14th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 14 – May 24 - 26, 2011, Email: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

Supersonic Nozzle Inverse Design Using Analytic Gradient and Inviscid Finite Volume Aerodynamic Simulation on Unstructured Grids

M. B. Azab* , M. I. Mustafa†, C. Ollivier-Gooch‡

Abstract: A gradient based aerodynamic optimizer has been built for nozzle inverse design problem. Solution sensitivity at nozzle walls is calculated using the analytic solution of quasi one dimensional nozzle flow while the flow field solution is computed using CFD finite volume solver. The gradient of the inverse design objective function is computed using the solution sensitivity at nozzle walls which is computed analytically, and using also the computed Mach distribution resulting from CFD simulation. As the nozzle wall contour changes during optimization iterations; we used linear spring mesh movement analogy to adapt the mesh grid points in the entire flow field to the new nozzle contour shape. A supersonic inverse design test case is presented and it shows the success of the proposed method as an aerodynamic design and optimization technique.

1. Introduction

Aerodynamic design using numerical optimization has been widely used in aerospace industry nowadays [1]. Gradient based optimization techniques is known to be faster in finding the optimal point in a design space compared to the non-gradient based optimization techniques[2]. Quadratic programming is the fastest among gradient based optimization methods and hence it is implemented in our research[3].

We use MATLAB® optimization toolbox as an optimizer; Quasi- Newton line search method is used with BFGS approximation of the Hessian matrix; it is chosen due to its super-linear/quadratic convergence to the optimal design point [3, 4].

The objective function gradient can be calculated using finite differences or adjoint strategy and it requires a lot of effort and programming [5]. We present results of using CFD simulations to calculate the objective function value, and using the analytic quasi one dimensional flow equations to calculate the solution sensitivity (wall Mach sensitivity) with respect to geometry design variables. The nozzle contour is represented by higher order least squares spline which is controlled by geometry design points. In order to avoid grid regeneration after updating the nozzle contour during optimization iterations, we use spring analogy mesh morphing scheme to update the mesh grid in the entire flow field to fit with the new boundary shape (nozzle walls) [6, 7]. Typical aerodynamic optimization cycle is shown in figure (1). We present a supersonic inverse design test case which shows that the use of a simple analytic solution as a gradient evaluation technique can be of useful use in aerodynamic design.

* Advanced Numerical Simulation Laboratory, the University Of British Columbia, Vancouver, Canada.
† Egyptian Armed Forces, Egypt.
‡ Mechanical Engineering Department, The University Of British Columbia, Vancouver, Canada.

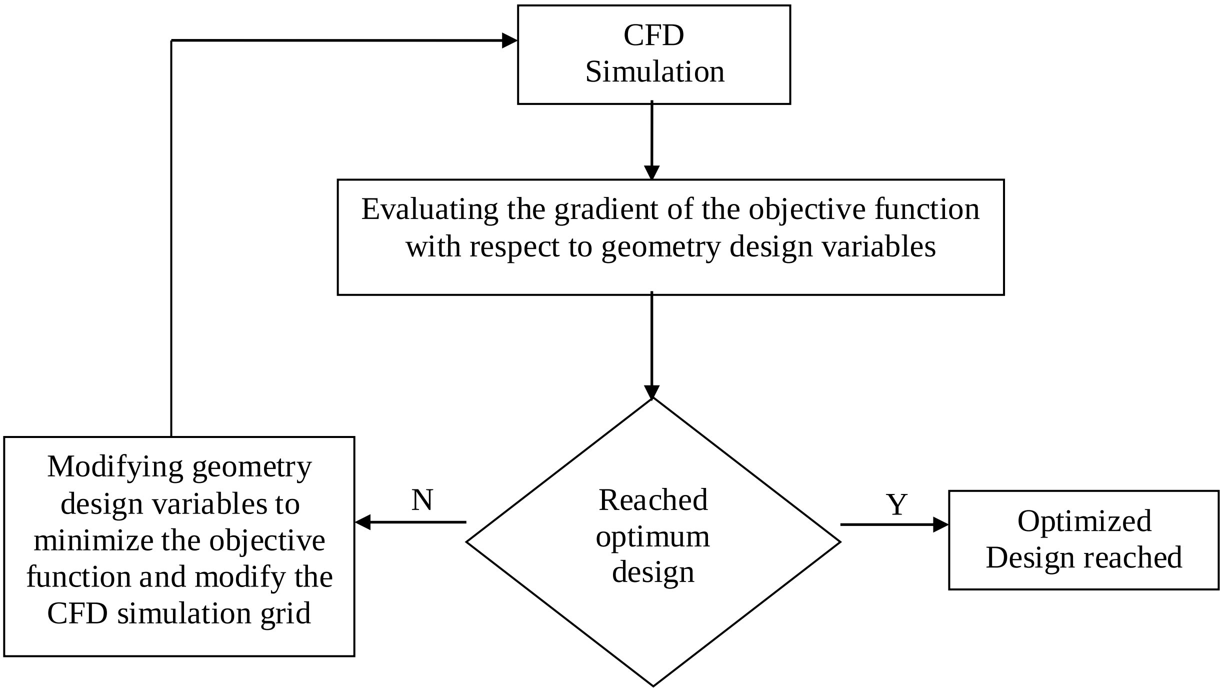

CFD Simulation
Evaluating the gradient of the objective function with respect to geometry design variables
Reached optimum design
N
Y
Modifying geometry design variables to minimize the objective function and modify the CFD simulation grid
Optimized Design reached

Figure (1): Schematic flowchart of the aerodynamic optimization cycle.

2. Flow Governing Equations

Euler's flow model is used, the integral form of the two dimensional Euler's equations can be written as,

$$\frac{\partial}{\partial t}\iint Q.dV + \oint_{d\Omega} \vec{F}.\hat{n}.dl = 0\,, \tag{1}$$

where $\hat{n} = n_x i + n_y j$ is the outward pointing normal to the control volume faces, and

$$Q = \begin{bmatrix} \rho \\ \rho.u \\ \rho.v \\ E_t \end{bmatrix},\ \vec{F}.\hat{n} = \begin{bmatrix} \rho.U \\ \rho.U.u + n_x.p \\ \rho.U.v + n_y.p \\ (E_t + p).U \end{bmatrix},$$

$$p = (\gamma - 1).\left[E_t - \rho.\frac{(u^2+v^2)}{2}\right],$$

$$U = n_x u + n_y v$$

Equation (1) is integrated for each control volume in the unstructured mesh used for numerical simulation. Computing element faces fluxes is done using Roe flux differencing scheme [8]. The flow properties at control volume center point is updated as follows,

$$Q^{n+1} = Q^n + \Delta Q \tag{2}$$

3. Flow Sensitivity Calculations

In this section, we present the equations needed to compute the flow sensitivity values at the wall for Laval nozzle flow and we will compare it with the flow sensitivity values computed using finite difference. Figure (2) shows the mesh grid used for CFD simulation, the nozzle contour is described by the following equation,

$$y = \pm\left(1 + \frac{x}{10}\right), \;\; 0 \leq x \leq 10.$$

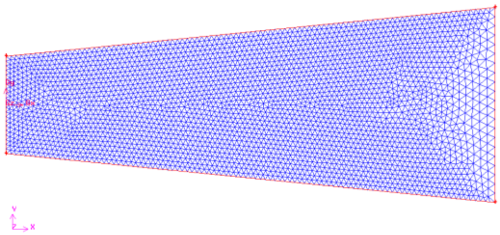


Figure (2): Unstructured triangular mesh of Laval nozzle

The flow sensitivity with respect to geometry design variables using finite difference strategy requires two CFD simulations, and is calculated as follows,

- Calculate the flow solution Q as the geometry design variables is changed from its original value y_{io} to $y_{io} + \epsilon$. Using CFD simulation.
- Calculate the flow solution Q as the geometry design variables is changed y_{io} to $y_{io} - \epsilon$ using additional CFD simulation.
- Solution sensitivity is calculated as,

$$\frac{\partial Q}{\partial y_{io}} = \frac{Q(x_{io}+\varepsilon) - Q(x_{io}-\varepsilon)}{2\varepsilon} \tag{3}$$

The solution sensitivity can be approximated using the analytical quasi one dimensional solution. As an example of we present a procedure to evaluate density sensitivity with respect to specific y location in the nozzle contour, which can be summarized as follows,

- Let $A_i = y_i$.

$$\frac{\partial \rho}{\partial y_i} = \left\{ \frac{\partial \frac{\rho}{\rho^*}}{\partial M} \Big/ \frac{\partial \frac{A}{A^*}}{\partial M} \right\} \cdot \frac{A^*}{\rho^*} \tag{4}$$

Figure (3) shows a comparison between the density sensitivity computed using analytical quasi one dimensional relations as of equation (4), and the density sensitivity calculated using finite difference as of equation (3). In this validation case, nozzle inlet was at sonic speed, the geometry design variable is the wall point at the mid section of the nozzle. The analytical density sensitivity value resulting from equation (4) is (-0.222) while the finite difference density sensitivity using equation (3) is of value (-0.251). The analytical sensitivity value is of the same order of magnitude of the finite difference sensitivity and doesn’t require two expensive CFD simulations to be evaluated. This will dramatically reduce the amount of computational work needed to evaluate the objective function gradient needed for gradient based optimizer.

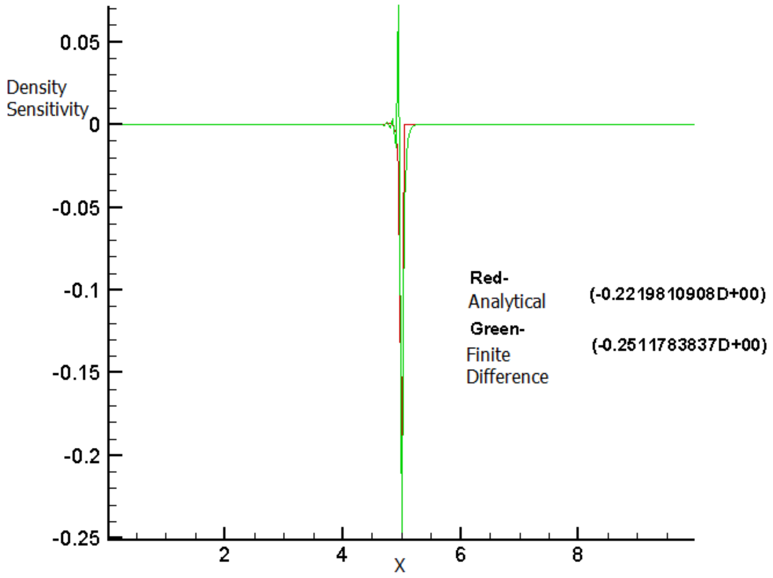


Figure (3): Density sensitivity computed using finite difference "green" and analytical quasi one-dimensional relations "red".

4. Geometry Parameterization

The wall contour can be represented using a higher order polynomial, to prevent obtaining oscillations that gives non smooth geometry; the wall contour polynomial is obtained using number of control points larger than the number of its unknown coefficients. The higher order polynomial that is used to represent the nozzle contour take the form:

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 \qquad (5)$$

$where \quad 0 \le x \le 10 \;\;,$

we use 11 control points to find the wall contour as shown in figure (4); the resulting least squares system can be written as follows,

$$[A]\cdot\{P\} = \{b\}$$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 & x_1^6 & x_1^7 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & x_2^6 & x_2^7 \\ 1 & x_3 & x_3^2 & x_3^3 & x_3^4 & x_3^5 & x_3^6 & x_3^7 \\ 1 & x_4 & x_4^2 & x_4^3 & x_4^4 & x_4^5 & x_4^6 & x_4^7 \\ 1 & x_5 & x_5^2 & x_5^3 & x_5^4 & x_5^5 & x_5^6 & x_5^7 \\ 1 & x_6 & x_6^2 & x_6^3 & x_6^4 & x_6^5 & x_6^6 & x_6^7 \\ 1 & x_7 & x_7^2 & x_7^3 & x_7^4 & x_7^5 & x_7^6 & x_7^7 \\ 1 & x_8 & x_8^2 & x_8^3 & x_8^4 & x_8^5 & x_8^6 & x_8^7 \\ 1 & x_9 & x_9^2 & x_9^3 & x_9^4 & x_9^5 & x_9^6 & x_9^7 \\ 1 & x_{10} & x_{10}^2 & x_{10}^3 & x_{10}^4 & x_{10}^5 & x_{10}^6 & x_{10}^7 \\ 1 & x_{10} & x_{11}^2 & x_{11}^3 & x_{11}^4 & x_{11}^5 & x_{11}^6 & x_{11}^7 \end{bmatrix} \cdot \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \end{Bmatrix} \qquad (6)$$

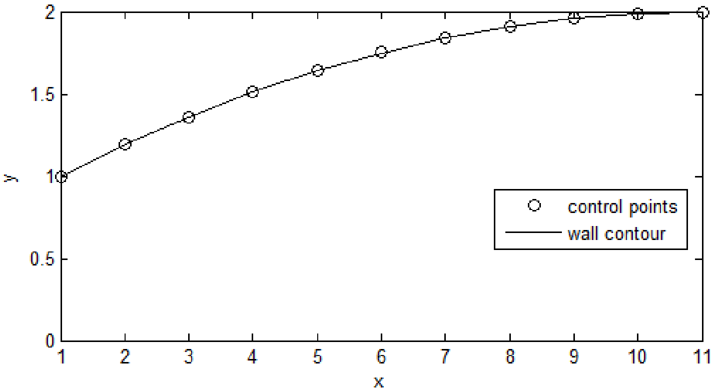


Figure (4): schematic drawing of a nozzle contour and its 11 control points

Solution of this least squares linear system is done using singular value decomposition (SVD) method as it is the best method for solving ill conditioned linear systems [9]; the condition number of the above system is of order 10^7. We use (**dgesvd)** function of LAPACK® library to find the SVD decomposition of the least squares system in equation (6) and use it to compute pseudo inverse of the coefficient matrix and the polynomial coefficients as follows,

$\{P\} = [A]^{\dagger} \cdot \{b\}$

$$\begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{Bmatrix} = \begin{bmatrix} a^{\dagger}_{1,1} & a^{\dagger}_{1,2} & a^{\dagger}_{1,3} & a^{\dagger}_{1,4} & a^{\dagger}_{1,5} & a^{\dagger}_{1,6} & a^{\dagger}_{1,7} & a^{\dagger}_{1,8} & a^{\dagger}_{1,9} & a^{\dagger}_{1,10} & a^{\dagger}_{1,11} \\ a^{\dagger}_{2,1} & a^{\dagger}_{2,2} & a^{\dagger}_{2,3} & a^{\dagger}_{2,4} & a^{\dagger}_{2,5} & a^{\dagger}_{2,6} & a^{\dagger}_{2,7} & a^{\dagger}_{2,8} & a^{\dagger}_{2,9} & a^{\dagger}_{2,10} & a^{\dagger}_{2,11} \\ a^{\dagger}_{3,1} & a^{\dagger}_{3,2} & a^{\dagger}_{3,3} & a^{\dagger}_{3,4} & a^{\dagger}_{3,5} & a^{\dagger}_{3,6} & a^{\dagger}_{3,7} & a^{\dagger}_{3,8} & a^{\dagger}_{3,9} & a^{\dagger}_{3,10} & a^{\dagger}_{3,11} \\ a^{\dagger}_{4,1} & a^{\dagger}_{4,2} & a^{\dagger}_{4,3} & a^{\dagger}_{4,4} & a^{\dagger}_{4,5} & a^{\dagger}_{4,6} & a^{\dagger}_{4,7} & a^{\dagger}_{4,8} & a^{\dagger}_{4,9} & a^{\dagger}_{4,10} & a^{\dagger}_{4,11} \\ a^{\dagger}_{5,1} & a^{\dagger}_{5,2} & a^{\dagger}_{5,3} & a^{\dagger}_{5,4} & a^{\dagger}_{5,5} & a^{\dagger}_{5,6} & a^{\dagger}_{5,7} & a^{\dagger}_{5,8} & a^{\dagger}_{5,9} & a^{\dagger}_{5,10} & a^{\dagger}_{5,11} \\ a^{\dagger}_{6,1} & a^{\dagger}_{6,2} & a^{\dagger}_{6,3} & a^{\dagger}_{6,4} & a^{\dagger}_{6,5} & a^{\dagger}_{6,6} & a^{\dagger}_{6,7} & a^{\dagger}_{6,8} & a^{\dagger}_{6,9} & a^{\dagger}_{6,10} & a^{\dagger}_{6,11} \\ a^{\dagger}_{7,1} & a^{\dagger}_{7,2} & a^{\dagger}_{7,3} & a^{\dagger}_{7,4} & a^{\dagger}_{7,5} & a^{\dagger}_{7,6} & a^{\dagger}_{7,7} & a^{\dagger}_{7,8} & a^{\dagger}_{7,9} & a^{\dagger}_{7,10} & a^{\dagger}_{7,11} \\ a^{\dagger}_{8,1} & a^{\dagger}_{8,2} & a^{\dagger}_{8,3} & a^{\dagger}_{8,4} & a^{\dagger}_{8,5} & a^{\dagger}_{8,6} & a^{\dagger}_{8,7} & a^{\dagger}_{8,8} & a^{\dagger}_{8,9} & a^{\dagger}_{8,10} & a^{\dagger}_{8,11} \end{bmatrix} \cdot \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \end{Bmatrix}, \qquad (7)$$

To find the dependency of the wall profile(y_{wall}) at certain x location (x_m) on one of the design variables y_k, the pseudo inverse $A^{\dagger}$ is used as follows,

$$\frac{\partial y_{wall}}{\partial y_k}(x_m) = \frac{\partial a_0}{\partial y_k} + \frac{\partial a_1}{\partial y_k} x_m + \frac{\partial a_2}{\partial y_k} {x_m}^2 + \frac{\partial a_3}{\partial y_k} {x_m}^3 + \frac{\partial a_4}{\partial y_k} {x_m}^4 + \frac{\partial a_5}{\partial y_k} {x_m}^5 + \frac{\partial a_6}{\partial y_k} {x_m}^6 + \frac{\partial a_7}{\partial y_k} {x_m}^7 \qquad (8)$$

where $\frac{\partial a_i}{\partial y_k}, i = 1..7$ are the eight numerical elements of the kth column in the pseudo inverse matrix $A^{\dagger}$. The above procedure can be modified to apply any geometric constrained according to Mohammad Azab and Carl Ollivier-Gooch [5].

5. Mesh Morphing

We use the linear spring analogy [5, 6, 10] to adapt the grid in the flow field as the wall contour is changed by the optimizer. This saves the computational effort needed for regenerating a new mesh grid based of the new boundary shape (nozzle contour). The mesh grid is treated as a net of springs, the difference between the original boundary shape and the new boundary profile is used as displacement vector applied to this spring net. The final mesh grid shape is the shape that satisfies the equilibrium of the linear spring system. The edge stiffness is inversely proportional to its length; the nodal forces at edge end points related to displacement by Hook's law as follows [5],

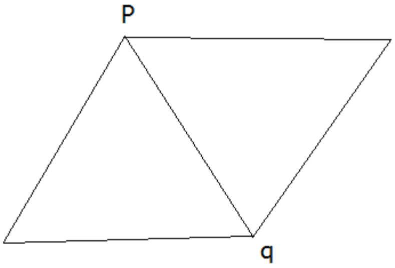


Figure (5): General face in a triangular mesh

$$\begin{bmatrix} F_{px} \\ F_{py} \\ F_{qx} \\ F_{qy} \end{bmatrix} = \frac{1}{l_{pq}} \cdot \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} u_{px} \\ u_{py} \\ u_{qx} \\ u_{qy} \end{bmatrix}, \tag{9}$$

where l_{pq} is the length of the edge as shown in figure (5). After assembly of the global stiffness matrix, the system of equations that relates grid point displacement with nodal forces can be written as

$$[K].\bar{U} = \{F\}, \tag{10}$$

where U is the displacement vector of the mesh points. The displacement vector U consist of two groups, the interior mesh points displacement U_i and the boundary mesh points U_b. Therefore Eq. (10) can be written as

$$\begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix} \cdot \begin{bmatrix} U_i \\ U_b \end{bmatrix} = \begin{bmatrix} 0 \\ F_b \end{bmatrix}, \tag{11}$$

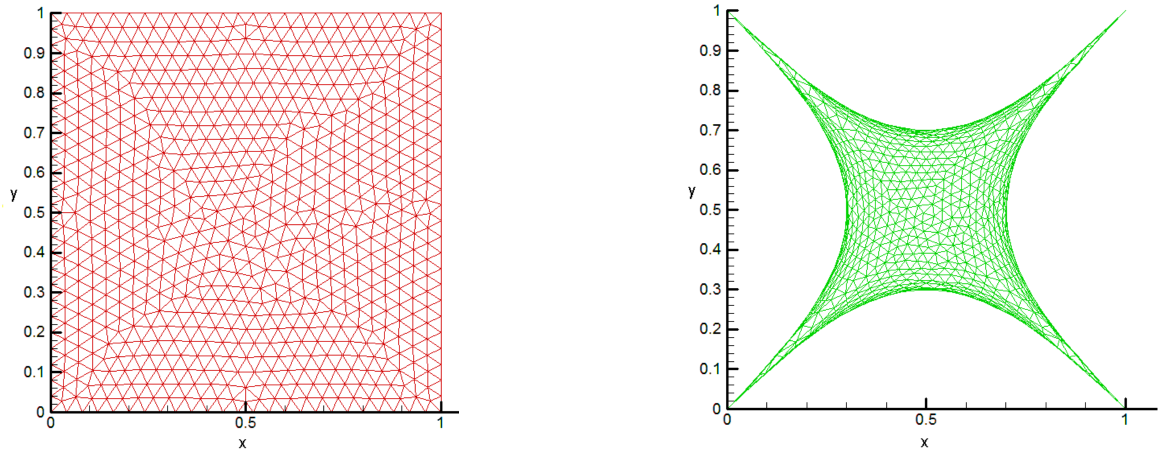
We do not need to know the values of boundary nodal forces F_b because the boundary points displacement vector U_b is known explicitly; it is basically the deformation required in the nozzle contour to minimize the objective function; therefore, equation (11) can be written as,

$$[K_{ii}]\{U_i\} = -[K_{ib}]\{U_b\}, \tag{12}$$

The stiffness matrices $[K_{ii}], [K_{ib}]$ depends on the mesh face lengths, which is changed during the mesh morphing stage; therefore equation (12) is a non linear equation and need to be solved iteratively as follows,

$$[K_{ii}]^k \cdot \vec{r}_i^{\,k+1} = [K_{ii}]^k \cdot \vec{r}_i^{\,k} - [K_{ib}]^k . \{U_b\}, \tag{13}$$

where $\vec{r}_i = \begin{Bmatrix} x \\ y \end{Bmatrix}$ the position vector of the internal nodes in the mesh grid. Equation (13) is obtained by substituting $\{U_i\} = \vec{r}_i^{\,k+1} - \vec{r}_i^{\,k}$; equation (13) is solved iteratively till $\|U_i\| \le tol$, where *tol* is mesh morphing solver specified tolerance. We use *tol* = 10^{-7} in our mesh morphing solver. As a validation of the above procedure, the proposed mesh morphing procedure is applied to a rectangular square domain with unstructured triangular mesh, the boundaries as deformed and the mesh is morphed in order to adapt to the new boundary shape; the above procedure successfully adapt the grid in the entire field. More than 50% reduction in the domain total area in this validation case as shown in figure (6) but the proposed non-linear mesh morphing solver showed the robustness of the proposed scheme.



a) before mesh morphing

b) after mesh morphing

Figure (6): Mesh morphing validation case, deformation of unstructured grids in a square domain.

6. Supersonic Nozzle Inverse Design

In this section we present an inverse design test case for a supersonic nozzle. Supersonic operation of a nozzle is of great importance for rocket engine.

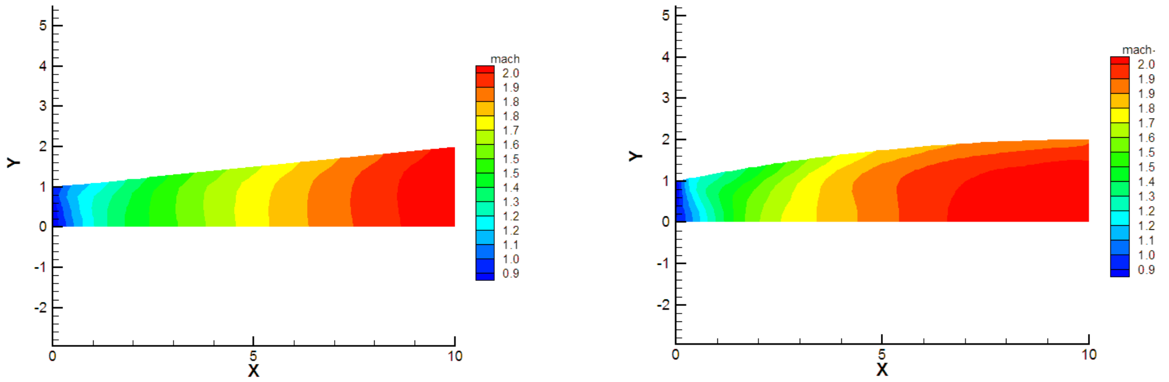
Inverse design is basically trying to find a nozzle contour that produces wall Mach distribution identical to a specified wall Mach distribution. We use inverse design test case because the optimization space contains only one minima of the objective function. The objective function, (F), to be minimized and its gradient can be written as,

$$F = \oint_{wall} (M - M_t)^2 \cdot dx$$

$$\frac{\partial F}{\partial y_{k.design}} = \oint_{wall} 2 \cdot (M - M_t) \cdot \frac{\partial M}{\partial y_{wall}} \cdot \frac{\partial y_{wall}}{\partial y_{k.design}} \tag{14}$$

Noting that $(M - M_t)$ is calculated from CFD simulation, $\frac{\partial M}{\partial y_{wall}}$ is calculated using analytical quasi one-dimensional relations, and $\frac{\partial y_{wall}}{\partial y_{k.design}}$ is computed using equation (8).

Figure (7) shows the initial and target nozzle geometries with a surface plot of the Mach distribution.



a) Initial geometry and Mach solution **b) Target geometry and Mach solution**

Figure (7): initial and target nozzle Mach distribution

We use only nine wall contour control points as we fix the inlet and exit area, so we excluded the first and last nozzle wall control points from the optimization design space. The optimization results showed that the proposed technique for evaluating the solution sensitivity and the objective function gradient was successful. Optimization convergence history is shown in figure (8). The resulting inverse design contour is in excellent match with the target contour as shown in figure (9).

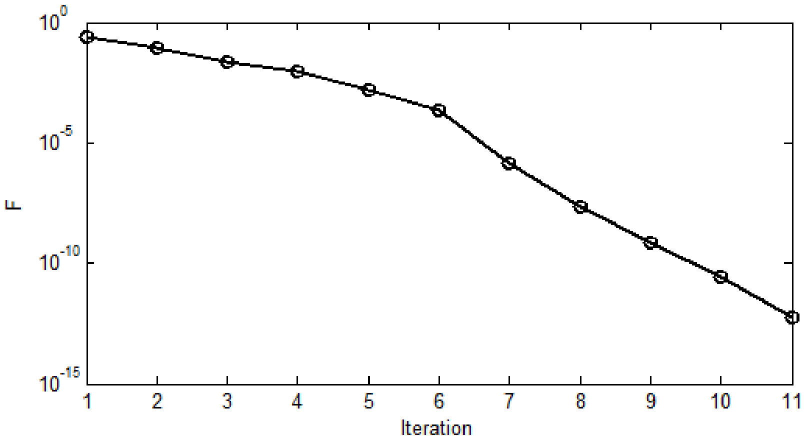


Figure (8): Optimization convergence history

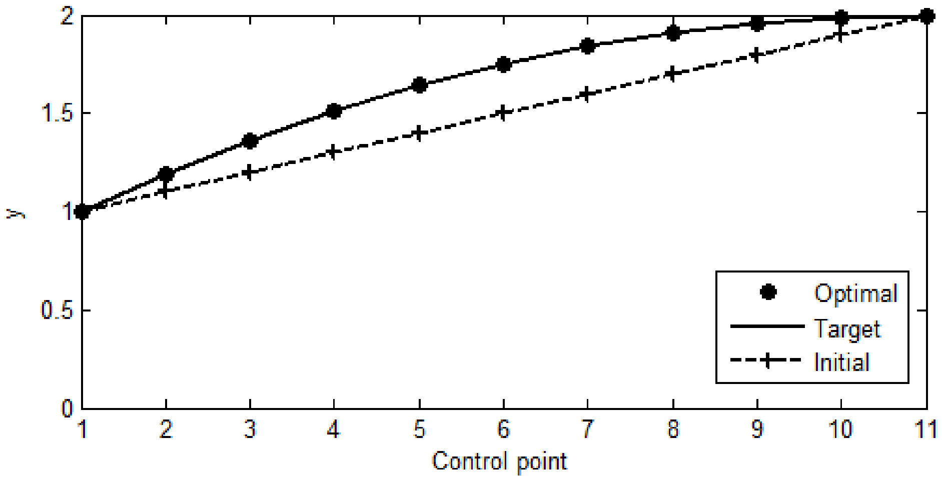


Figure (9): Inverse design profile

7. Conclusion

We demonstrated the concept of using the quasi one-dimensional relations as a solution sensitivity evaluator; we used the computed sensitivity to evaluate the objective function gradient. The objective function value is calculated using 2nd order Euler's CFD simulation. Although the solution sensitivity is not accurate enough (error 18%) but the computed gradient direction is almost the same, so the error in the gradient magnitude will not affect the optimizer because it uses line search technique. The nozzle inverse design test case demonstrated a success in using low order and computational cheap flow models in evaluating the objective function gradient, while evaluating the objective function value using expensive CFD simulation. This can also be applied in rocket motor design and optimization using quasi one-dimensional chemically reacting model.

References

[1] Jameson, A. Optimum transonic wing design using control theory. p. 253 (Springer Netherlands, 2003).

[2] Nadarajah, S. and Jameson, A. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. AIAA paper, 2000, 667, 2000.

[3] Nocedal J, W.S.J., ed. Numerical Optimization. (Springer New York, 2006).

[4] Zhu, C., Byrd, R.H., Lu, P. and Nocedal, J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS), 1997, 23(4), 550-560.

[5] Azab, M.B., and Ollivier-Gooch C. F. Higher Order Two Dimensional Aerodynamic Optimization Using Unstructured Grids and Adjoint Sensitivity Computations. 48th AIAA Aerospace sciences meetingOrlando FL, 2010).

[6] Farhat, C., Degand, C., Koobus, B. and Lesoinne, M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. Computer methods in applied mechanics and engineering, 1998, 163(1-4), 231-245.

[7] Masuda, H., Yoshioka, Y. and Furukawa, Y. Interactive mesh deformation using equality-constrained least squares. Computers & Graphics, 2006, 30(6), 936-946.

[8] Roe, P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. Journal of Computational Physics, 1981, 43(2), 357-372.

[9] Demmel, J.W. APPLIED NUMERICAL LINEAR ALGEBRA. SIAM ISBN 0-89871-389-7, 1997.

[10] Markou, G.A., Mouroutis, Z.S., Charmpis, D.C. and Papadrakakis, M. The ortho-semi-torsional (OST) spring analogy method for 3D mesh moving boundary problems. Computer methods in applied mechanics and engineering, 2007, 196(4-6), 747-765.

[11] Blom, F.J. Considerations on the spring analogy. International Journal for Numerical Methods in Fluids, 2000, 32(6), 647-668.