Transaction B: Mechanical Engineering
Vol. 16, No. 2, pp. 111–120
© Sharif University of Technology, April 2009

SCIENTIA
IRANICA

Direct Design of Branched Ducts

F. Ghadak[1], M. Taiebi-Rahni[1,*] and A. Ashrafizadeh[2]

Abstract. *A fully coupled formulation of thermo-fluid shape design problems has recently been developed in which the unknown nodal coordinates appear explicitly in the formulation of the problem. This "direct design" approach is, in principle, generally applicable and has been successfully applied in the context of potential and Euler flow models.This paper focuses on the direct design of ducts using the ideal flow model and may be considered as an addendum to the paper entitled "Direct Design of Ducts" [1]. However, a cell-vertex finite volume method is used and a different boundary condition implementation technique is applied, as compared to the method presented in the previous paper. The other new feature is that a non-linear algebraic method is used for grid generation. The method is also proved to be capable of designing complex flow passages, such as branched ducts.*

Keywords: *Fully-coupled inverse method; Direct design; Internal flow; Stagnation point.*

INTRODUCTION

Surface Shape Design (SSD) in fluid flow problems usually involves finding a shape associated with a prescribed distribution of surface pressure or velocity. SSD methods may be broadly categorized as iterative or direct (fully coupled) methods. In iterative shape design methods, an iteration consists of a flow solution, using available geometry, followed by a geometry modification. To automate the geometry modification, SSD problems can be solved by optimization methods [2,3]. In optimization methods, an objective function (e.g., the difference between a current surface pressure and the target surface pressure) is minimized, subjected to the constraints that the flow equations are satisfied. Even though the iterative methods are general and powerful, they are computationally expensive.

The traditional fully coupled approaches, on the other hand, transform the flow equations to computational domains, in which the unknown coordinates appear as dependent variables. Stanitz [4] solved two- and three-dimensional potential flow duct design problems, using stream and potential functions as independent variables.

The work of Stanitz has been extended to two-dimensional rotational flows by Dedoussis et al. [5]. Chaviaropoulos et al. [6,7] solved the three-dimensional potential duct design problem using another approach. Their methods [5-7] were based on rewriting the governing equations in a natural system of coordinates. In traditional fully coupled approaches, the desired geometry is obtained by the solution of a set of partial differential equations. For this reason, these approaches are called single-pass methods.

The traditional fully coupled shape design methods, based on coordinate transformations, are not generally applicable and are mathematically complex. However, their computational cost is comparable to the corresponding analysis problem.

Recently, a fully coupled SSD method has been proposed by Ashrafizadeh et al. [1,8,9]. The method is called a direct design approach, because the unknown nodal coordinates appear as dependent variables in the formulation of the problem in the physical domain (no transformation to or from a computational space is needed). The method is a simple extension of existing CFD algorithms, which were successfully applied in the context of potential and Euler flow models [1,8-10]. An element-based finite volume method and a linear algebraic grid were used in [1,8,9].

In the present paper, a cell-vertex finite volume method and a non-linear algebraic grid are used to

[1] *Department of Aerospace Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-8639, Iran.*
[2] *Department of Mechanical Engineering, Khajeh Nasire Toosi University of Technology, Tehran, P.O. Box 16765-3381, Iran.*
[*] *Corresponding author. E-mail: taeibi@Sharif.edu*

Received 26 September 2006; received in revised form 8 December 2007; accepted 2 June 2008

discretize the governing equation and the solution domain. Particular attention has been paid to the design of branched ducts.

The stagnation point has always been a challenge in inverse design problems. It occurs in airfoil and branched duct design processes. In the vicinity of a stagnation point, the coefficient matrix becomes singular or nearly singular and the problem is ill-conditioned. One way to achieve well-posedness is to introduce free parameters in the prescribed pressure distribution. These parameters are determined as part of the solution, so that the constraints on the pressure distribution are automatically satisfied; the specific choice for the adjustable free parameters determines, implicitly, the class of admissible solutions. Volpe and Melnik [11] considered freestream speed as a free parameter while maintaining a specified location of the forward stagnation point. Drela [12] chose to fix the freestream speed, but left the location of the forward stagnation point in physical space unspecified. The Stanitz method [4] needs special treatment at the stagnation points. Ashrafizadeh et al. [9] use geometry patches to resolve this issue. A least square method is proposed in this paper to overcome the difficulties associated with the stagnation point.

OVERVIEW

Consider a flow analysis problem in a duct as shown in Figure 1. To numerically solve the analysis problem, a computational grid is needed, which can be easily generated if the boundaries of the duct are described by an adequate number of nodes. Flow unknowns at nodal points can then be calculated if appropriate boundary conditions are employed.

Now, consider the corresponding shape design problem, in which, in addition to the flow unknowns, the boundary nodes are also missing. Without the boundary nodes, the computational grid cannot be generated and in the absence of grid points, a solution for the flow unknowns is impossible. Therefore, in SSD problems, constraint equations at the boundary nodal points are needed to close the system of equations. The specified target pressure at an unknown boundary point provides one of the required constraints. In a tow-dimensional SSD problem, an additional constraint on the location of a boundary point is still needed. One way of obtaining the required constraint is to force a boundary node to move along a specified direction called the spine [8]. By fixing spine directions, only one constraint, spine coordinate R, is needed to specify the location of a grid point in a two-dimensional domain. The spine coordinates of a node on the lower boundary (R_{Li}) and a node on the upper boundary (R_{Ui}) are shown in Figure 1. Simple interpolation formulas can be used to relate internal node coordinates to the corresponding boundary spine coordinates (R_{Li}, R_{Ui}). Therefore, the computational grid in an SSD problem can be fully described by spine coordinates of boundary nodes. The relations between the Cartesian and the spine coordinates are as follows:

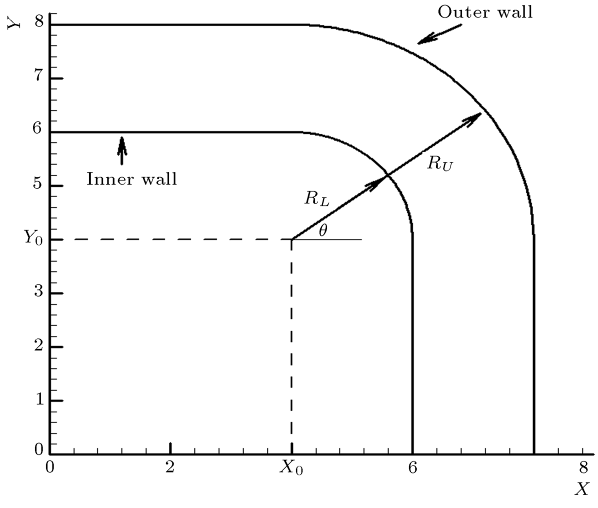

Figure 1. The spine used in the present work.

$$R_L = (X_L - X_0)/\cos\theta = (Y_L - Y_0)/\sin\theta, \tag{1a}$$

$$R_U = (X_U - X_0)/\cos\theta = (Y_U - Y_0)/\sin\theta, \tag{1b}$$

where X_0 and Y_0 are the coordinates of the spine's origin (Figure 1). Due to the non-linearity of the equations in the direct design approach, an iterative solution is unavoidable. Therefore, initial guesses are used to start the iterations.

DISCRETIZATION AND LINEARIZATION OF THE GOVERNING EQUATION

The direct design method used in this work is based on ideal flow. Using the Bernoulli equation, the given wall pressure is converted to velocity. The governing equation for the incompressible, steady, two-dimensional potential flow [13] applied to the $ABCD$ cell (Figure 2) is the Laplace equation as follows:

$$\int\limits_{ABCD} \left(\frac{\partial^2\Psi}{\partial x^2} + \frac{\partial^2\Psi}{\partial y^2}\right) dxdy = \iint (\nabla\Psi.\mathbf{n}')dA = 0, \tag{2}$$

$$(\nabla\Psi.\mathbf{n}')dA = \frac{\partial\Psi}{\partial x}dy - \frac{\partial\Psi}{\partial y}dx = \frac{\partial\Psi}{\partial\mathbf{n}'}ds, \tag{3}$$

where Ψ is the stream function, $\mathbf{n}'$ is the unit vector perpendicular to the surface, and s is the distance along the cell boundary. Using the finite volume method,

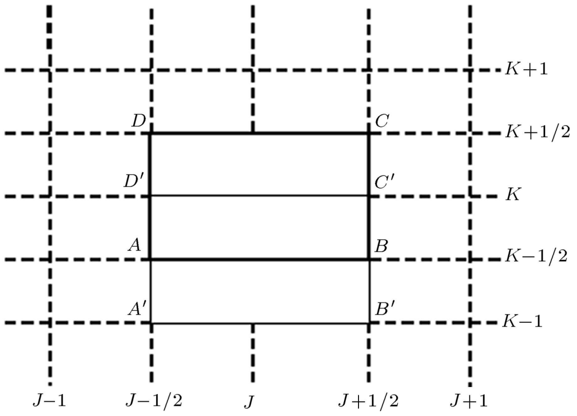

Figure 2. A computational cell.

Equation 2 becomes:

$$\left[\frac{\partial\Psi}{\partial x}\right]_{j,k-\frac{1}{2}}\Delta y_{AB}-\left[\frac{\partial\Psi}{\partial y}\right]_{j,k-\frac{1}{2}}\Delta x_{AB}$$

$$+\left[\frac{\partial\Psi}{\partial x}\right]_{j+\frac{1}{2},k}\Delta y_{BC}-\left[\frac{\partial\Psi}{\partial y}\right]_{j+\frac{1}{2},k}\Delta x_{BC}$$

$$+\left[\frac{\partial\Psi}{\partial x}\right]_{j,k+\frac{1}{2}}\Delta y_{CD}-\left[\frac{\partial\Psi}{\partial y}\right]_{j,k+\frac{1}{2}}\Delta x_{CD}$$

$$+\left[\frac{\partial\Psi}{\partial x}\right]_{j-\frac{1}{2},k}\Delta y_{DA}-\left[\frac{\partial\Psi}{\partial y}\right]_{j-\frac{1}{2},k}-\Delta x_{DA}=0, \tag{4}$$

where as an example, $\left[\frac{\partial\Psi}{\partial x}\right]_{j,k-\frac{1}{2}}$ is evaluated as the mean value over the area, $B'BC'D'AA'B'$, in Figure 2. After some algebraic manipulations, we get:

$$\left[\frac{\partial\Psi}{\partial x}\right]_{j,k-\frac{1}{2}}=$$

$$\frac{\Delta y_{AB}(\Psi_{j,k-1}-\Psi_{j,k})+\Delta y_{k-1,k}(\Psi_B-\Psi_A)}{S_{AB}}, \tag{5}$$

$$\left[\frac{\partial\Psi}{\partial y}\right]_{j,k-\frac{1}{2}}=$$

$$\frac{-\Delta x_{AB}(\Psi_{j,k-1}-\Psi_{j,k})-\Delta x_{k-1,k}(\Psi_B-\Psi_A)}{S_{AB}}, \tag{6}$$

where $\Delta x_{AB}=X_B-X_A$ and $\Delta x_{k-1,k}=X_{j,k}-X_{j,k-1}$. Therefore, we get:

$$\frac{\Delta\Psi}{\Delta n'.\Delta S}=\left[\frac{\partial\Psi}{\partial x}\right]_{j,k-\frac{1}{2}}\Delta y_{AB}-\left[\frac{\partial\Psi}{\partial y}\right]_{j,k-\frac{1}{2}}\Delta x_{AB}$$

$$=Q_{AB}(\Psi_{j,k-1}-\Psi_{j,k})+P_{AB}(\Psi_B-\Psi_A), \tag{7}$$

where:

$$Q_{AB}=\frac{(\Delta x_{AB}^2+\Delta y_{AB}^2)}{S_{AB}},$$

$$P_{AB}=\frac{(\Delta x_{AB}\Delta x_{k-1,k}+\Delta y_{AB}\Delta y_{k-1,k})}{S_{AB}},$$

where Ψ_A, X_A and Y_A are evaluated as the average of the four surrounding nodal values as follows:

$$X_A=0.25(X_{j-1,k}+X_{j,k}+X_{j-1,k-1}+X_{j,k-1}), \tag{8}$$

where S_{AB} is the area of side AB in cell $ABCD$ (i.e. area of cell $A'B'C'D'$). Other terms are defined similarly.

Note that Ashrafizadeh et al. [1,8,9] implemented the target wall velocity right at the wall nodes while we have done something different. Since the flow is inviscid, the tangential flow velocity adjacent to the solid wall is not zero. Therefore, if dense enough grids are used there, it is possible to cancel out the corner and boundary cells with no significant effects on the target velocity implementation. Thus, velocity is implemented at point B (Figure 3) instead of boundary node A. This treatment causes some simplification in the computer code and increases the stability and convergence rate of the present method.

DIRECT DESIGN METHOD

As mentioned previously, the coordinates of the interior nodes are defined in terms of the boundary nodes. Then, the unknown coordinates of the boundary nodes are transformed to the spine coordinates, (R_L, and R_U). Here, the particular grid generation technique and the direct design method used are described.

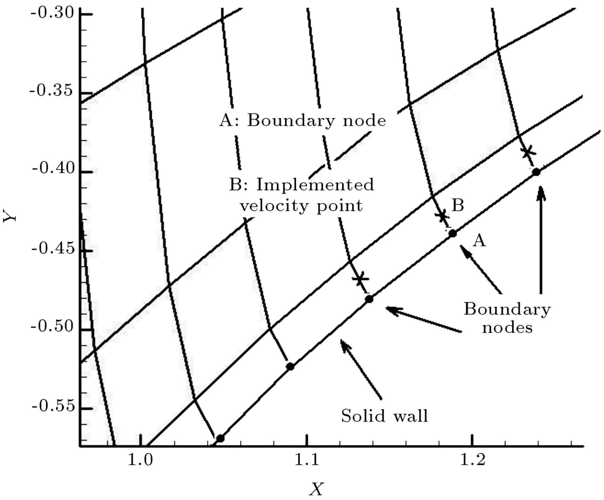

Figure 3. Boundary node and implemented velocity point.

Linear and Orthogonal Grid Generation

The general interpolation formulas for linear and non-linear algebraic grids are:

$$X(\xi,\eta) = H_1 * XL(\xi) + H_2 * XU(\xi) + Hx(\xi,\eta), \quad (9)$$

$$Y(\xi,\eta) = H_1 * YL(\xi) + H_2 * YU(\xi) + Hy(\xi,\eta), \quad (10)$$

$$0 \le \xi \le 1, \quad 0 \le \eta \le 1,$$

where ξ and η are the body fitted coordinates along and perpendicular to the surface boundary, respectively. For linear grids, we have:

$$H_1 = 1 - \eta, \quad H_2 = \eta, \quad Hx = Hy = 0,$$

while, for non-linear grids, we have:

$$H_1 = 2\eta^3 - 3\eta^2 + 1,$$

$$H_2 = -2\eta^3 + 3\eta^2,$$

$$H_3 = \eta^3 - 2\eta^2 + \eta,$$

$$H_4 = \eta^3 - \eta^2,$$

$$H_x(\xi,\eta) = H_3 * \partial X(\xi,\eta=0)/\partial\eta + H_4 * \partial X(\xi,\eta=1)/\partial\eta,$$

$$H_y(\xi,\eta) = H_3 * \partial Y(\xi,\eta=0)/\partial\eta + H_4 * \partial Y(\xi,\eta=1)/\partial\eta,$$

where H_x and H_y contain terms which enforce orthogonality at the wall boundaries [14]. In the final system of equations, H_x and H_y appear at the right hand side and are taken from the previous iteration. Now, the final forms of the internal node coordinates, in terms of the spine coordinates, are as follows:

$$X(i,j) = H_1 \cos(\theta_i) R_L(i) + H_2 \cos(\theta_i) R_U(i) + (H_1 + H_2) X_0(i) + Hx(i,j), \quad (11)$$

$$Y(i,j) = H_1 \sin(\theta_i) R_L(i) + H_2 \sin(\theta_i) R_U(i) + (H_1 + H_2) Y_0(i) + Hy(i,j), \quad (12)$$

where i and j designate the grid numbers in ξ and η directions, respectively.

Formulation of the Direct Design Method

Since in direct design methods the unknown nodal coordinates appear explicitly in the formulations, there are several non-linear terms due to the product of the geometrical and physical unknowns. In this section, the required linearizations related to the internal cells are discussed. The Newton linearization technique is applied as follows:

$$R^{n+1}.S^{n+1} = R^{n+1}.S^n + R^n.S^{n+1} - R^n.S^n, \quad (13)$$

where R and S are two arbitrary unknown variables and n is the iteration number. "Q_{AB}" and "P_{AB}" terms in Equation 7 contain physical and geometrical unknowns. Hence, the linearization of these two terms is discussed. In most cases, the P_{AB} term is small compared to the Q_{AB}. Therefore, the "Q" term is dominant and the "P" can be lagged. However, to increase the convergence rate of the method in "P" term, the quantities having index k (i.e. $\Delta x_{k-1,k}$ and $\Delta y_{k-1,k}$) are considered as unknowns, while the quantities with indices A and B are obtained from the previous iteration (lagged). The product of the unknown Ψ and the quantities with the index k make non-linear terms, which are linearized using Relation 13. When Newton's linearization method is used in the domain, the product of Ψ and "Q" terms becomes:

$$Q^{n+1}.\Psi^{n+1} = \frac{2(\Delta x^n.\Delta x^{n+1} + \Delta y^n.\Delta y^{n+1})}{S.\Psi^n} + Q^n.\Psi^{n+1} - 2Q^n.\Psi^n. \quad (14)$$

Now, replacing the Cartesian coordinates with the spine ones, Equation 4 can be cast in an algebraic form containing nodal values of R_L, R_U, and Ψ. After imposing the boundary conditions and assemblage, the system of equations becomes:

$$[A]\left\{\begin{array}{c}\{\Psi\}\\ \{R_U\}\\ \{R_L\}\end{array}\right\} = \{B\}. \quad (15)$$

A standard direct sparse solver is used to solve the system of equations for the unknown nodal values of R_L, R_U, and Ψ. This technique works well and converges fast.

In inverse design, the desired wall velocity distribution is presented based on the body fitted coordinate ($0 \le \xi^* \le 1$). Here, ξ^* is defined separately for each wall as $\frac{\xi}{\xi_{max}}$, where ξ is the coordinate along the wall and ξ_{max} is the total length of the wall. This way, the designer does not need to know the total duct length, which is an unknown itself. The solution starts with an initial guess. Besides, the ratio of outlet to inlet areas and the locations of the inlet and the outlet have to be known. An under relaxation factor is often needed for convergence.

Boundary Conditions

We have a Dirichlet boundary condition on all the flow boundaries as formal boundary conditions. Ψ is taken to be zero on the lower wall, while it is set to the volumetric flow rate on the upper wall. In the inlet and outlet, Ψ is taken proportional to the passing flow rate through stream tubes.

$\Psi_{\text{Lower wall}} = 0,$

$\Psi_{\text{Upper wall}}=\Psi_{\max}=\text{Entrance Area} * \text{Entrance Velocity},$

$\Psi_{\text{entrance}} = \Psi_{\max} * \eta$ at entrance,

$\Psi_{\text{exit}} = \Psi_{\max} * \eta$ at exit,

$0 <= \eta <= 1,$

where η is the curvilinear coordinate perpendicular to the wall.

Implementation of the Target Velocity

In inverse design, besides these formal boundary conditions, we need to prescribe the velocity distribution at the solid boundaries. Here, we have used $\frac{\partial\Psi}{\partial n'} = V_{Given}$ on the lower wall, as follows:

$$Q_{AB}.(\Psi_{j,k-1}-\Psi_{j,k})+P_{AB}.(\Psi_B-\Psi_A)=V_{Given}.\Delta S, \tag{16}$$

where ΔS stands for the elemental distances along the wall. The extra boundary condition for the upper wall is defined similarly. The linearization of term $Q_{AB}\Psi$ in the above extra boundary condition differs slightly from what was stated in Equation 14, which yields instability at the wall. The following form is appropriate to obtain satisfactory convergence:

$$\Psi^{n+1}Q^{n+1} = \frac{\Psi^n(\Delta x^n \Delta x^{n+1} + \Delta y^n \Delta y^{n+1})}{S} + \Psi^{n+1}Q^n - \Psi^n Q^n. \tag{17}$$

DESIGN CASES

Ideal straight nozzles, Ideal S-shaped ducts and branched ducts are studied as design examples. Computations were all run using a Pentium IV personal computer having 2.8 GHz CPU and 512 MB RAM (Table 1). Since the computed and the target pressure distributions along the solid walls are extremely close, only the target ones are plotted in the related figures.

Straight Nozzles

In nozzle design, if there is no region at the wall where the velocity decreases, the designer is confident that there is no adverse pressure gradient and, thus, no flow separation anywhere along the wall. We call this velocity profile the "ideal wall velocity distribution" (Figure 4) and a nozzle will be designed based on this ideal wall velocity distribution.

Here, we used Michael nozzle (widely used in subsonic wind tunnels) as our initial guess (Figures 5 and 6). The ratio of inlet to outlet area is taken to be 4. Note that the wall of the designed nozzle has a non-zero slope at the inlet, which conflicts with the inlet uniform flow condition (Figure 7). Making the nozzle longer reduces this slope. The slope can also be reduced or eliminated by specifying an undershoot in the target velocity distribution (it should be small enough to avoid separation). The results of this modification are shown in Figures 4 and 7; Figure 8 shows the convergence rate and the required CPU time of the method for this ideal nozzle design case.

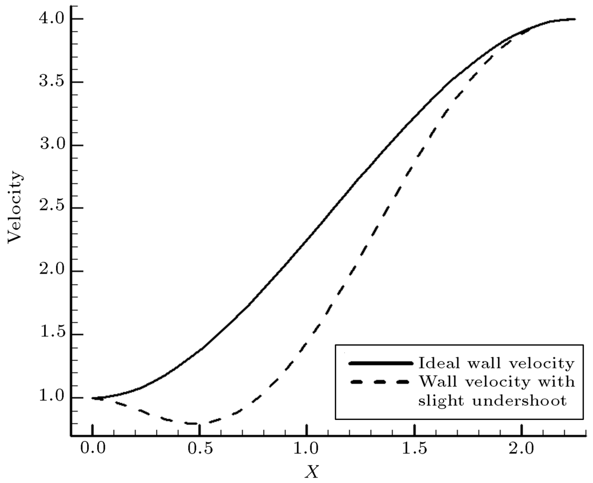

Figure 4. Ideal wall velocity distributions for a nozzle.

Table 1. The test cases studied here.

Applications	Grid	No. of Iteration	CPU Time (Sec.)	Under Relaxation Factor	Log (Residual)
Ideal Nozzles	51*20	18	3.6	0.15	-3
Ideal S-Shaped Ducts	51*20	286	53	0.115	-2.5
Branched Ducts	51*20	165	34.5	0.225	-1.3

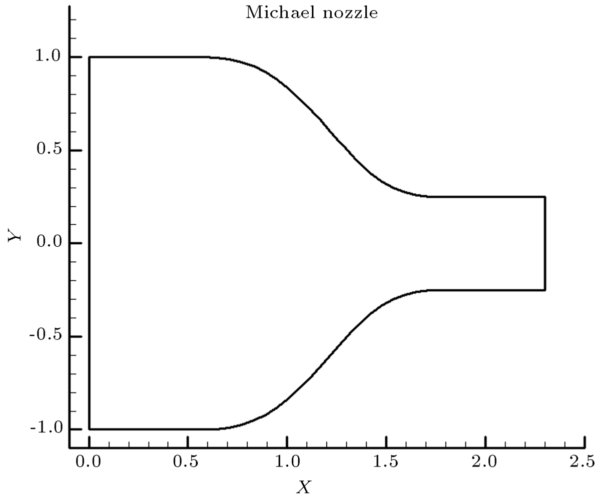

Figure 5. Michael nozzle.

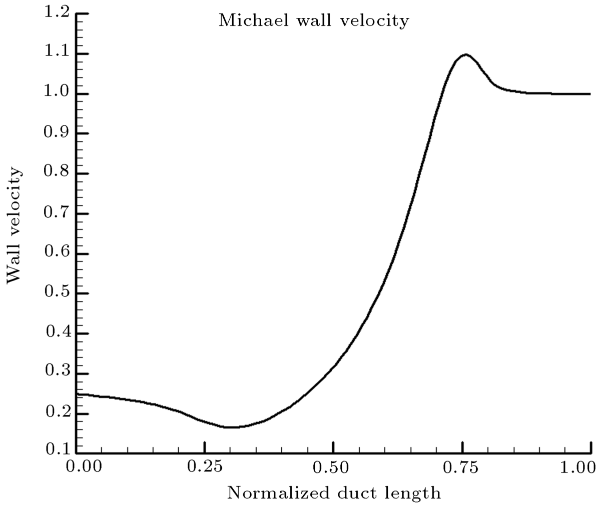

Figure 6. Michael wall velocity distribution.

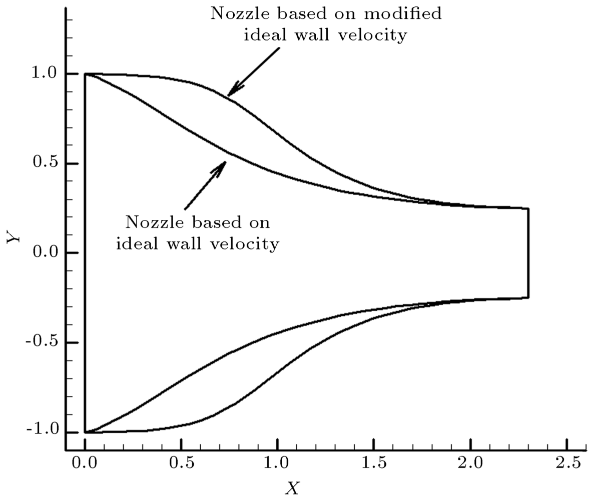

Figure 7. Designed nozzles based on ideal wall velocity distributions.

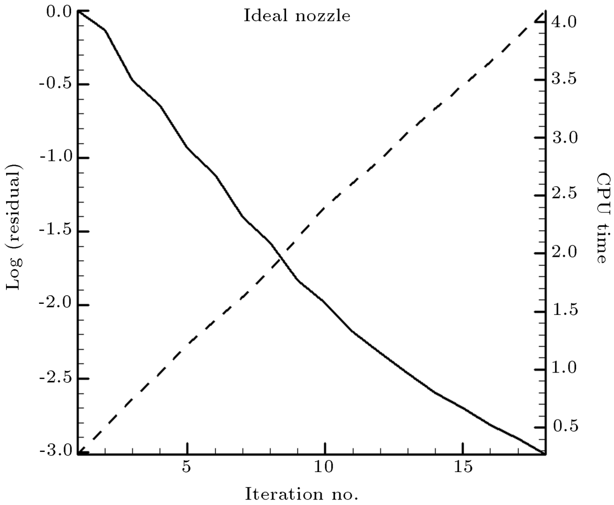

Figure 8. Convergence rate and CPU time for design of the ideal nozzle.

S-Shaped Ducts

One of the applications of direct design is in the intake of jet engines, which has a S-shaped geometry and acts like a diffuser. In these ducts, because of different curvatures at the upper and lower walls (Figure 9), the convergence of the direct design computer code becomes slower (Figure 10). Moreover, because of considerable adverse pressure gradients, the possibility of flow separation in a viscous flow is higher. Although adverse pressure gradients are not avoidable in such ducts, one prefers for S-shaped ducts without overshoots and undershoots in its wall velocity profile, in order to reduce the flow separation possibility (Figure 11).

Here, we consider a diffuser with area ratio (outlet to inlet) of 2. The diffuser length was assumed to be 6 (the outlet width is taken to be unity). The target (desired) wall velocity distribution is demonstrated in

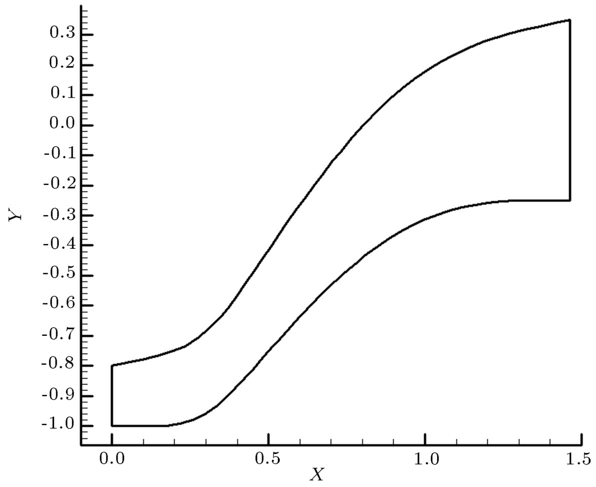

Figure 9. A typical S-shaped diffuser.

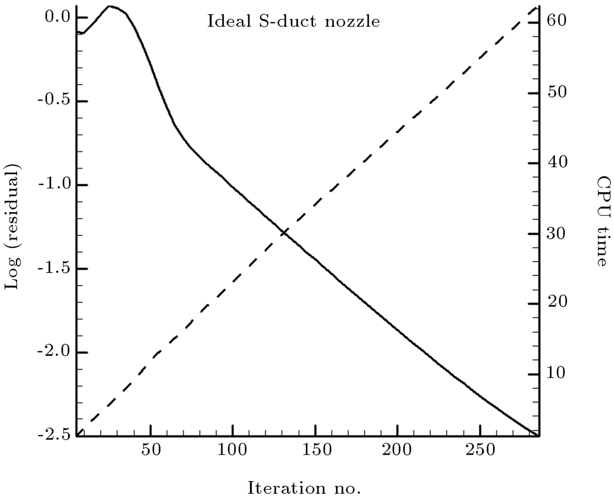

Figure 10. Convergence rate and CPU time for design of the ideal S-duct.

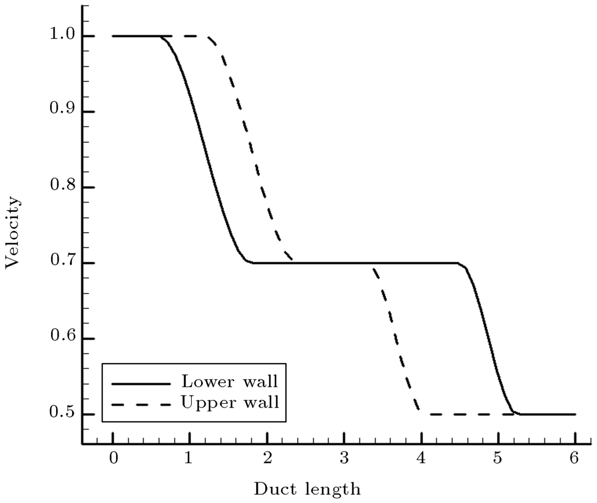

Figure 11. Wall velocity profile for an ideal S-shaped diffuser.

Figure 11 and the initial and final (desired) ducts are shown in Figure 12. Also, Figure 13 shows the wall velocity distributions of the initial S-duct.

Branched Ducts

Another application of the direct design approach is in designing branched ducts. These types of ducts have broad applications in air intake and exhaust manifolds of internal combustion engines. As mentioned before, a key difficulty in designing these types of ducts is the existence of a stagnation point, wherein the velocity is zero, causing a singularity that may lead to a divergence of the computer code.

Here, an asymmetric branched duct is considered as a design case. The splitter is composed of two circular arcs in the vicinity of the leading edge (Figure 14). This is our desired and final shape (just as a test case).

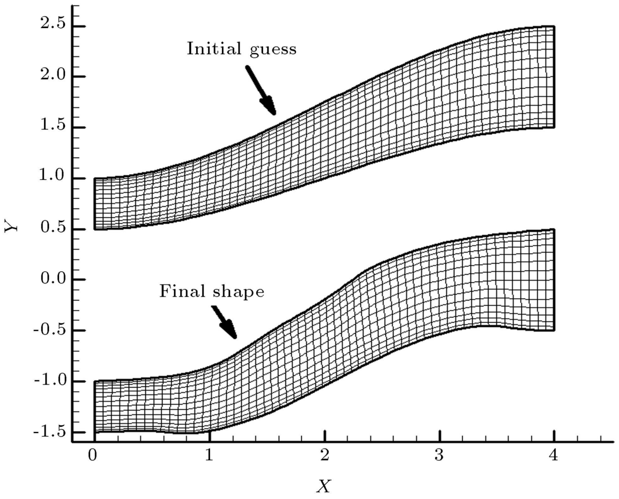

Figure 12. Initial and final shape for a S-shaped duct.

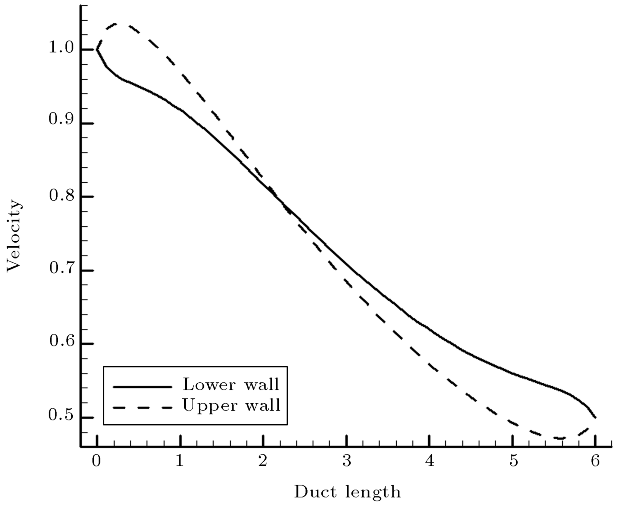

Figure 13. Wall velocity distribution for the initial S-shaped duct guess.

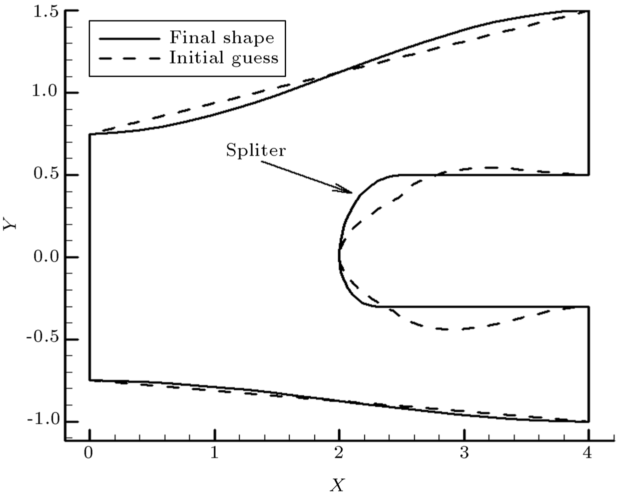

Figure 14. Initial guess and final shape of the branched duct.

Also, in this figure, the initial guess is shown, which is very different from the final shape. The wall velocity distributions of these two shapes are shown in Figure 15 (only for the splitter parts).

As reported in [9], after a few iterations, some wiggles appear near the leading edge (shown in Figure 16 after 3 iterations). The reason is that the diagonal coefficients of matrix A in Equation 15 become extremely small in the vicinity of the stagnation point. Examinations have shown that, near this point, the changes in surface geometry have little effect on the wall velocity profile. However, at other surface locations where the equations are well conditioned, the wall velocity profile is much more sensitive to the wall profile shape.

The following procedure was used to overcome this difficulty in [9]. The equations for geometrical unknowns (spine values) with nearly zero diagonal coefficients were removed from the equation set (matrix A), leaving a surface "patch" of nodes. The surface shape within the patch was then approximated by $y = f(s)$, where s is the arc length around the shape. The function was a cubic spline one, where the continuity of slopes at starting and ending points are satisfied.

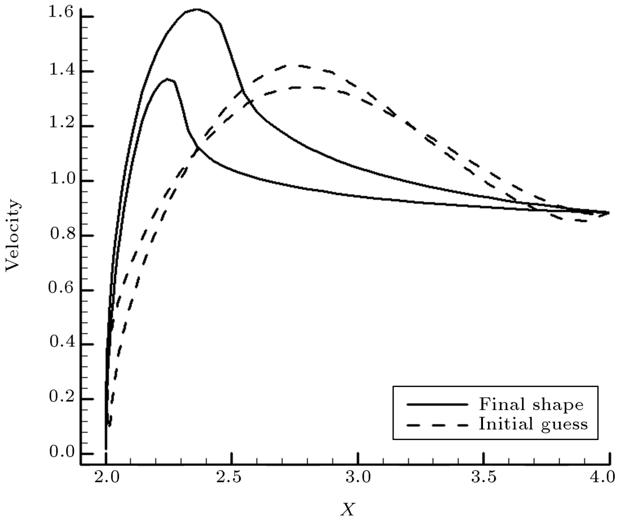

Figure 15. Initial guess and final wall velocity profiles of the branched duct.

In our method, the coefficient matrix does not change, but at each iteration a least square curve fitting (order 2 or 3) is used for several nodes in the vicinity of the stagnation point. Since this process starts from the first iteration where the wiggles have not yet appeared, the points which are used for curve fitting are not oscillatory ones. It is natural that in this curve, the condition of the slope continuity at starting and ending points is not satisfied. However, this makes no problem since the smooth shapes are usually used in such problems. This algorithm is simpler than the method used in [9], because there is no modification in the coefficient matrix and only a curve fitting is used at the end of each iteration. A problem occurring here was that the obtained shape did not pass through the stagnation point which needs to be modified (Figure 17).

In the modified least square algorithm, passing the curve through the stagnation point is enforced (Figure 18) and the figure shows the new shape after 20 iterations. Therefore, the used curve fitting is not a pure or natural least square algorithm. Since the fitted curve is not completely similar to the target shape, the reduction of error from the first iteration (initial guess) to the best computed shape is up to two orders (this is enough in practical applications). Figure 19 shows the computed and target shapes and Figure 20 is related to their velocities. As seen, the

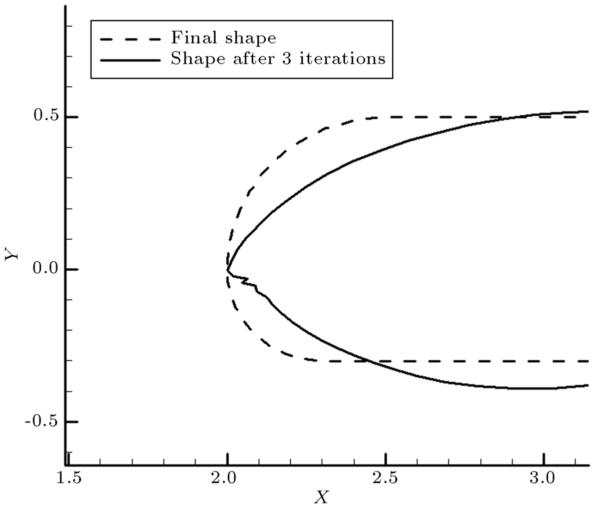

Figure 16. The wiggles at the vicinity of the leading edge in the branched duct case.

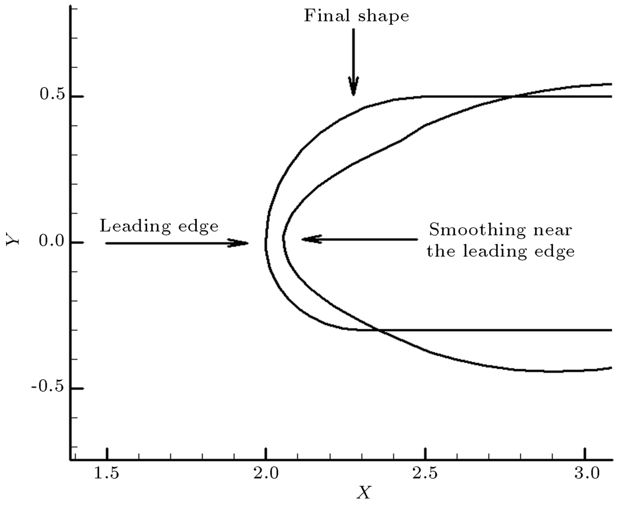

Figure 17. The effect of smoothing near the leading edge in the branched duct.

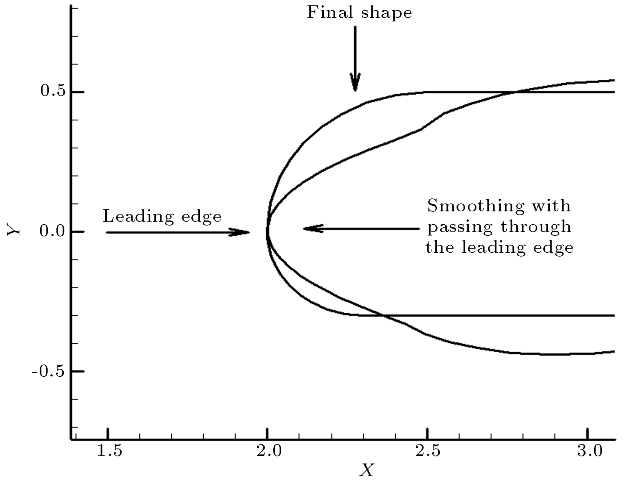

Figure 18. Effect of smoothing with enforcing to pass the curve through the leading edge in the branched duct.

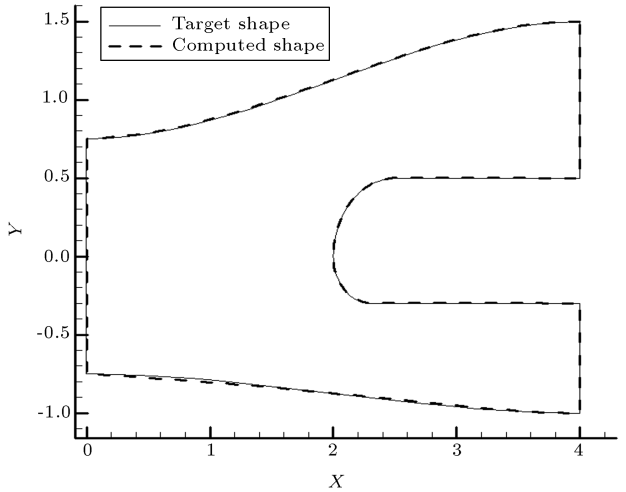

Figure 19. Comparison between target and computed shapes in the direct design process.

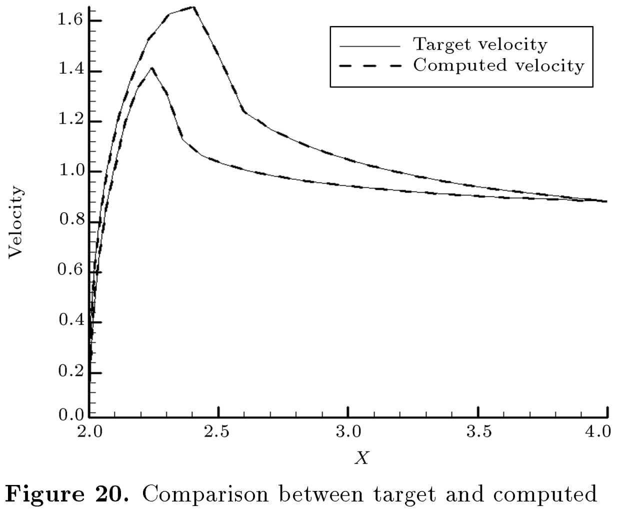

Figure 20. Comparison between target and computed velocities along the splitter in the direct design process.

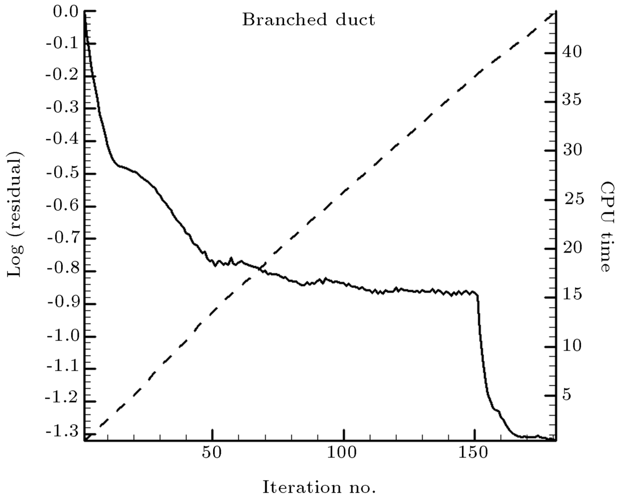

Figure 21. Convergence rate and CPU time for design of the branched duct.

result is satisfactory while the reduction of error is less than two orders (see Table 1). Figure 21 shows the convergence rate and the required CPU time of the method for this branched duct design. Contrary to the two design cases, the convergence rate has a big flat zone and residual decreasing is low.

CONCLUSION

In this work, we modified and applied the direct design technique, in which the body coordinates appear as dependent variables in the context of internal flows. A cell-vertex finite volume method and an orthogonal grid were used to discretize the placeLaplace equation and the solution domain. Our test cases included ideal straight nozzles, ideal S-shaped ducts and branched ducts. This work is novel due to:

- The use of an orthogonal grid;
- The method's applicability for complex geometries such as branched ducts (including stagnation points);
- The method's high convergence rates.

ACKNOWLEDGMENT

This work was sponsored by Sharif University of Technology, research funds, under the appreciated support of Dr. Kermanshah.

REFERENCES

1. Ashrafizadeh, A., Raithby, G.D. and Stubley, G.D. "Direct design of ducts", *Journal Fluids Engineering, ASME Transactions*, **125**, pp. 158-165 (January 2003).
2. Chin-Hsiang, Cheng and Chun-Yin, Wu "An approach combining body fitted grid generation and conjugate

gradient methods for shape design in heat conduction problems", *Numerical Heat Transfer, Part B*, **37**(1), pp. 69-83 (2000).

3. Jameson, A. "Optimal design via boundary control", *Optimal Design Methods for Aeronautics*, AGARD, pp. 3.1-3.33 (1994).
4. Stanitz, J.D. "A review of certain inverse methods for the design of ducts with 2- or 3-dimensional potential flows", *Proceedings of the Second International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-II),* The Pennsylvania State University, University Park, PA, pp. 26-28 (October 1987).
5. Dedoussis, V., Chaviaropoulos, P. and Papailliou, K. "Rotational compressible design method for two-dimensional internal flows", *AIAA Journal*, **31**, pp. 551-558 (1993).
6. Chaviaropoulos, P., Dedoussis, V., and Papailliou, K. "On the 3-D inverse potential target pressure problem. Part 1: Theoretical aspects and method formulation", *Journal Fluid Mechanics*, **282**, pp. 121-146 (1995).
7. Chaviaropoulos, P., Dedoussis, V. and Papailliou, K. "On the 3-D inverse potential target pressure problem. Part 2: Numerical aspects and application to duct design", *Journal Fluid Mechanics*, **282**, pp. 147-162 (1995).
8. Ashrafizadeh, A., Raithby, G.D. and Stubley, G.D. "Direct design of shape", *Numerical Heat Transfer - Part B*, **41**, pp. 501-510 (2002).
9. Ashrafizadeh, A., Raithby, G.D. and Stubley, G.D. "Direct design of airfoil shape with a prescribed surface pressure", *Part B: Fundamentals Journal Numerical Heat Transfer*, **46**(6), pp. 505-527 (December 2004).
10. Ghadak, F., Taiebi-Rahni, M. and Ashrafizadeh, A. "A direct design approach using the Euler equations", *Journal of Inverse Problems in Science and Engineering*, **16**(2), pp. 217-231 (2008).
11. Volpe, G. and Melnik, R.E. "The role of constraints in the inverse design problem for transonic airfoils", *AIAA Paper 81* (1981).
12. Drela, M. "Two-dimensional transonic aerodynamic design and analysis using the Euler equations", MIT Gas Turbine & Plasma Dyn. Lab. Report, **187** (1986).
13. Fletcher, C.A.J. "Computational techniques for fluid dynamics", *Springer-Verlag*, **1**, Berlin (1991).
14. Kowalski, E.J. "Boundary fitted coordinate systems for arbitrary computational regions", *Numerical Grid Generation Techniques*, NASA Conf. Publications, **2166**, pp. 331-353 (1980).