ARTIFICIAL NEURAL NETWORKS FOR OPTIMAL AIRFOIL DESIGN

R. López, R. Flores, E. Escolano and E. Oñate

International Center for Numerical Methods in Engineering (CIMNE)
Universidad Politécnica de Cataluña
Campus Norte UPC, 08034 Barcelona, Spain
E-mail: rlopez@cimne.upc.edu, rflores@cimne.upc.edu, escolano@cimne.upc.edu and onate@cimne.upc.edu
Web page: http://www.cimne.com

Key words: Neural Networks, Multilayer Perceptron, Optimal Shape Design, Aeronautical Applications

Abstract. *A neural networks approach is proposed in this work for the designing of airfoil profiles with maximum aerodynamic efficiency. A practical application is solved by integrating the software tools Flood, GiD and PUMI for neural networks, pre and postprocessing and computational fluid dynamics, respectively.*

1 INTRODUCTION

The aerodynamic efficiency is the amount of lift generated by a wing or vehicle compared to the drag it creates by moving through the air [1]. It can therefore be defined as the C_l / C_d ratio, where C_l and C_d are the lift and drag coefficients, respectively.

Improvement of the efficiency in one of the major goals in wing design. Indeed, since a given aircraft's needed lift does not change, delivering that lift with lower drag leads directly to better fuel economy, climb performance and glide ratio.

In order to obtain high efficiency it is essential to design airfoils using shape optimization techniques [2]. The use of neural networks within a variational formulation provides a direct method for the solution of optimal shape design problems [3].

Neural networks have been widely applied in aeronautics. For instance, an aerodynamic design procedure that incorporates that computational tool is described in [4]. However, the approach here is different since an extended class of neural network is used to define the airfoil shape, and not a response surface for the objective function.

The method is applied to the designing of the shape and the fligth conditions of a transonic airfoil. The results demonstrate that the approach is very useful for optimization, since the constucted neural network is able to represent an extensive family of airfoils for any flight condition.

2 PROBLEM STATEMENT

An airfoil can be generated using analytical equations that describe the camber (curvature) of the mean-line (geometric centerline) of the airfoil section, $y_c(x)$, as well as the section's thickness distribution along the length of the airfoil, $y_t(x)$, for $x \in [0,1]$. This will provide us with the final coordinates for the airfoil upper surface (x_U, y_U) and lower surface (x_L, y_L) [1].

A suitable airfoil must hold some conditions. The camber distribution must be zero at both borders of the airfoil, $y_c(0) = 0$ and $y_c(1) = 0$. On the other side, at the left border, the thickness must be zero, $y_t(0) = 0$, and its derivative must be infinite, $y_t'(0) = \infty$. At the position of maximum thickness p, the thickness distribution must take the maximum thickness value m, $y_t(p) = m$, and the derivative here must be zero, $y_t'(p) = 0$. Finally, at the right border, the thickness must be zero, $y_t(1) = 0$, and its derivative can not be infinite, $y_t(1) \neq \infty$.

The aim is to determine the shape and the angle of attack, α, of an airfoil with given thickness and under transonic flight conditions, providing maximum aerodynamic efficiency. The maximum thickness is here set to $m = 0.15$, and the free-stream Mach number to $M_\infty = 0.85$. The flow is assumed to be Eulerian [5].

3 NUMERICAL RESULTS

The optimal shape design problem stated in the preceding Section is approached in this one using a variational formulation for an extended class of multilayer perceptron [3] [6].

3.1 Selection of function space

A neural network with a sigmoid hidden layer and a linear output layer is used here to represent the camber and the thickness distributions [3]. It must have one input, x, and two outputs, y_c and y_t. Three neurons are set in the hidden layer. Figure 1 is a graphical representation of this network architecture.

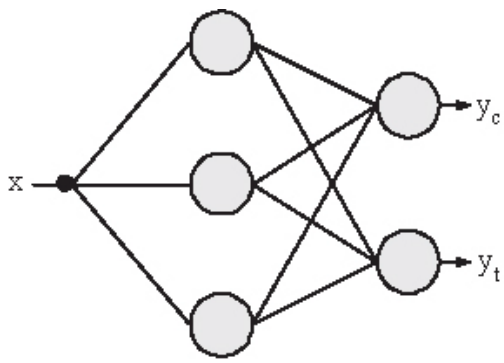

Figure 1. Network architecture representing the airfoil.

On the other hand, this multilayer perceptron must be extended with two independent parameters, the position of maximum thickness, p, and the angle of attack, α. The total set of free parameters is thus $\underline{\beta} = (\underline{\omega}, p, \alpha)$, where $\underline{\omega}$ is the vector of biases and synaptic weights

in the multilayer perceptron [6].

Finally, the camber must hold the conditions $y_c(0)=0$ and $y_c(1)=0$. Also, the thickness needs to satisfy $y_t(0)=0$, $y_t'(0)=\infty$, $y_t(p)=m$, $y_t'(p)=0$, $y_t(1)=0$ and $y_t'(1)\neq\infty$. For that purpose, a suitable set of particular and homogeneous solutions for the camber and the thickness, $\varphi_{0c}(x)$, $\varphi_{0t}(x)$, $\varphi_{1c}(x)$ and $\varphi_{1t}(x)$, is chosen [6].

Such a neural network spans a family V of parameterized functions $(y_c, y_t)(x;\underline{\beta})$ of dimension $s=14+2$, being 14 the number of biases and synaptic weights in the neural network and 2 the number of independent parameters.

Multilayer perceptron neural networks are a class of universal approximators [7]. In this way, the function space constructed in this Section can be considered a very suitable family of airfoils.

3.2 Formulation of variational problem

The statement of this problem is then to find a function $(y_c, y_t)^*(x;\underline{\beta}^*)$ for which the functional $F\left[(y_c, y_t)(x;\underline{\beta})\right]=C_l/C_d$, defined on V, takes on a maximum value.

Evaluation of the objective functional is performed in three steps. First, an airfoil is proposed by the neural network using Flood [8]. Second, a GiD batch file is automatically written to create the geometry, apply the boundary conditions, generate the mesh and write the fluid solver input file. Third, the lift and drag coefficients are calculated by solving the Euler equations using PUMI [5]. Figure 2 shows the activity diagram for the evaluation of the objective functional.

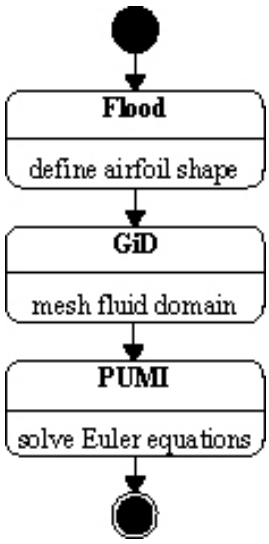

Figure 2. Activity diagram for evaluating the objective functional.

3.3 Solution of reduced function optimization problem

An evolutionary algorithm with linear ranking fitness assignment, stochastic universal sampling selection, intermediate recombination and normal mutation is applied for training the neural network [8]. The dimension of each individual is 16, the population size is set to 160 and the number of generations is set to 100. The total number of evaluations performed is thus 16000.

The upper and lower coordinates of the most efficient airfoil found by the neural network

are plotted in Figure 3, together with the corresponding pressure coefficient colour map and velocity vector field. These have resulted in a supercritical airfoil with $C_l / C_d = 4.894$. The best performance is for an angle of attack α=3.890º .

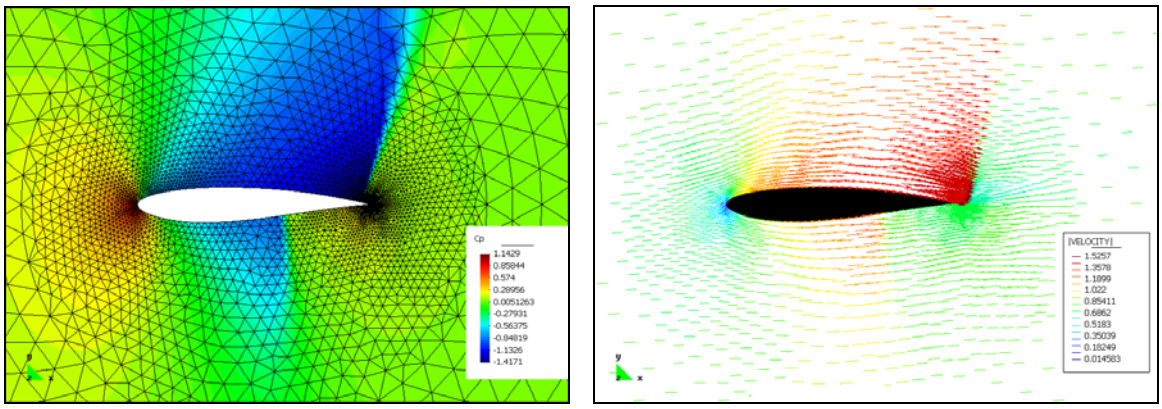

Figure 3. Pressure and velocity fieds for the optimal airfoil design, with $\alpha = 3.890^{o}$.

4 CONCLUSIONS

A multilayer perceptron extended with independent parameters and boundary conditions is able to represent a very complete family of airfoils. This neural network has been used here to find an optimal design with given thickness and under for transonic flight conditions, which has resulted in a supercritical airfoil.

Future work relies on distributing the computation over a cluster, in order to decrease computational time and to achieve better training results.

REFERENCES

[1] H. Ashley, *Engineering Analysis of Flight Vehicles*, Dover Publishing (1992).
[2] D. Bucur and G. Buttazzo, *Variational Methods in Shape Optimization Problems*, Birkhauser (2005).
[3] R. Lopez, E. Balsa-Canto and E. Oñate, “Neural networks for variational problems in engineering”, *International Journal for Numerical Methods in Engineering*, in press (2008).
[4] M.M. Rai and N.K. Madavan, “Aerodynamic design using neural networks”, *AIAA Journal*, 38(1), 173-182 (2000).
[5] E. Ortega, R. Flores and E. Oñate, “An edge-based solver for compressible flow”, *CIMNE* (2005).
[6] R. Lopez and E. Oñate, “An extended class of multilayer perceptron”, *Neurocomputing*, in press (2008).
[7] K. Hornik, M. Stinchcombe and H. White, “Multilayer feedforward networks are universal approximators”, *Neural Networks*, 2(5), 359-366 (1989).
[8] Flood: An open source neural networks C++ library, www.cimne.com/flood (2008).
[9] GiD – The personal pre and postprocessor, www.gidhome.com (2008).