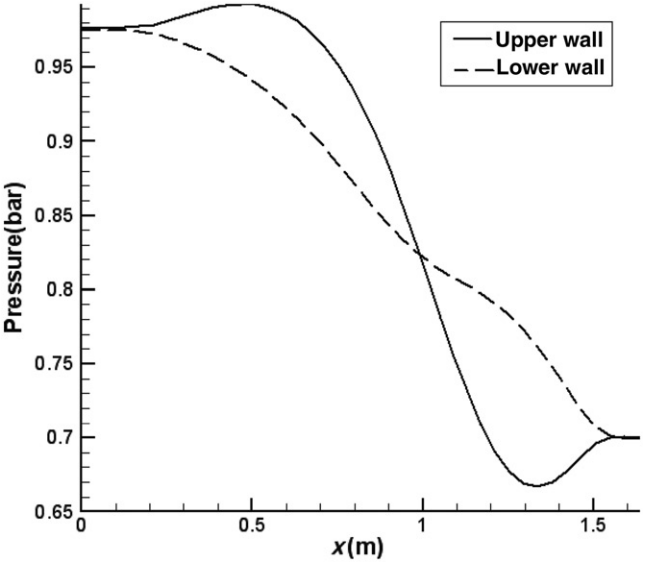


Figure 12. Pressure distribution along the lower and upper wall of the S-duct.

4.5. *S-shaped nozzle with two unknown walls*

Redesigned ducts at previous sections have only one unknown wall. S-shaped nozzle is a good example of ducts with two unknown walls. Figure 12 shows the pressure distributions along the walls of the S-shaped nozzle with the inlet Mach number of 0.4. A 40×7 grid has been used for the analysis code. As shown in Figure 13, a straight nozzle as the initial guess is deformed to the desired S-shaped duct after 3000 modification steps. The number of modifications for this case is considerably more than that for ducts with

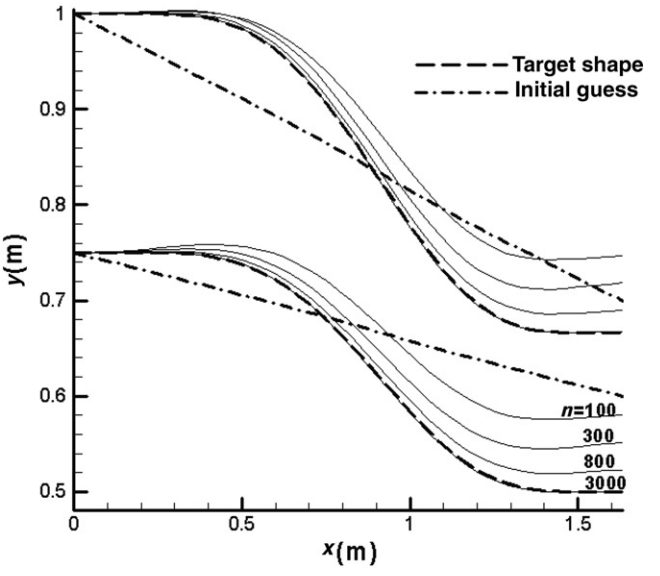


Figure 13. Shape modification process from a nozzle to the S-duct nozzle.

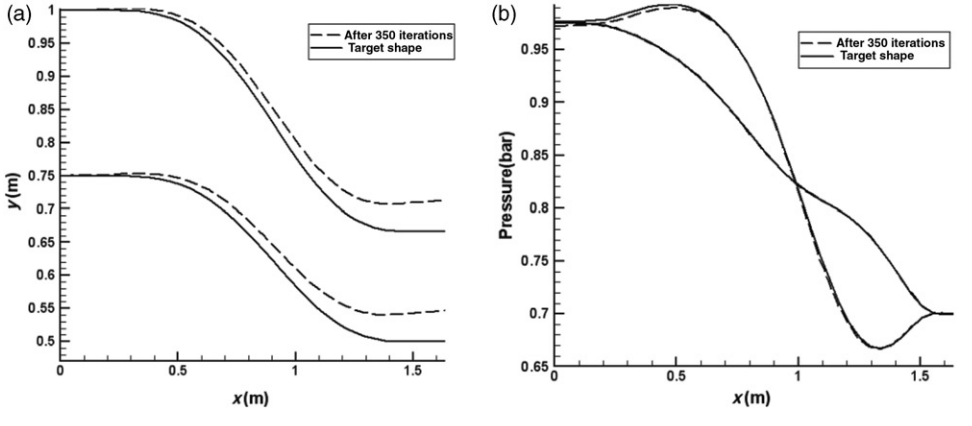


Figure 14. (a) S-duct geometry after 350 modifications with the target shape and (b) their corresponding wall pressure distributions.

one unknown wall because of the intense sensitivity of two-wall ducts to the wall pressure distribution. In this regard, the residuals must be reduced by five orders of magnitude to converge completely to the desired shape. As shown in Figure 14, though there exist a little difference between the target and current wall pressure distribution after 350 iterations, the difference between the target and current shape is considerable.

4.6. *Convergent–divergent nozzle*

The next validation test case is a convergent–divergent nozzle that includes both subsonic and supersonic regimes. Its TPD was obtained from numerical analysis of this nozzle with

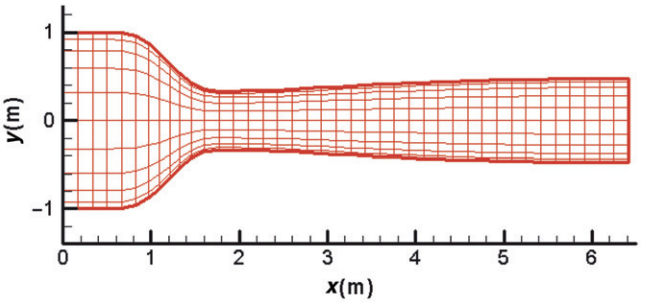


Figure 15. Convergent–divergent nozzle with its grid.

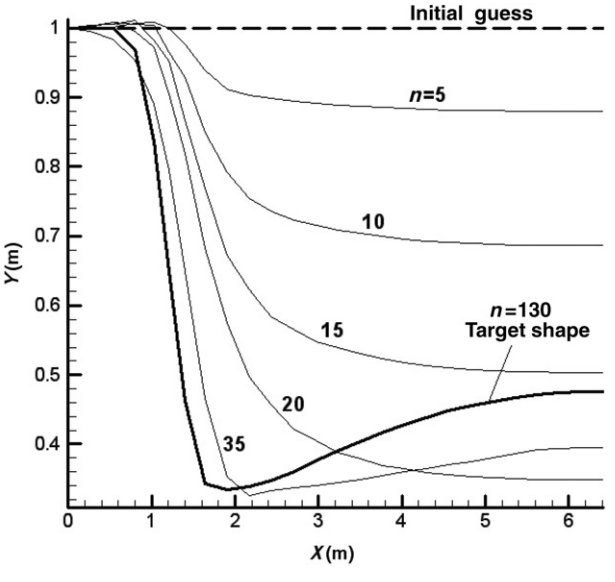


Figure 16. Wall shape modification from straight convergent duct to the supersonic nozzle with normal shock.

a 40×10 grid shown in Figure 15. The case specifications are as follows: inlet mach number $= 0.2$, inlet pressure $= 1$, back pressure $= 0.5$ and outlet mach number $= 1.5$. Starting from a constant area duct as initial shape, the inverse design algorithm converges to the convergent–divergent nozzle after 130 evolution steps (Figure 16) despite the large difference between initial and final shapes. Figure 17 shows the initial guess and TPD.

5. Performance of the method

5.1. *Convergence rate of the method*

The results indicate that the following parameters would affect the convergence rate of the method:

(1) The convergence rate of the fluid flow analysis code.
(2) The order of error reduction in analysis code after every shape modification.

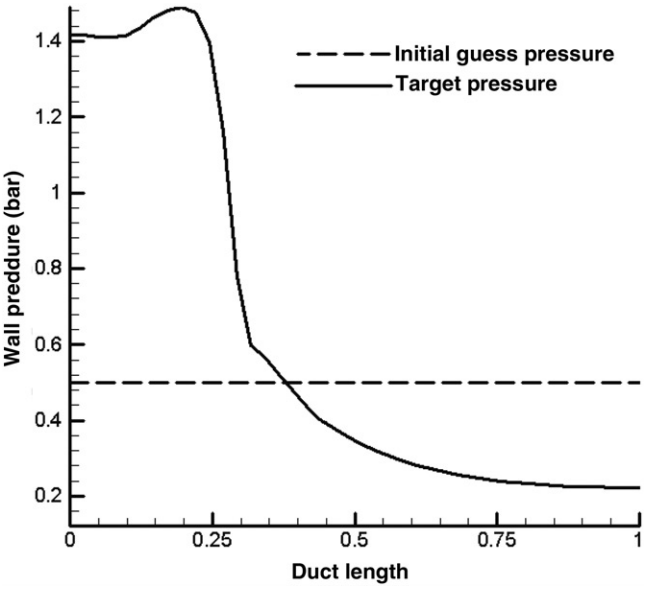


Figure 17. Initial guess and TPD for convergent–divergent nozzle.

(3) Mass per unit length of the string (density: ρ): scale analysis of the string kinematic relations shows that the pressure applying to the string (Δp_i) is proportional to the product of the string density, angular acceleration (α_i) and link length (Δs_i):

$$\Delta p_i \approx \rho \alpha_i \Delta s_i \tag{16}$$

Therefore, the lesser the density, the greater the string motion at each time step will be. Certainly, inordinate decrement of the string density will result in high oscillation and divergence.

(4) The time step for each shape modification process (Δt): scale analysis of the string kinematic relations shows that the string motion (Δy_i) is proportional to time step squared (Δt^2), angular acceleration and the link length:

$$\Delta y_i \approx \alpha_i \Delta s_i \Delta t^2 \tag{17}$$

Dividing Equation (16) by Equation (17) results in:

$$\Delta y_i \approx \Delta p_i \Big/ \left(\frac{\rho}{\Delta t^2}\right) \tag{18}$$

Equation (18) shows that if $\rho/\Delta t^2$ remains constant during the solution process, the design procedure approaches to the target shape and decrement of Δp_i causes the string to move slowly. If $\rho/\Delta t^2$ is decreased proportional to Δp_i, the string motion increases during intermediate steps and the convergence rate is improved. Further decrease of $\rho/\Delta t^2$ towards final steps may destabilize the string motion. In Table 2, an example showing the effect of ρ decrement at intermediate steps on the improvement of convergence rate is presented.

Table 2. Effect of ρ decrement on improvement of convergence rate.

Test case	Grid	Number of shape modifications	
		$\rho = \rho_0$	For $n<25$, $\rho = \rho_0(1 - n/50)$ For $n>25$, $\rho = \rho_0$
Michael nozzle	30×15	62	44

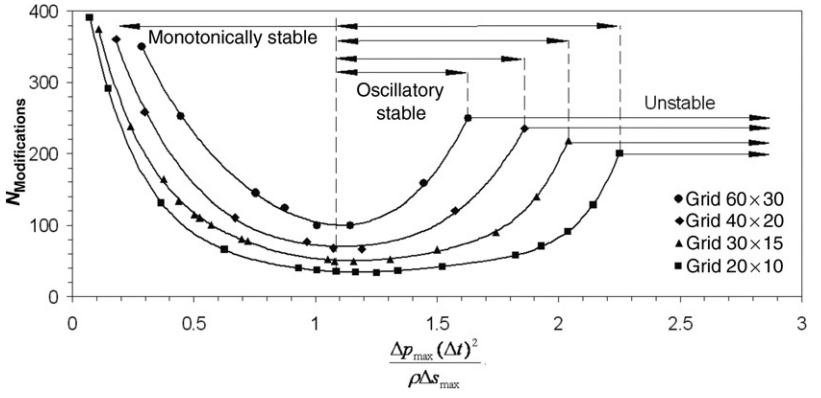


Figure 18. The number of shape modifications for the Michael nozzle at four different grids *vs.* $\Delta p_{\max}(\Delta t)^2/\rho\Delta s_{\max}$.

5.2. *Stability criteria*

If $\rho/\Delta t^2$ remains constant during the solution process, we can conclude:

$$\frac{\Delta y_i}{\Delta p_i} \approx \frac{\Delta y_{\max}}{\Delta p_{\max}} \approx \frac{\Delta t^2}{\rho} \Rightarrow \frac{\Delta y_{\max}}{\Delta s_{\max}} \approx \frac{\Delta p_{\max}(\Delta t)^2}{\rho \Delta s_{\max}} \tag{19}$$

$\Delta p_{\max}$ and $\Delta y_{\max}$ usually occur in the early steps of modification process. Assessment of the results indicates if $\Delta y_{\max}/\Delta p_{\max}$ or $\Delta p_{\max}(\Delta t)^2/\rho\Delta s_{\max}$ exceeds a certain limit, the string motion will be faced with high oscillatory and unstable behaviour. The limit of $\Delta p_{\max}(\Delta t)^2/\rho\Delta s_{\max}$ for each case is different but its order of magnitude is about 1. Figure 18 shows the number of shape modifications required for convergence of the Michael nozzle case at four different grids *versus* $\Delta p_{\max}(\Delta t)^2/\rho\Delta s_{\max}$. The figure has a minimum point which can be called 'critical point'. In the left-hand side of this point, the string motion is completely stable and monotonic, but in the right-hand side of this point, the behaviour is oscillatory. In the oscillatory region, after exceeding a certain limits, the method becomes unstable. Also, as seen in Figure 18, the coarser the grid size, the quicker the convergence rate and the wider the stability range.

5.3. *Filtering operation*

Author's experiences indicate that if the string wiggles grow, the program will diverge. To prevent the wiggles growth and instability, the following filtering operation is used in the inverse design code.

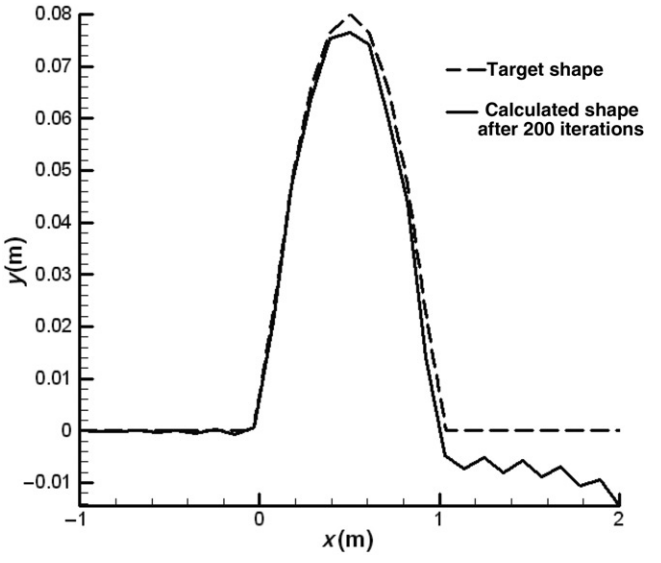


Figure 19. Calculated bumped wall after 200 iterations with no filtering operation.

Let us consider a numerical integration using three points $i-1$, i and $i+1$; if a quadratic polynomial is used for these three points, the following second-order accurate trapezoidal filter is obtained.

$$y_j^{t+\Delta t} = \frac{1}{4}\left(y_{j-1}^t + 2y_j^t + y_{j+1}^t\right) \tag{20}$$

Figure 19 shows the calculated shape for a bumped wall after 200 shape modifications with no filtering operation. As seen in this figure, the wiggles prevent the code from converging.

6. Design of an ideal nozzle

In a subsonic nozzle in which the flow velocity gradually increases in the streamline direction, a favourable pressure gradient exists along most parts of the walls. As shown in Figure 20, pressure increases in two small regions, in which a viscous flow might separate from the walls.

The goal of this example is to design an ideal nozzle in which there is no undershoot and overshoot in its wall pressure profile. For such a smooth wall pressure distribution (Figure 20, solid line), the flow remains subsonic and separation does not occur in real flows. Figure 21 shows the initial guess (a Michael nozzle with inlet area of 2, the inlet Mach number of 0.3 and a 28×20 grid) and the designed ideal nozzle in which a monotonic target pressure is satisfied. To create parallel flow at the inlet and outlet faces of ideal nozzle, the ideal pressure distribution must remain constant at the small part of inlet and outlet.

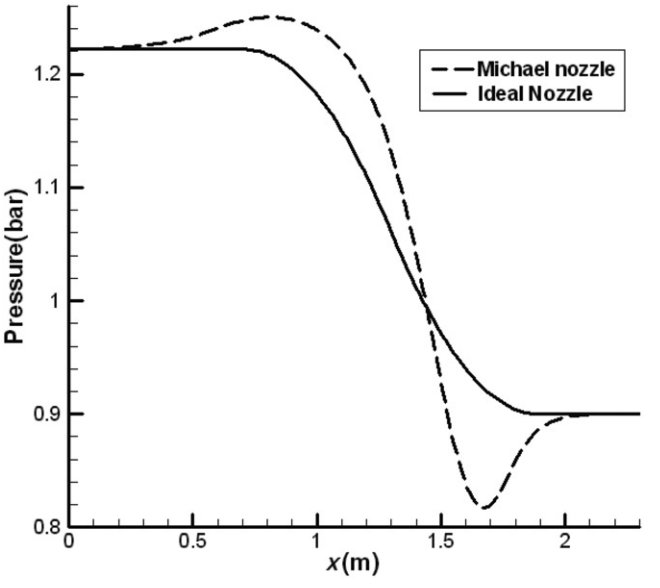


Figure 20. Wall pressure distribution of an ideal nozzle and Michael nozzle.

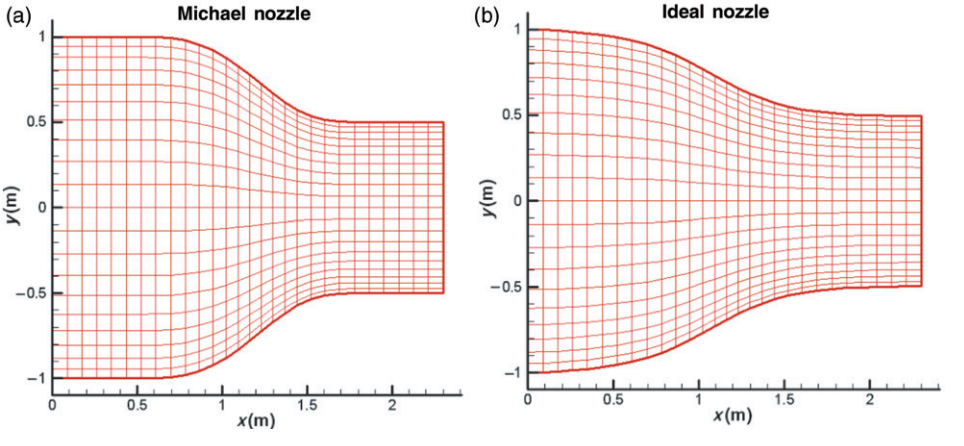


Figure 21. (a) Michael nozzle as the initial guess and (b) the designed ideal nozzle as the final shape.

7. Conclusions

The FSA design procedure is incorporated into an existing Euler code with AUSM method. FSA turns the inverse design problem into a fluid solid interaction scheme that is physical base. The method is quick converging and can efficiently utilize commercial flow analysis software as a black-box. The results show that the method can be very promising in duct and other flow conduit design. Some important results of this research are revealed as follows:

- Number of iterations required to converge the design procedure for S-shaped nozzle is considerably more than that for ducts with one unknown wall. This is

because of the intense sensitivity of two-wall ducts to the wall pressure distribution. For this reason, the residuals must be reduced five orders of magnitude to converge completely to the desired shape.

- If $\rho/\Delta t^2$ is decreased proportional to Δp_i, the string motion increases during intermediate steps and the convergence rate is improved.
- If $\Delta y_{\max}/\Delta s_{\max}$ or $\Delta p_{\max}(\Delta t)^2/\rho\Delta s_{\max}$ exceeds a limit, the string motion will become high oscillatory and unstable.
- In the completely stable region, the string approaches to the target shape monotonically.
- The coarser the grid size, the quicker the convergence rate and the wider the stability range.
- If the string wiggles grow, the divergence occurs.
- By eliminating the overshoot and undershoot in the Michael nozzle wall pressure profile, it is possible to design an ideal nozzle.

References

[1] J.D. Stanitz, *Design of two-dimensional channels with prescribed velocity distributions along the duct walls*, Tech. Rep. 1115, Lewis Flight Propulsion Laboratory, 1953.
[2] J.D. Stanitz, *General design method for three-dimensional, potential flow fields, I – Theory*, Tech. Rep. 3288, NASA, 1980.
[3] J.D. Stanitz, *A review of certain inverse methods for the design of ducts with 2- or 3-dimensional potential flows, in Proceedings of the Second International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-II)*, G.S. Dulikravich, ed., Pennsylvania State University, University Park, PA, 1987, October 26–28.
[4] L. Zanetti, *A natural formulation for the solution of two–dimensional or axis–symmetric inverse problems*, Int. J. Numer. Methods Eng. 22 (1986), pp. 451–463.
[5] A. Ashrafizadeh, G.D. Raithby, and G.D. Stubley, *Direct design of ducts*, J. Fluids Eng. Trans. ASME 125 (2003), pp. 158–165.
[6] F. Ghadak, *A direct design method based on the Laplace and Euler equations with application to internal subsonic and supersonic flows*, PhD Thesis, Sharif University of Technology, Aero Space Department, Iran, 2005.
[7] C.-H. Cheng and C.-Y. Wu, *An approach combining body fitted grid generation and conjugate gradient methods for shape design in heat conduction problems*, Num. Heat Tr. B 37(1) (2000), pp. 69–83.
[8] A. Jameson, *Optimal design via boundary control*, Optimal Design Methods for Aeronautics, AGARD, 1994, pp. 3.1–3.33.
[9] J.S. Kim and W.G. Park, *Optimized inverse design method for pump impeller*, Mech. Res. Commun. 27(4) (2000), pp. 465–473.
[10] V. Dedoussis, P. Chaviaropoulos, and K.D. Papailiou, *Rotational compressible inverse design method for two-dimensional, internal flow configurations*, AIAA J. 31(3) (1993), pp. 551–558.
[11] A. Demeulenaere and R. van den Braembussche, *Three-dimensional inverse method for turbomachinery blading design*, ASME J. Turbomachinery 120(2) (1998), pp. 247–255.
[12] L. De Vito and R. van den Braembussche, *A novel two-dimensional viscous inverse design method for turbomachinery blading*, Trans. ASME 125 (2003), pp. 310–316.
[13] R. van den Braembuussche and A. Demeulenaere, *Three-dimensional inverse method for turbomachinery blading design*, J. Turbomachinery 120 (1998), pp. 247–255.

[14] O. Leonard, and R. van den Braembussche, (1997), *A two-dimensional Navier–Stokes inverse solver for compressor and turbine blade design*, Proc. IMECH E Part A J. Power Energy, 211 pp. 299–307.
[15] P.A. Henne, *An inverse transonic wing design method*, AIAA Paper No. 80–0330, 1980.
[16] G. Volpe, *Inverse design of airfoil contours: Constraints, numerical method applications*, AGARD 1989, Paper 4, 1989.
[17] R.L. Barger and C.W. Brooks, A streamline curvature method for design of supercritical and subcritical airfoils, NASA TN D-7770, 1974.
[18] P. Garabedian and G. McFadden, *Design of supercritical swept wings*, AIAA J 30(3) (1982), pp. 444–446.
[19] J. Malone, J. Vadyak, and L.N. Sankar, *Inverse aerodynamic design method for aircraft components*, J. Aircraft 24(1) (1986), pp. 8–9.
[20] J. Malone, J. Vadyak and L.N. Sankar, *A technique for the inverse aerodynamic design of nacelles and wing configurations*, AIAA Paper 85–4096, 1985.
[21] R.L. Campbell and L.A. Smith, A hybrid algorithm for transonic airfoil and wing design, AIAA Paper 87–2552, 1987.
[22] R.A. Bell, and R.D. Cedar, *An inverse method for the aerodynamic design of three-dimensional aircraft engine nacelles*, in *Proceedings of the Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences*, *ICIDES-III*, G.S. Dulikravich, ed, Washington, D.C., 23–25 October 1991, pp. 405–417.
[23] J.B. Malone, J.C. Narramore and L.N. Sankar, An efficient airfoil design method using the navier–stokes equations, AGARD 1989, Paper 5, 1989.
[24] J.B. Malone, J.C. Narramore, and L.N. Sankar, *Airfoil design method using the Navier–Stokes equations*, J. Aircraft 28(3) (1991), pp. 216–224.
[25] S. Takanashi, *Iterative three-dimensional transonic wing design using integral equations*, J. Aircraft 22 (1985), pp. 655–660.
[26] N. Hirose, S. Takanashi, and N. Kawai, *Transonic airfoil design procedure utilizing a Navier–Stokes analysis code*, AIAA J. 25(3) (1987), pp. 353–359.
[27] G.S. Dulikravich and D.P. Baker, *Aerodynamic shape inverse design using a Fourier series method*, AIAA Paper No. 99–0185, 1999.
[28] M.-S. Liou, Ten *years in the making-AUSM-family*, AIAA Paper 2001–2521, June 2001.