The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-–9 June 2022, Oslo, Norway

A MACHINE LEARNING BASED EXPERT SYSTEM FOR OPTIMISING CFD SOLVER PARAMETERS

LINA EL ZAATARI, TOBIAS LEICHT, STEFAN LANGER, PHILIPP BEKEMEYER, STEFAN GÖRTZ

German Aerospace Center (DLR)
Institute of Aerodynamics and Flow Technology, C²A²S²E
Lilienthalplatz 7, 38108 Braunschweig, Germany,
lina.elzaatari@dlr.de, tobias.leicht@dlr.de, stefan.langer@dlr.de, philipp.bekemeyer@dlr.de,
stefan.goertz@dlr.de

Key words: Computational Fluid Dynamics, Optimization, Machine Learning, Expert System

Abstract. **C**omputational **F**luid **D**ynamics is a viable tool in the field of aerodynamics enabling to reduce time, effort and budget required for experimental testing. Although powerful and established for various years, it remains a complex tool calling for experienced users to ensure consistent high-quality results. This complexity primarily stems from the underlying model, namely the Navier-Stokes equations typically combined with a set of equations resolving the effects of turbulence. Additionally, to obtain accurate high-fidelity result appropriate meshes are required. As a consequence, a substantial number of parameters needs to be selected carefully and the quality of a result often highly depends on individual knowledge and experience of a user. Hence, a strong desire exists to reduce the number of input parameters without causing a loss of accuracy and efficiency. Such reduction of parameters might be viewed as a prerequisite to CFD as a tool in process chains for multidisciplinary applications where typically no user interaction is possible. In this article we propose a machine-learned Expert System for CFD to provide guidance for users in selecting optimal or at least near optimal parameter combinations. The proposed Expert System is divided into two macro steps, the surrogate model and a genetic algorithm to determine from the surrogate model the parameters. Numerical examples are presented to demonstrate the approach.

1 Introduction

Even after decades of development, the automatic and reliable application of Computational Fluid Dynamics (CFD) to simulate flows around complex objects is still considered a challenge. Nevertheless, CFD has been successfully used for many configurations and results of great accuracy have been achieved [1, 2]. When looking at the entire process chain including CAD modeling, meshing as well as the selection of various parameters such as parameters influencing the spatial and temporal discretization schemes [3, 4] and solution process [5, 6], it becomes evident that the amount of required user interaction is high. Hence, individual knowledge of a user plays a vital role in ensuring accurate and reliable results. Nevertheless, there is the undeniable industrial requirement that such methods can be used in process chains with as little user interaction as possible to minimize the risk of failures while simultaneously reducing turnaround-

times. This ultimately yields the question if it's possible to systematically decrease the amount of user interaction required without losing accuracy and ideally also boosting reliability.

Here the topic of Machine Learning (ML) comes into play which had a significant impact in various fields (e.g. medical sector, stock market, banking system, speech recognition, image classification, weather forecast etc...) within the past few years. An Expert System is defined as a system that is capable of executing tasks that are generally inherent to human intelligence, by using a database of previous knowledge [7]. The so-called expert system combines AI with a database of CFD simulations to tackle the aforementioned shortcomings. This way of tackling the problem at hand combines the advantages of both AI and the conventional systems [7]. The idea of creating such an Expert System is not new; some research papers date back to the early 80s [7, 8, 9, 10]. Although the idea of exploiting Artificial Intelligence is very interesting, it lacked a fundamental aspect, which is adequate processing power. New interest emerged as the era of artificial intelligence started to become prominent. More precisely, Machine Learning is being used to enhance performance of CFD simulations in a wide range of applications. Research was and is being conducted on the benefit of exploiting machine learning to produce reduced order models which aim at increasing the speed and efficiency of simulations, developing high-fidelity turbulence models, or modeling fluid mechanics related heat transfer[11].

In this paper, we propose an Expert System that has the capability of optimizing CODA [12, 13] variables to ensure efficient, robust and accurate results without user intervention. The idea of the proposed approach can be broken down in two steps. First, previous user experience, e.g. results from various simulations, are concatenated in a database that includes a representative number of test cases created with a CFD solver that solves the underlying equations of interest. The database includes as many input parameters as possible that have to be defined from a user and which influence the simulations outcome. This includes in particular all parameters that influence the accuracy of the result and the solver behavior, but is not limited to these. Afterwards, this database forms the backbone and is used to train a surrogate model. Note, the quality and meaningfulness of our surrogate model increases with every calculation that is entered into the database. This also includes parameters and results that did not meet the require level of accuracy and/or reliability. Hence, the surrogate model is also able to learn from failures and does not simply relies on success. In the second step, the surrogate model is exploited to predict optimal parameters for previous untried simulation cases utilizing global optimization algorithms. For this purpose, we may apply suitable algorithms to this function to produce a set of appropriate parameters for a new, unknown case. In principal this approach is fully automated. Moreover, new test cases can be added to the database (after finishing the simulation), automatically increasing the information density. Even if the determined parameters are not "optimal", they might still be better compared to an initial guess of an inexperienced user.

This article is organized as follows: In Section 2 the applied methodology is explained. The underlying CFD code is introduced briefly and the parameters of interest are classified. Section 2.3 is devoted to the numerical algorithms including applied surrogate model and optimization algorithms. A numerical example is presented in Section 3 showcasing the proposed approach. The article closes with a conclusion in Section 4 also shedding some light on next steps.

2 Methodology

In this section, a general introduction to every aspect of the expert system is presented. Figure 1 shows the expert system in a more detailed way. As mentioned before, the expert system will be divided in two macro steps; furthermore, each macro step will contain one or more sub-steps that will be outlined in more detail in the following.

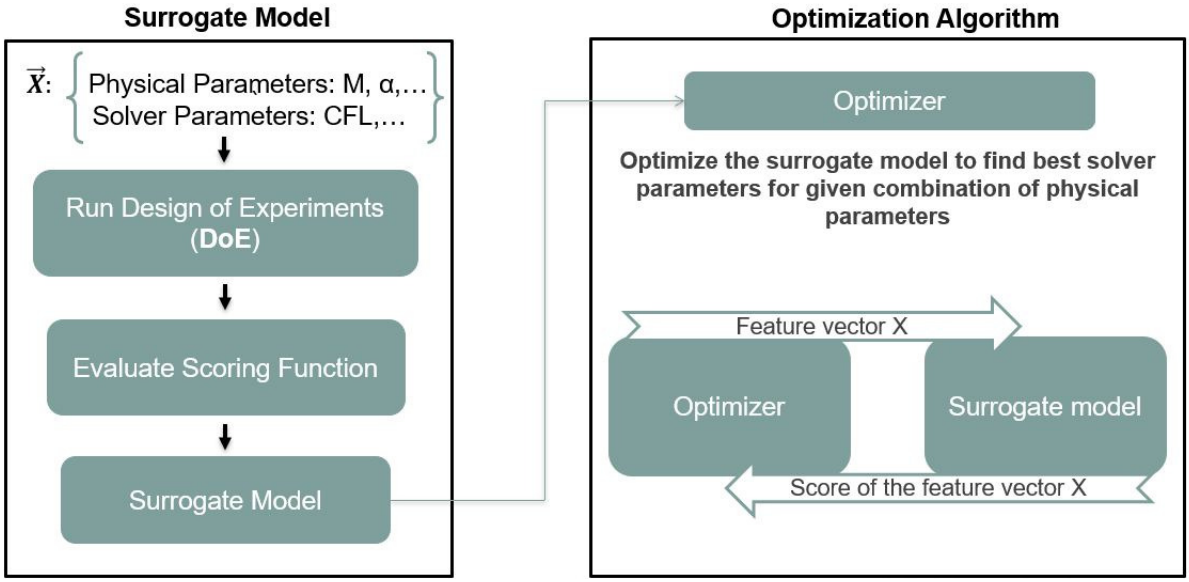


Figure 1: Expert System Breakdown

2.1 CFD Code and governing equations

For a practical implementation a prerequisite is to set up a database and to decide for a selection of parameters. For this purpose, as a code to set up a database, we chose CODA, the CFD solver that is jointly developed by Airbus, ONERA and DLR [12, 13]. It is a tool which solves the Reynolds-averaged Navier-Stokes equations in integral form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\mathbf{W}\,\mathrm{d}\mathbf{x}+\int_{\partial\Omega}\left(\mathbf{f}_c\cdot\mathbf{n}-\mathbf{f}_v\cdot\mathbf{n}\right)\,\mathrm{d}s=0,\qquad \Omega\subset\mathbb{R}^3, \tag{1}$$

where $\mathbf{W}:=(\rho,\rho u_1,\rho u_2,\rho u_3,\rho E)$ are the conservative variables representing density, momentum and energy, Ω is the domain of interest, typically an airfoil or an aircraft and $\mathbf{n}=(n_1,n_2,n_3)$ denotes the outer unit normal vector.

2.2 Definition parameters for database

When looking at the multitude of different parameters affecting a numerical simulation, one can group them into three different categories:

1) Physical input parameters (Passive Parameters), for example geometry, Mach number, angle of attack and Reynolds number,

2) Solution algorithm parameters (Active Parameters), for example CFL number,

3) Parameters influencing the solution accuracy (Active Parameters), such as choice of the turbulence model and entropy fix.

The emphasis on passive and active parameters is important since only active parameters are going to be optimized, while the passive parameters are defined by the application of interest and are therefore given from a user. To create a comprehensive database, for many test cases calculations based on a wide variety of these parameters need to be performed and stored. In this article we limit the active parameters involved to the determination of the time step, that is a set of parameters which influence the solution algorithm (type 2 in the list above). Passive parameters and active parameters from the third category are omitted herein to keep the demonstration as simple as possible. Note that, the proposed methodology is valid if all parameters are involved. An overview of the parameters which are accounted for in the preliminary study is given in Figure 2. As one notices, the determination of the time step in the CODA code involves 10 parameters which can be combined in different ways. This heuristic view already sheds a glimpse at the complexity of an Expert System.

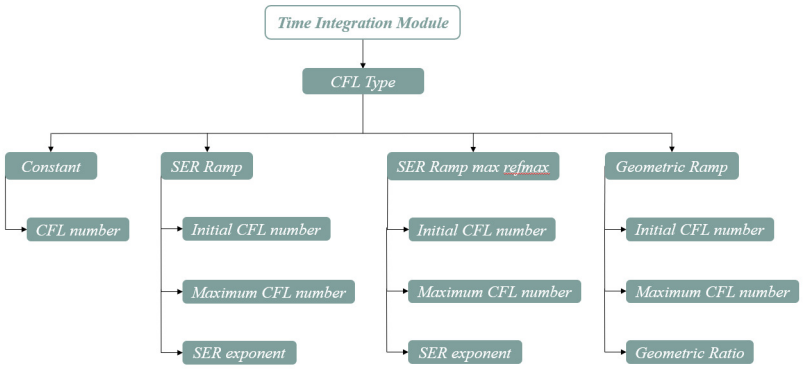


Figure 2: CFL Ramping Strategies

2.3 Surrogate model

If one assumes a large number of input and output parameters and additionally a large number of test cases, a direct search inside the database may be very inefficient and most certainly will not lead to an optimal results because the specific set of passive parameters might not even be included in the database. Hence, a further important step to realize the proposed Expert System for CFD is the introduction of a suitable surrogate model. In general, surrogate models try to predict the behaviour of complex systems based on a selected set of sample points. In our application, every sample point from the database represents a single simulation. A well-suited and trained model allows us to mimic the correlation between input and output data and enables an efficient search through the database providing solutions for varying inputs. To design a surrogate model several different methods can be considered. It's important to note

that the algorithm to determine the parameters later on highly dependent on the performance of the underlying model. In the follow we will introduce two different methodologies that are applied herein for the surrogate modeling task.

2.3.1 Nearest Neighbor Search

Perhaps the most obvious idea is, to use the database entries themselves as a surrogate model. On the one hand, this has the advantage that we do not have to construct a surrogate model. On the other hand, we only have information in the database for given discrete data. The nearest neighbour [14] as the name implies employs the information given from data points that are in close proximity of the point of interest. The user chooses the number of neighbours (K) that the algorithm needs to take into consideration and it will then predict the value of the dependent variable by averaging the neighbor information. Obviously, this approach is only successful if an entry already exists that is close to the given new test case. Note that, this approach can be seen as a formalization of the typical engineering set-up in which an experienced user spreads his knowledge to other users based on simulations he has previously performed.

2.3.2 Regression models

Regression models are a common choice as surrogates yielding a continuous representation of the database. This model can afterwards be used to predict parameters for any given set of parameters of interest. Throughout the years a large number of different regression models have been proposed and applied. Polynomial interpolation can be used to obtain such a model based on a multidimensional polynomials. Depending on the size of the database and the data, this process can be very expensive and the polynomials tend to introduce oscillations. These oscillations could lead to problems when determining a new suitable set of parameters for new test cases. Instead of polynomial interpolation, we may consider interpolation using splines. The thin plate spline (TPS) is a spline-based technique for data interpolation and smoothing, first introduced by Jean Duchon [15]. The analogy is with a thin metal plate that is warped to fit the data. This can range from a very rigid surface to a very flexible one perfectly fitting every observation [16]. A further approach to realize a surrogate model is to apply ideas from the context of artificial intelligence such as deep neural networks. In this work the thin plate spline algorithm is applied based on the implementation within the DLR in-house code SMARTy (Surrogate Modeling for AeRo data Toolbox Python Package), a toolbox for various data-driven tasks such as predictive modeling of scalar or high-dimensional quantities [17].

2.4 Scoring Function

To obtain a surrogate model, we need to define the independent and dependent variables. The independent variables are all the parameters we provide to the surrogate model (passive and active parameters). Here we decided to use the parameters determining the CFL number to be our independent input. The dependent variable is the output of the surrogate model (also called scoring function), and this is defined by the user depending on specific needs. In our case, we are interested in solver parameters that are able to give us an accurate, robust and efficient solution. Therefore, we defined the dependent variable (scoring function) according to

the following equation:

$$f = \alpha \cdot \log_{10}\left(\overline{C_{L_e}}\right) + \beta \cdot \log_{10}\left(\overline{C_{D_e}}\right) + \gamma \cdot \log_{10}\left(DR\right) + \delta \cdot \log_{10}\left(TR\right) \tag{2}$$

where, $\overline{C_{L_e}}$ is the fluctuation of the lift coefficient C_L, $\overline{C_{D_e}}$ is the fluctuation of the drag coefficient C_D, DR is the normalized density residual norm, TR is the normalized run time of the simulation. All quantities of interest are transformed to the logarithmic scale since the combination leads to a skewed data, so a very efficient way to remove this issue is to use the logarithmic scale. It is important to mention that $\alpha+\beta+\gamma+\delta = 1$ are the weighting coefficients. $\overline{C_{L_e}}$ and $\overline{C_{D_e}}$ are defined as follows:

$$\overline{C_{L_e}} = \frac{1}{0.25 \cdot N} \sum_{i=n}^{N} C_{L_e} \tag{3}$$

where:

$$C_{L_e} = |C_{L_i} - C_{L_{i+1}}| \tag{4}$$

$$\overline{C_{D_e}} = \frac{1}{0.25 \cdot N} \sum_{i=n}^{N} C_{D_e} \tag{5}$$

where:

$$C_{D_e} = |C_{D_i} - C_{D_{i+1}}| \tag{6}$$

where, N is the number of iterations, n is the starting iteration number ($n = \frac{3}{4} * N$), i is the previous iteration and $i + 1$ is the current iteration.

2.5 Determination of new set of parameters

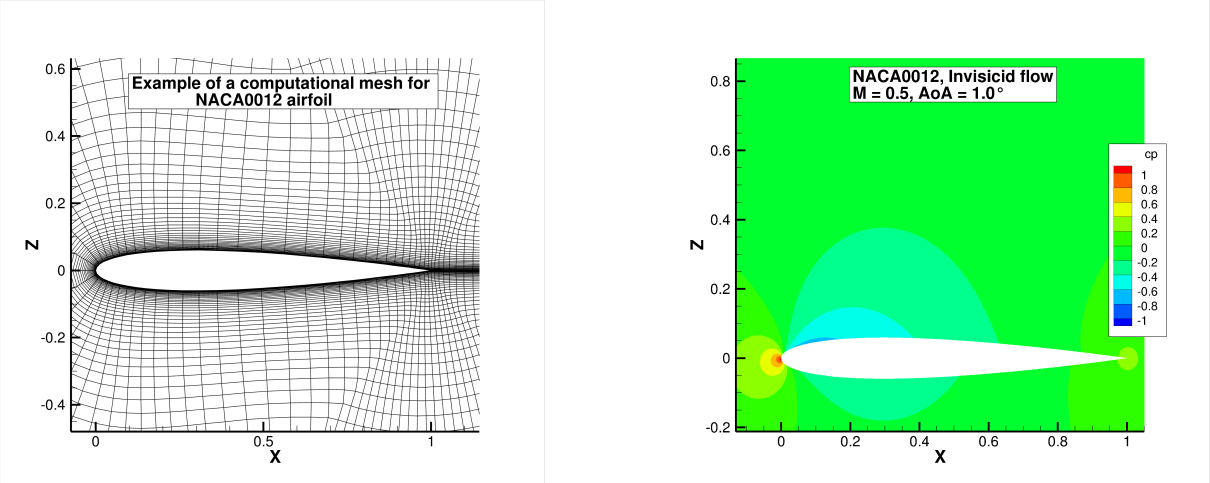
Once the surrogate model is constructed, an algorithm is needed to determine automatically a suited set of parameters for new test cases. Herein an optimization algorithm is employed to scan the design space and find the best solution in terms of minimizing the scoring function. Optimization algorithms are divided in two categories, gradient based approaches and direct approaches. Gradient based optimization algorithms are limited as the function to be optimized has to be locally quadratic, if not the optimizer will converge to a local minimum. Therefore, direct optimization algorithms are preferred, specifically Genetic algorithms (GA) [18], due to their pseudo-stochastic property (random part and deterministic part). The random part explores the design space randomly while the deterministic part ensures that the algorithm moves towards the global optimum. Furthermore, since we don't know the best optimization algorithm for this application a-priori, GA has been selected as a well-established and stable algorithm.

The genetic algorithm will produce a tentative population of solutions in the design space. Subsequently, the surrogate model will evaluate each individual in that population and score it based on the defined scoring function. We thus obtain a population of possible parameters

that are ranked in increasing order of the scoring function. In the end, the genetic algorithm will choose a couple of individual solutions having the lowest scores and then use them as parents for the second generation. At this point, the evolution theory will be used to produce a new population called the offspring, having genes (parameters) from both parents. The loop will keep repeating until the algorithm is not capable of finding better solutions or the predefined budget has been used. In general we cannot expect that there exists a unique solution of the aforementioned minimization problem. Since we only want to figure out a suited set of parameters this is however not strictly necessary. For a realization of this approach we use PyGAD 2.16.3 [19] which is an open source Genetic Algorithm Python library containing different crossover and mutation strategies [19].

3 Test Case 1: Expert System for CFL parameters

As a first, preliminary study, it is our goal to set-up an Expert System capable of determining the CFL parameters for simulations with CODA, see Figure 2. To demonstrate the potential of the proposed Expert System we consider the inviscid flow over the NACA0012 airfoil. The flow conditions are defined as Mach number $M_\infty = 0.5$ and angle of attack $\alpha = 1°$. Instead of the Navier-Stokes equations (1), for this feasibility study, we restrict ourselves to the compressible Euler equations, that is the viscous terms in (1) are neglected. An illustration of this test case is given in Figure 3.



(a) Computational Mesh

(b) Pressure Coefficient Distribution

Figure 3: Illustration of the Inviscid NACA0012 Test Case

To set up the database, computations are performed with respect to a variation of the parameters shown in Figure 2. CODA implements an implicit solution method based on a backward Euler method and the time step is calculated based on a choice of a ramping strategy of the CFL number. For a large CFL number, the method is close to an approximate Newton's method, for a small CFL number it has the character of the backwards Euler scheme. Since Newton's method requires a suited initial guess, in the beginning of the time iteration the CFL number is

small, and it is then successively increased. This ramping is steered by several parameters and an overview of these parameters is shown in Figure 2. In total 900 CFD simulations were done using CODA. The simulation are terminated and considered successful if the density residual norm is below 10^{-15}. Alternatively, if the number of maximum iterations has been exceeded (10 000 iterations) the simulation is also terminated but regarded as non converged.

Tables 1 and 2 represent sample data sets from the produced database. Table 1 shows the set of passive and active input parameters, while Table (2) shows the set of output parameters in the logarithmic scale . It's worth mentioning that the first two columns in Table 2 (Fluctuation C_L and Fluctuation C_D) are calculated according to Equations (3), (4), (5) and (6). Those output parameters will be used to calculate the scoring function (last column of Table (2)), according to Equation (2) using the following weighting coefficients: $\alpha = 0.2$, $\beta = 0.2$, $\gamma = 0.2$, $\delta = 0.4$. Since there exist various CFL ramping strategies in CODA, not all parameters will be available in every simulation, thus we denoted the inactive parameters in Table 1 as NA (Not Applicable). This simplifies the problem at hand in this early stage, but certainly a more intelligent solution will be implemented in a later stage.

Table 1: Database Example [Input Parameters]

Simulation	**CFL Type**	**CFL Number**	**CFL Max**	**CFL SER Exponent**	**Geometric Ratio**
1	Constant	1.2685	NA	NA	NA
2	SER Ramp Max Refmax	1.18775	7.42e5	0.2375	NA
3	SER Ramp	1.31125	4.57e5	0.4075	NA
4	Geometric Ramp	1.0025	8.95e5	NA	4.47775

Table 2: Database Example [Output Parameters]

Simulation	**Fluctuation** C_L	**Fluctuation** C_D	**Density Residuals**	**Run Time**	**Scoring Function** f
1	-9.426	-9.896	-6.333	0	-5.13
2	-8.155	-8.752	-15.014	-1.25	-6.88
3	-5.448	-7.265	-15.038	-1.84	-6.28
4	-12.67	-13.54	-15.01	-2.4	-9.204

Having the database available, the next step is to create the surrogate model. In total 30% of the initial database is used for testing the accuracy of the different surrogate models, while 70% is used for training. The Nearest Neighbor (KNN) algorithm was used first, changing the number of neighbours (K) variable and monitoring the Mean Square Error and the coefficient of determination (R2). The results are graphed in Figure 4, thus concluding that the Nearest Neighbor algorithm works best with a minimal error and a maximum R2 score when K=2.

Furthermore, using K=2, Figure 5 shows the capabilities of this type of modeling on the test set divided for better visualization between the SER Ramp and SER Ramp Max Refmax CFL types (Figure 5a), the Geometric Ramp CFL type (Figure 5b) and the Constant CFL type (Figure 5c). The graphs show the predicted fitness of every point in the test set with their respective inputs, and the color map represents the error calculated as predicted score minus

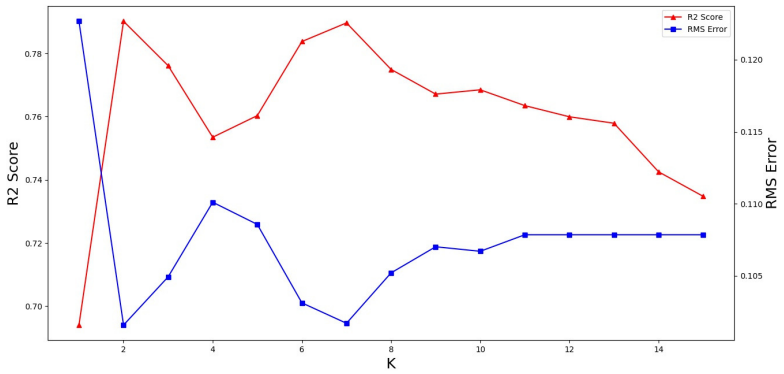


Figure 4: Nearest Neighbour Error Metrics

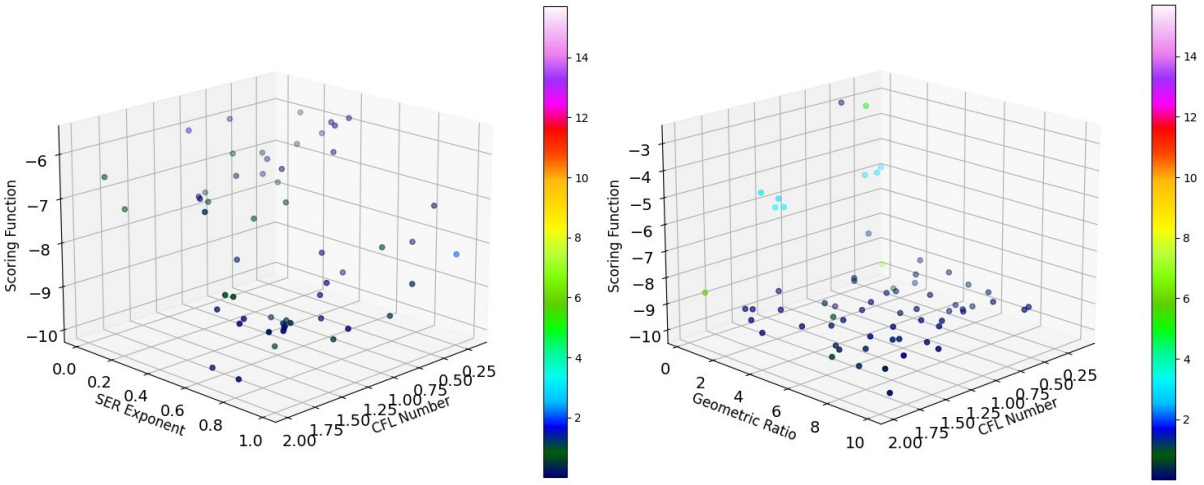
Table 3: Surrogate Models Comparison

Surrogate Model	**Root Mean Square Error**	**R2 Score**
Nearest Neighbor (K=2)	0.101862042	0.790181633
Thin Plate Spline	0.064954564	0.9146825

actual score. The figures show what is expected based on theory, that is with the increase of CFL number the solver tend to converge faster. We also see how the SER exponent for SER Ramp and SER Ramp maxRefmax strategies and the geometric ratio for the geometric ramp strategy influence the convergence of simulations which is also inline with the general theory. The error metric shows the discrepancy between the predicted solution and the solutions computed using CODA.

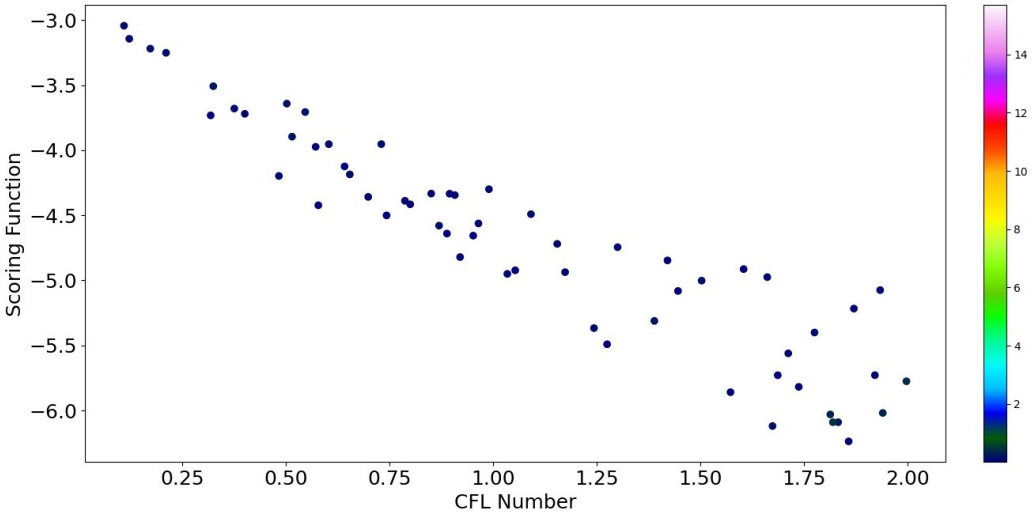
The Thin Plate Spline algorithm has also been applied to verify if an improved accuracy compared to the KNN algorithm can be achieved. The SMARTY TPS module was used here employing a linear trend function, with an LU decomposition solver, and a regularization constant of 1e-6. Comparing Figure 6 to Figure 5 we can see that the data points have a more intense blue color indicating a lower error compared to the KNN surrogate model. In fact, if we compare the Root Mean Square errors and the R2 scores (shown in Table 3) it is clear that the TPS surrogate model outperforms the KNN approach. For this reason, the TPS surrogate model is used for all following investigations.

After training the surrogate model, it will now be used in a loop with the GA. An initial population of 20 solutions, with the number of parents selected to create the next generation is set to 4, and then the GA will run for 50 generations. After implementing this procedure the optimal solver parameters are shown in Table 4. Since every CFL type activates different sub-parameters, the table shows four rows one for every CFL type that was optimised. We can observe that the best results are obtained for the Geometric Ramp type with a fitness of -13.20. After running the GA and obtaining these parameters, CODA was used to perform simulations using the parameters found by the GA.We observed that the most efficient CFL type in terms of iterations computed is the Geometric Ramp strategy. Furthermore, we can see



(a) SER Ramp + SER Ramp Max Refmax

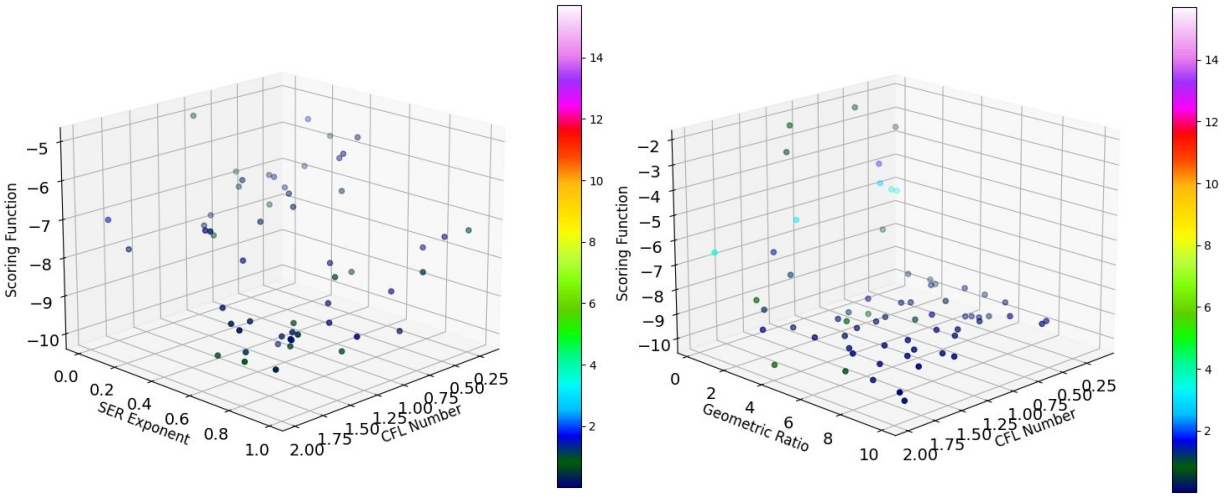
(b) Geometric Ramp



(c) Constant

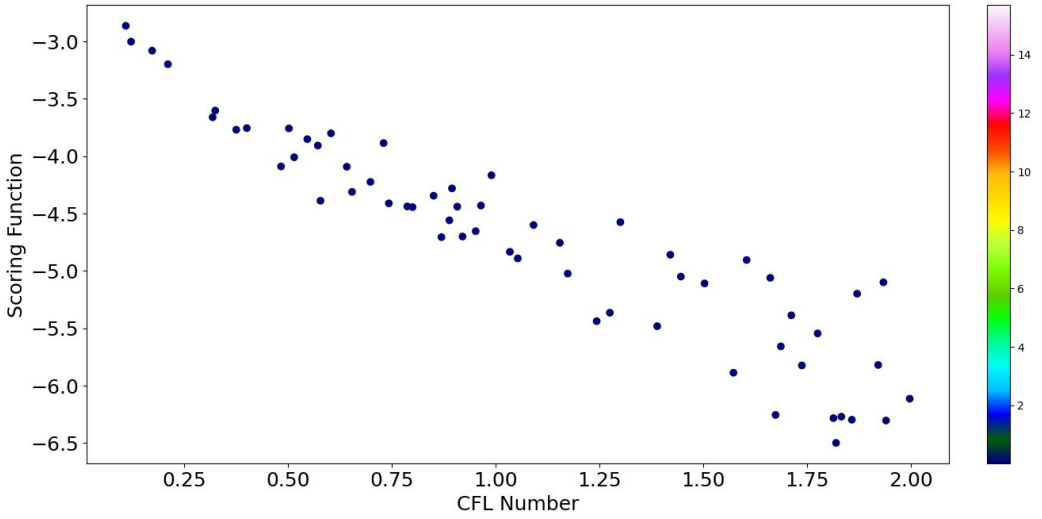
Figure 5: Nearest Neighbour Algorithm Predictions

that the Geometric Ramp strategy was able to reach stable Lift and Drag coefficients and low residuals faster than any other CFL strategy. Moreover, we computed the real scores of the data given in Table 4 using CODA, and we concluded that the results are inline with the predictions of the surrogate model. The errors in percentage between the predicted values and the values computed using CODA are shown in the final two columns in Table 4.



(a) SER Ramp + SER Ramp Max Refmax

(b) Geometric Ramp



(c) Constant

Figure 6: Thin Plate Spline Algorithm Predictions

Table 4: Best solver parameters per CFL strategy

CFL Type	**CFL Number**	**CFL Max**	**CFL SER Exponent**	**Geometric Ratio**	**Predicted Score**	**Actual Score**	**Error**
Geometric Ramp	1.99	4.37e7	NA	9.9	-13.20	-9.5	38.9%
SER Ramp	1.66	8.43e5	0.556	NA	-10.75	-9.37	14.7%
SER Ramp Max Refmax	1.38	7.36e6	0.64	NA	-10.59	-9.36	13.1%
Constant	1.96	NA	NA	NA	-6.47	-6.597	1.9%

4 Conclusion

In this work, the idea and general approach towards an **Expert System** for CFD codes was presented. The discussion includes the classification of the parameters, the technical hurdles involved in setting up a database, the selection and generation of a surrogate model, and algorithms that automatically determine parameters for new test cases in order to get accurate solutions that are computed using efficient algorithms. Furthermore, a proof of concept was made for the CFL number ramping strategy in CODA, where we took the parameters that govern the CFL number as inputs to the database while using some CFD post-processing parameters to compute the scoring function or the output of the database. The model was trained using Nearest Neighbour algorithm and the Thin Plate Spline interpolation method. The trained surrogate was then used alongside a genetic algorithm to optimise the CFL parameters. The study shows a promising behavior of the Expert System, although on a primitive level. In the case study of the NACA0012 with a Mach number of 0.5 and an angle of attack of 1, the optimal CFL type was found to be the Geometric Ramp with an initial CFL number of 1.02, maximum CFL number of 4.14e7 and a Geometric ratio of 8.75. The next step was to evaluate the score of these parameters using CODA.

Future work will concentrate on improving the surrogate modeling part. Neural networks are known to be more accurate than any algorithm used in this study, particularly with the non linear behaviour we observed in our case. Moreover, they enable to respect the tree-like structure most parameters have. We are also investigating the potentials of using the hidden gene genetic algorithm (HGGA) which helps to use only one GA to investigate the entire design space using the variable chromosome length strategy. On the other hand, we are looking on extending the database to include a range of Mach numbers and angles of attacks. In addition, we are planning to shift from the Euler equations to RANS equations thus incorporating the effect of turbulence models into the surrogate model and further increasing the amount of involved parameters. Finally, we need to include other test cases and connect the Expert System to a suited database system.

REFERENCES

[1] N. Kroll, M. Abu-Zurayk, D. Dimitrov, T. Franz, T. Führer, T. Gerhold, S. Görtz, R. Heinrich, C. Ilic, J. Jepsen, J. Jägersküpper, M. Kruse, A. Krumbein, S. Langer, D. Liu, R. Liepelt, L. Reimer, M. Ritter, A. Schwöppe, J. Scherer, F. Spiering, R. Thormann, V. Togiti, D. Vollmer, and J.-H. Wendisch. DLR project Digital-X: Towards Virtual Aircraft Design and Flight Testing based on High-fidelity Methods. *CEAS Aeronautical Journal*, 7:3–27, 2016.

[2] Arpit Aggarwal, Ralf Hartmann, Stefan Langer, and Tobias Leicht. Steady-State Flow Solutions for Delta Wing Configurations at High Angle of Attack using Implicit Schemes. *New Results in Numerical and Experimental Fluid Mechanics XIII*, 151:271–281, March 2021.

[3] Sebastian Braun, Stefan Langer, and Tobias Leicht. Convection Treatment for RANS Turbulence Model Equations. *22nd STAB/DGLR Symposium on New Results in Numerical and Experimental Fluid Mechanics XIII*, 151:646–655, October 2021.

[4] Stefan Langer. Investigations of a Compressible Second Order Finite Volume Code Towards the Incompressible Limit. *Computers & Fluids*, 149:119 – 137, 2017.

[5] Stefan Langer. Agglomeration Multigrid Methods with Implicit Runge-Kutta Smoothers Applied to Aerodynamic Simulations on Unstructured Grids. *Journal of Computational Physics*, 277(0):72–100, 2014.

[6] Stefan Langer. Preconditioned Newton Methods to Approximate Solutions of the Reynolds Averaged Navier-Stokes Equations. Technical report, Institut für Aerodynamik und Strömungstechnik, June 2018.

[7] John Dannenhoffer III and Judson R. Baron. A Hybrid Expert System for Complex CFD Problems. In *8th Computational Fluid Dynamics Conference*.

[8] P. Kutler and U. Mehta. Computational Aerodynamics and Artificial Intelligence. In *17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*.

[9] P. Kutler, U. Mehta, and A. Andrews. Potential Application of Artificial Concepts to Aerodynamic Simulation. pages 340–345, 1985.

[10] Anthony D. Williams. The Development of an Intelligent Interface to a Computational Fluid Dynamics Flow-Solver Code. *Computational Structural Mechanics and Fluid Dynamics*, page 431–438, 1988.

[11] Ricardo Vinuesa and Steven L. Brunton. The Potential of Machine Learning to Enhance Computational Fluid Dynamics. 10 2021.

[12] Lars Reimer, Ralf Heinrich, Sven Geisbauer, Tobias Leicht, Stefan Görtz, Markus Raimund Ritter, and Andreas Krumbein. Virtual Aircraft Technology Integration Platform: Ingredients for Multidisciplinary Simulation and Virtual Flight Testing. In *AIAA SciTech Forum*, January 2021.

[13] Tobias Leicht, Jens Jägersküpper, Daniel Vollmer, Axel Schwöppe, Ralf Hartmann, Jens Fiedler, and Tobias Schlauch. DLR-Project Digital-X - Next Generation CFD Solver 'Flucs'. In *Deutscher Luft- und Raumfahrtkongress 2016*, February 2016.

[14] Tiffany Timbers, Trevor Campbell, and Melissa Lee. *Data Science: A First Introduction (Chapman & Hall/CRC Data Science Series)*. Chapman and Hall/CRC, 1 edition, 2022.

[15] Jean Duchon. Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer Berlin Heidelberg, 1977.

[16] D G Rossiter. Exercise: Thin Plate Spline Interpolation, March 2016.

[17] Philipp Bekemeyer, Anna Bertram, Derrick Armando Hines Chaves, Mateus Dias Ribeiro, Andrea Garbo, Anna Kiener, Christian Sabater Campomanes, Mario Stradtner, Simon Wassing, Markus Widhalm, and Stefan Görtz. Data-Driven Aerodynamic Modeling using the DLR SMARTy Toolbox". AIAA Aviation, 2022.

[18] Melanie Mitchell. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. MIT Press, reprint edition, 1998.

[19] Ahmed F. Gad. Pygad: An intuitive genetic algorithm python library. June 2021. PyGAD package version 2.16.3.