World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:3, 2012

Computational Fluid Dynamics Expert System using Artificial Neural Networks

Gonzalo Rubio[1]*, Eusebio Valero[1] and Sven Lanzan[2]
[1]*School of Aeronautics, Polytechnic University of Madrid, Madrid, Spain, 28040*
[2]*AIRBUS Operations S.L., Getafe, Spain, 28906*
**g.rubio@upm.es*

Abstract—The design of a modern aircraft is based on three pillars: theoretical results, experimental test and computational simulations. As a results of this, Computational Fluid Dynamic (CFD) solvers are widely used in the aeronautical field. These solvers require the correct selection of many parameters in order to obtain successful results. Besides, the computational time spent in the simulation depends on the proper choice of these parameters.

In this paper we create an expert system capable of making an accurate prediction of the number of iterations and time required for the convergence of a computational fluid dynamic (CFD) solver. Artificial neural network (ANN) has been used to design the expert system. It is shown that the developed expert system is capable of making an accurate prediction the number of iterations and time required for the convergence of a CFD solver.

Keywords—Artificial Neural Network, Computational Fluid Dynamics, Optimization

I. Introduction

DUE to the highly non-linear nature of the Navier-Stokes equations, analytical solutions for real problems are not available [10]. Hence, computational fluid dynamics (CFD) solvers are used to obtain numerical results. These solvers require a high level of user's knowledge in order to optimize their performance. Moreover, the calculations involved are expensive, as far as computational time is concerned, i.e., a non-expert user employs long times trying different program's parameters in order to achieve the required results. Afterwards, when this user becomes an expert, his own experience will be used to select the correct CFD parameters. On the other hand, when a new aircraft is designed, a lot of CFD calculations are carried out to estimate the aircraft performances. These calculations involve a lot of CPU computational time, which is projected according only to the experience. The moral of the story is that important decisions are taken using the gained experience.

Every CFD calculation performed in a design process is recorded in a database. Therefore all the experience that the manager or the user need is stored there. A mathematical algorithm can be used in order to extract this information; afterwards it can be used to create an expert system. Usefulness of artificial neural networks (ANN) in modeling unknown complex nonlinear systems has been broadly proved [13]. Therefore in recent years, problems in the aeronautic field such as: modeling high speed turbulent boundary layer inducing optical distortions [1], optimization of low Reynolds number airfoils [2] and restricted flight dynamic models for aircraft parameters estimation [3] have been solved using these techniques. Consequently our expert system will be based on ANN technology.

In this work, TAU solver has been used to run the simulations. TAU is one of the most used CFD solvers used in Airbus. TAU originates from the MEGAFLOW project and is constantly developed by the German Aerospace Center DLR. More information about TAU could be obtained in [6].

In this paper we present a novel use of ANN which is the creation of an expert system for the CFD software TAU. In section II a brief description of artificial neural networks is presented. The database used to feed the expert system is depicted in section III. The expert system is described in section IV. The results are exposed in section V. Finally conclusions and future directions are shown in section VI.

II. Brief description of artificial neural networks

Artificial neural networks (ANN) are very sophisticated modelling techniques inspired in the brain. A biological neural network is a group of highly interconnected parallel processing units called neurons. Consequently the mathematical model for the neuron and the interconnections defines the ANN [7]. There are many different types of neural networks. The choice of the type of network depends on the nature of the problem to be solved [13]. The most used ANN for problems with a database of run cases is the MLP. In the MLP the neuron model is called perceptron, and the interconnections structure, layered feedforward (see Fig. 1). The MLP shown in Fig. 1 has one output. Although it is possible to use MLPs with more than one output, one unique output is suggested in order to avoid crosstalk [11]. Crosstalk produces a deterioration of the solution caused by the noise one output introduces in the others.

As far as the connections are concerned, it should be noticed that not all the connections have the same power. Each connection is weighted (similar to synapses is biological systems). These weights act like free parameters and varying them any function can be approximated.

The MLP learns by example, therefore, in order to train the MLP, a database is required. The database is constituted by a set of examples (inputs-outputs) of the performance the MLP is supposed to learn. The learning process is carried out using the training algorithm. This algorithm alters the MLP weights

TABLE I
PARAMETERS OF THE MLPS CREATED WITH THE SOFTWARE FLOOD 2

Transfer function	Hyperbolic tangent
Objective functional	Sum of squared errors
Training algorithm	Quasi-Newton method
Pre-training method	Evolutionary algorithm
Pre/post-processing method	Mean and standard deviation

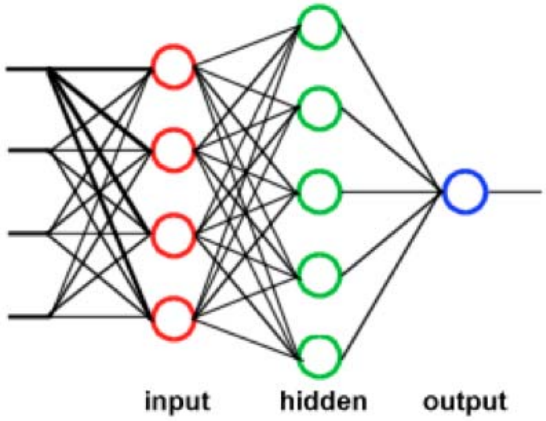

Fig. 1. **Scheme of a MLP with the perceptrons and its interconnections.** *Each circle is a perceptron and the lines represent the connections. It can be seen that the information is propagated from left to right, from one layer to the next. The first layer is called input layer and it is the one in charge of receiving the input information to the MLP. The last one is called output layer and is the one which generates the output of the MLP. Each layer between these is called hidden layer.*

in order to minimize an error (See Fig. 2). The error is defined as the difference of the MLP output (when the inputs of the database are computed) and the outputs of the database.

The number of hidden layers and the number of perceptrons in each hidden layer, define the complexity of the function the MLP can approximate. Moreover, it's known that the multilayer perceptron with at least one hidden layer is a class of universal approximator [8]. Any function defined between two finite-dimensional spaces could be approximate to any grade of accuracy, provided enough hidden neurons are available. The optimal number of hidden layers and neurons can be obtained using cross validation techniques [12]. These techniques are based on dividing the set of examples available into subsets, in order to use some for training and the rest for validating the MLP.

More information about ANNs and MLPs could be found in [4], [7].

In order to perform all the calculations, FLOOD 2, an open source library, has been used [9]. This software provides an easy way to create and train MLPs. The technical details of the MLPs used can be seen in Table I. The hidden layer's number and the neuron's number are different in each MLP depending on the complexity of the problem. More information about these details could be found in FLOOD 2 userguide [11].

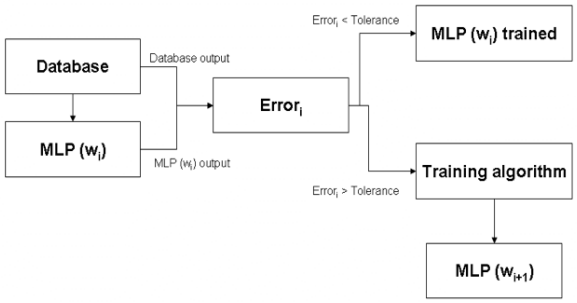

Fig. 2. **Adjusting of weights of a MLP. Scheme of the learning process for a MLP.** *The database is the set of examples used to train the MLP.*

III. DATABASE

In the previous section, the working of a MLP has been explained. It has been established that the MLP learns by example, so in order to teach the MLP, a database with examples is needed. Moreover, it is impossible to obtain a good trained MLP without a high-quality database. The desirable features of a good database are [5]:

-Completeness: the phenomenon the MLP is supposed to learn must be depicted by the variables in the database.
-Uniformity: a good database has to be well distributed so every example is equally represented.
-Size: the required minimum size of the database increases proportionally to the complexity of the problem.

In order to create our database, first of all it is important to remember the phenomenon the MLP is supposed to learn: CFD computation time and number of iterations for convergence. A real CFD database previously generated was used to train the MLP. The variables present in the CFD database are shown in Table II. It can be noticed that most of the parameters which affect the desired output are present in the database, so the completeness is fulfilled. Besides, there are no size problems. However, it lacks of uniformity. There are special cases poorly represented ($\beta \neq 0$, cases without multigrid, Mach$\sim$0.5, central inviscid flux discretization...). In order to fix this fault the database has been filtered. The specific filter used is different in each particular case and is detailed in what follows. Finally, the database is divided before using it to train the MLP in order to use cross validation techniques (see Section II).

IV. EXPERT SYSTEM

An expert system to estimate the computation time and number of iterations required for the convergence of the CFD simulation has been developed. The expert system uses ANN technology. The tool can provide very useful information: on the one hand, it can estimate in advance the required time and number of iterations for convergence. On the other hand, it can provide recommendations of the values of the parameters that improve the CFD solver, in terms of time consuming and iterations number for convergence. A scheme of the expert system is shown in Fig. 3.

TABLE II
DESCRIPTION OF THE VARIABLES PRESENT IN THE DATABASE. THE VARIABLES ARE TAU PARAMETERS

Reynolds	Reynolds number
Mach	Mach number
CFL	Courant-Friedrichs-Lewy number
β	Angle of Yaw
α	Angle of attack
IFDT	Inviscid Flux Discretization Type
Multigrid	Multigrid scheme used
Turbulence Model	Turbulence model used
Single grid iterations	Number of iterations using single grid
C_L	Lift coefficient
C_D	Drag coefficient
Time	Computational time
Processors' number	Number of processors used for the calculation
Iterations	Number of iterations
Mesh size	Mesh's size (Gb)
HPC	High Performance Machine where the solver runs

The expert system is constituted by four MLPs. The reasons of using four MLPs instead of only one are: on the one hand, the crosstalk which makes a neural network works better if it only has one unique output [11]. Our expert system has three outputs (iterations convergence for CL and CD and time) therefore three MLPs are the minimum necessary to avoid crosstalk. On the other hand, the lack of uniformity of the database makes an extra MLP very useful to improve the performance of the expert system. This extra MLP has a filtering purpose (see section III).

The first MLP is created to divide the database between two possible scenarios: convergence with enough number of iterations and no convergence. The set of examples used to extract this performance is filtered from the database to $\beta = 0$, upwind inviscid flux discretization, Multigrid=3w. A scheme with the first MLP is shown in Fig. 4.

The purpose of the second and third MLPs is discerning if a case is converged with a specific number of iterations. The second MLP is CL (partial) convergence classifier. The third MLP is CL & CD (total) convergence classifier. The set of examples used is filtered to $\beta = 0$, upwind inviscid flux discretization, Multigrid=3w, and positive final convergence. A scheme with CL (partial) convergence MLP is shown in Fig. 5 and with CL & CD (total) convergence in Fig. 6.

The number of iterations for convergence is an output in the expert system (see Fig.3), though for the MLPs trained it is an input (see Fig.5 and Fig.6). In the expert system for each group of inputs (Mach, CFL and α) different number of iterations are computed and the number of iterations for convergence is displayed.

The fourth MLP is trained to predict the time for reaching convergence. The most important filter performed in this case is the HPC where the solver runs. Only the cases run in Marignan's HPC were considered. This is one of the most important Airbus' High Performance Machines. There are also filters for Multigrid=3w, upwind inviscid flux discretization, Turbulence model=2 and Processors' number

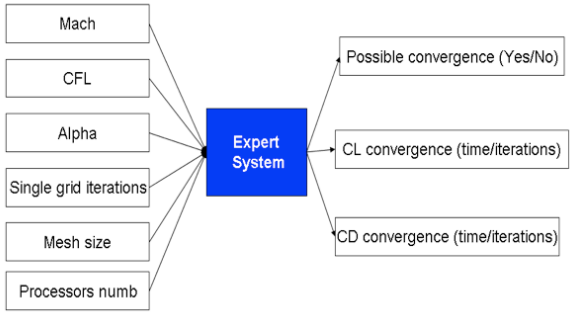

Fig. 3. **Scheme of the expert system.** *The inputs of the expert system can be seen at left side of the figure. On the right side we have the outputs.*

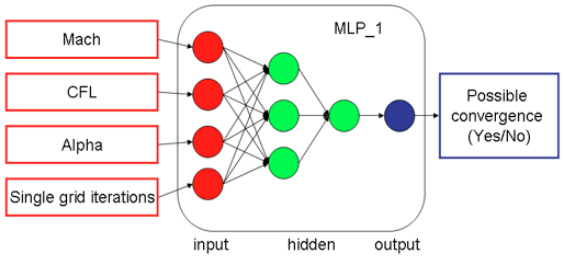

Fig. 4. **Scheme of the expert system.** *The inputs of the expert system can be seen at left side of the figure. On the right side we have the outputs.*

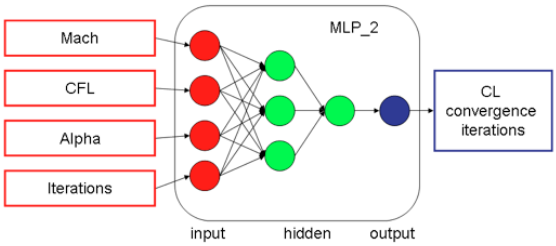

Fig. 5. **Scheme of the expert system.** *The inputs of the expert system can be seen at left side of the figure. On the right side we have the outputs.*

of 32, 64 and 128. A scheme with time for convergence MLP is shown is Fig.7.

Each MLP used has its optimum architecture. It has been found using cross validation [12]. This method consists on split the database in three. The first part is used to train the MLP. The second one is used to evaluate the trained MLP. The third one is reserved to obtain the Expert System results exposed in Section V.

V. RESULTS

In this section, the results obtained using the expert system developed are exposed. These results have been obtained using cross validation [12]. As it was explained in section IV the database was split in three sets. The first two thirds were used in the training stage. The third one was reserved, as

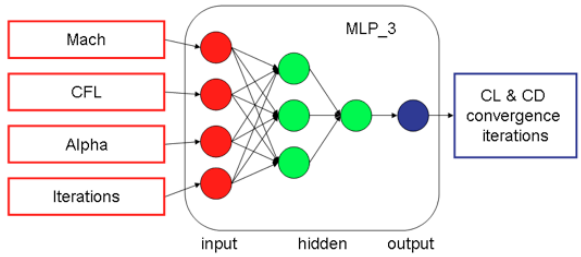

Fig. 6. **Scheme of the expert system.** *The inputs of the expert system can be seen at left side of the figure. On the right side we have the outputs.*

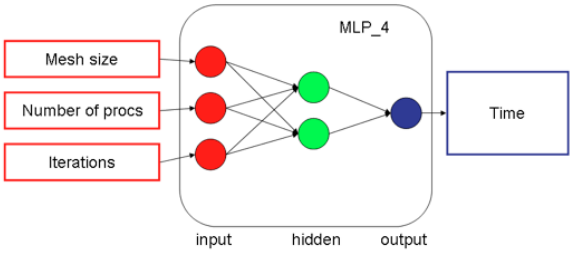

Fig. 7. **Scheme of the expert system.** *The inputs of the expert system can be seen at left side of the figure. On the right side we have the outputs.*

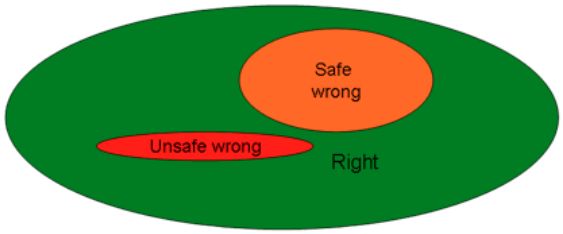

Fig. 8. **Scheme of the expert system.** *The inputs of the expert system can be seen at left side of the figure. On the right side we have the outputs.*

an independent data set, to check the accuracy of the expert system results. When the MLP is used for classification the next agreement is going to be used:

Right result: the result is correct.

Safe wrong result: the result is incorrect, however is safe. The expert system result is of non-convergence when the right answer is convergence.

Unsafe wrong result: the result is incorrect and unsafe. The expert system result is of convergence when the right answer is non-convergence.

Not wrong result: is the sum of right results and safe wrong results.

Wrong result: the result is incorrect. It is the sum of safe wrong results and unsafe wrong results.

A. First MLP - Possible convergence [Yes/No]

Following the agreement explained before, the next results for the first MLP can be presented:

Right / Total (%) = 71%
Not Wrong / Total (%) = 81%

This error was not as good as expected. However, the iterations prediction (second and third MLP) will be accurate where it counts (the cases with final convergence). The result would be better if information about the mesh's geometry, e.g. High lift configuration, was available in the database.

B. Second MLP - CL convergence

Right / Total (%) = 80%
Not Wrong / Total (%) = 99%

C. Third MLP - CL & CD convergence

Right / Total (%) = 78%
Not Wrong / Total (%) = 94%

The importance of the Not Wrong/Total error must be noticed. The expert system is designed in order to have a safe behavior. If any doubt about the result is harbored the answer is of non convergence. However a positive convergence result is always obtained by adding some iterations.

D. Fourth MLP - Time

We have created an error distribution with the independent data set. In order to check if the error distribution was normal, the next statistical analyses were performed: Shapiro-Wilk, Chi2, Kolmogorov-Smirnov, Kuiper V, Cramer-Von Mises W2, Watson U2, Anderson-Darling A2. The conclusion of these analyses was that the distribution is normal, with media and standard deviation of:

Media: -0,012
Standard deviation: 0,1251

With this information, we can assure we can predict time with the next accuracy:

Error $<$ 12,68 % with 68,27 % of confidence (1σ)
Error $<$ 25,36 % with 95,44 % of confidence (2σ)

VI. Conclusions and Future Directions

In this study, we have applied the artificial neural network technology to an interesting problem from industry's point of view. Specifically, we have shown that the MLP constitutes a perfectly valid analysis technique for predicting CFD convergence time and number of iterations. In order to obtain the expert system, an industrial CFD database was used. A MLP combination was needed to solve the drawbacks of the database. Finally the expert system results have been analyzed.

There are some aspects which could be improved in order to achieve better results. First of all the entire database used in this work was highly non uniform. A database specifically created for the expert system would improve the results; e.g. if information about the aircraft configuration was included a

better final convergence prediction could be done. Secondly the expert system could be more ambitious and auto optimize the TAU parameters in order to obtain minimum iterations number. Finally newer training algorithms instead of Quasi-Newton method for MLP training could be used in order to get better results.

REFERENCES

[1] L. Wu, J. Fang, Z. Yang, S. Wu, *Study on a neural network model for high speed turbulent boundary layer inducing optical distortions*, *Optik - International Journal for Light and Electron Optics*, Vol. 122, No. 17, pp. 1572 - 1575, 2011.

[2] M. Bellman, J. Straccia, B. Morgan, K. Maschmeyer, R. Agarwal, *Improving Genetic Algorithm Efficiency with an Artificial Neural Network for Optimization of Low Reynolds Number Airfoils*, *AIAA 2009-1096, 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition 5-8 January 2009, Orlando, Florida*

[3] N.K. Peyada, A.K. Ghosh, *Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method*, *The Aeronautical Journal, April 2009, Volume 113, No 1142 243*

[4] T. Hill, P. Lewicki, *Statistics: methods and applications: a comprehensive reference for science, industry and data mining*, *The Aeronautical Journal, April 2009, Volume 113, No 1142 243*

[5] K.-K. Sung, *Learning and Example Selection for Object and Pattern Detection*, *MIT AI Lab, Jan. 1996*

[6] T. Gerhold, V. Hannemann, D. Schwamborn, *On the Validation of the DLR-TAU Code*, New Results in Numerical and Experimental Fluid Mechanics, Notes on Numerical Fluid Mechanics, 72, pp. 426-433, 1999

[7] R. López, *Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method*, *The Aeronautical Journal, April 2009, Volume 113, No 1142 243*

[8] K. Hornik,M. Stinchcombe, H. White, *Multilayer feedforward networks are universal approximators*, *Neural Newtorks, 2(5):359-366, 1989*

[9] R. López,(2010) Flood: An Open Source Neural Networks C++ Library (Version 2) [software]

[10] C. Fefferman, *Existence and smoothness of the Navier-Stokes equation*, *Clay Millenium Problems 2000*

[11] H-C. Fu, Y-P. Lee, C-C. Chiang, H-T. Pao, *Divide-and-Conquer Learning and Modular Perceptron Networks*, *IEEE transactions on neural networks, Vol 12, No 2, March 2001*

[12] G. Zhang, M. Hu, B. Patuwo, D. Indro, *Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis*, *European Journal of Operational Research 116(1999) 16±32*

[13] S. Lek, J.F. Guégan, *Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method*, *The Aeronautical Journal, April 2009, Volume 113, No 1142 243*

[14] C.M. Bishop, *Neural Networks for Pattern Recognition*, *Oxford: Oxford University Press*