Evolutionary Design Optimisation of Complex Systems integrating FLUENT for parallel Flow Evaluation

Stefan Menzel, Markus Olhofer, Bernhard Sendhoff

2005

Preprint:

This is an accepted article published in Proceedings of the 2nd EACC - 2005 European Automotive CFD Conference. The final authenticated version is available online at: https://doi.org/[DOI not available]

Evolutionary Design Optimisation of Complex Systems integrating FLUENT for parallel Flow Evaluation

Stefan Menzel, Markus Olhofer, Bernhard Sendhoff
Honda Research Institute Europe GmbH
Offenbach/Main, Germany
Email: {stefan.menzel, markus.olhofer, bs}@honda-ri.de

ABSTRACT

Evolutionary algorithms have been successfully applied to various design optimisation problems. However, the success of the optimisation is strongly influenced by its components which have to be adjusted carefully to the given task. Main ingredients of an efficient evolutionary design optimisation are a proper representation and an adequate evaluation tool for calculating the performance index of every evolved design. Furthermore a fully automated overall process with minimal user interaction is needed in order to allow an efficient usage of computational power.

In the present paper an evolutionary optimisation is introduced which takes advantage of the concepts of free form deformation (FFD) for an efficient representation and the comfortable interface of FLUENT as parallelized flow solver in a computer cluster environment. A special focus is given to an autonomous adaptation of the design parameters which is achieved by an extension of the FFD method. The proposed methods are illustrated for a two-dimensional airfoil optimisation with the focus on the adaptivity of the chosen representation and its influence and impact on the evolutionary design optimisation process.

1. INTRODUCTION

The efficient development of optimal shape geometries plays a key role in every product design process. The integration of computational simulation methods into this process supports the designers and engineers. More and more additional information is available and mathematical models speed up the design process. In many cases, we can observe that autonomous design optimisation leads to a further essential reduction of development time. However, these methods are only applicable if certain requirements are fulfilled. At first it is important to guarantee the reliability of computer software, e.g. CFD or FEM simulators, which model the behaviour of developed designs in real world situations. Furthermore, the methods which drive the whole optimisation process have to be as efficient as possible to reduce the total number of optimisation steps and hence optimisation time while finding at the same time an optimal or suboptimal solution even in case of complex quality functions.

A multitude of methods and algorithms has been proposed in the past to find a design solution which performs best with respect to the technical and geometrical constraints of a given task. Besides deterministic approaches like gradient based methods, evolutionary algorithms have become more and more popular for the optimisation of complex systems. This is mainly due to their ability to overcome local optima and to allow the optimisation of noisy quality functions. Furthermore, the considerable increase of available computing power made the use of population based search methods like evolutionary algorithms more feasible. Evolutionary algorithms which mimic the concepts of biological evolution can be applied to standard design optimisation problems where a fine tuning of an existing parameter set is the target of optimisation. Due to their properties they are also able to discover new and unexpected solutions which are not necessarily based on heuristical experiences of engineers and designers any more. This is only possible if the underlying representation allows for those changes. In the beginning of each optimisation process the design with the maximum performance is unknown. Therefore, it is often difficult to define suitable parameters, i.e., a representation of the shape, with sufficient degree of freedom. This problem could be solved by including as many parameters as possible with the drawback that the dimensionality of the design space increases leading to a high number of optimisation steps and hence a high degree of computational resources and time. The main focus of the presented method is put on an adaptation method which allows to add parameters during the optimisation process. The result is a higher degree of shape flexibility and locality while keeping a low number of optimisation parameters and in the same way keeping the amount of necessary a priori knowledge as small as possible.

In the first part of this paper, we will introduce the theoretic concepts of evolutionary algorithms focusing on a special variant of Evolution Strategies, the covariance matrix adaptation (CMA-ES). Additionally, the chosen representation, the free form deformation, is briefly described focusing on the proposed extension which allows the adaptation of the representation as well as its consequences for the optimisation. With respect to the parallel fitness evaluation the integration of FLUENT for an automized flow calculation in a computer cluster is shown. In the second part of the paper, the basic applicability of the proposed methods is illustrated by an example of a two-dimensional airfoil design optimisation which is based on a FLUENT tutorial case.

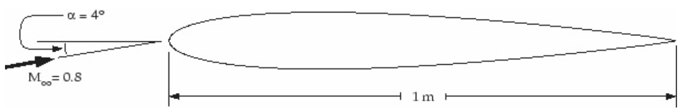

Figure 1. Airfoil Geometry [1]

2. EVOLUTIONARY OPTIMISATION OF COMPLEX SYSTEMS

Evolutionary algorithms belong to the group of stochastic optimisation algorithms. They mimic the concepts of biological evolution by applying operators for reproduction, mutation and/or recombination and selection. Prominent examples are Evolution Strategies (ES) and Genetic Algorithms (GA). Among the advantages of evolutionary algorithms are robustness against noisy or discontinuous quality functions, the ability to escape from local optima and to enable global search. In this paper, a special variant of Evolution Strategies, the Covariance Matrix Adaptation (CMA), is applied which provides the advantage of a high convergence rate for real-valued problems compared to other evolutionary algorithms. This characteristic is of special importance in the case of very time consuming evaluations like CFD simulations. The successful application of this type of algorithm has been shown previously e.g. for a turbine blade optimisation [2].

The remainder of this section is organized as follows: An overview about evolutionary algorithms is given in 2.1, followed by a brief discussion of the special type of the CMA-ES in 2.2. In Section 2.3 the design representation is explained in more details which is chosen in the airfoil optimisation, followed by an explanation of its combination with the applied flow solver FLUENT in a computer cluster environment in Section 2.4. Finally, the extension of the representation which allows its online adaptation and its consequences for the evolutionary optimisation are shown in Section 2.5.

2.1 Concepts of Evolutionary Optimisation

A typical evolutionary design optimisation procedure is depicted in Figure 2 using the airfoil example. First, the initial design which is the starting point of the optimisation is parameterized and its significant parameters are determined. These parameters are encoded as the so-called genotype in the parent's chromosome. In the reproduction phase parents are chosen randomly and copied to form the offspring population. The generated offspring individuals which are exact copies of their parents are randomly mutated without any recombination. In Evolution Strategies this is done by adding a normally distributed random vector with zero mean. The variance is determined by a set of so-called strategy parameters which play the role of a step size for the variations. In order to adapt the step size optimally to the topology of the search space they are subject to the same process of evolutionary optimization as the object parameters. Thus, the search strategy is adapted during search, this type of meta-evolution is usually referred to as self-adaptation.

The strategy parameters determine the search strategy. At the same time they provide information about the state of the whole optimisation process. A large step size indicates a more macroscopic search whereas a smaller one suggests a local search in a limited region of the search space. Finally, very small step-sizes also indicate the convergence of the optimisation. This behaviour will also be illustrated in the airfoil optimisation example. For the adaptation of the strategy parameters several different methods have been proposed which will be mentioned below.

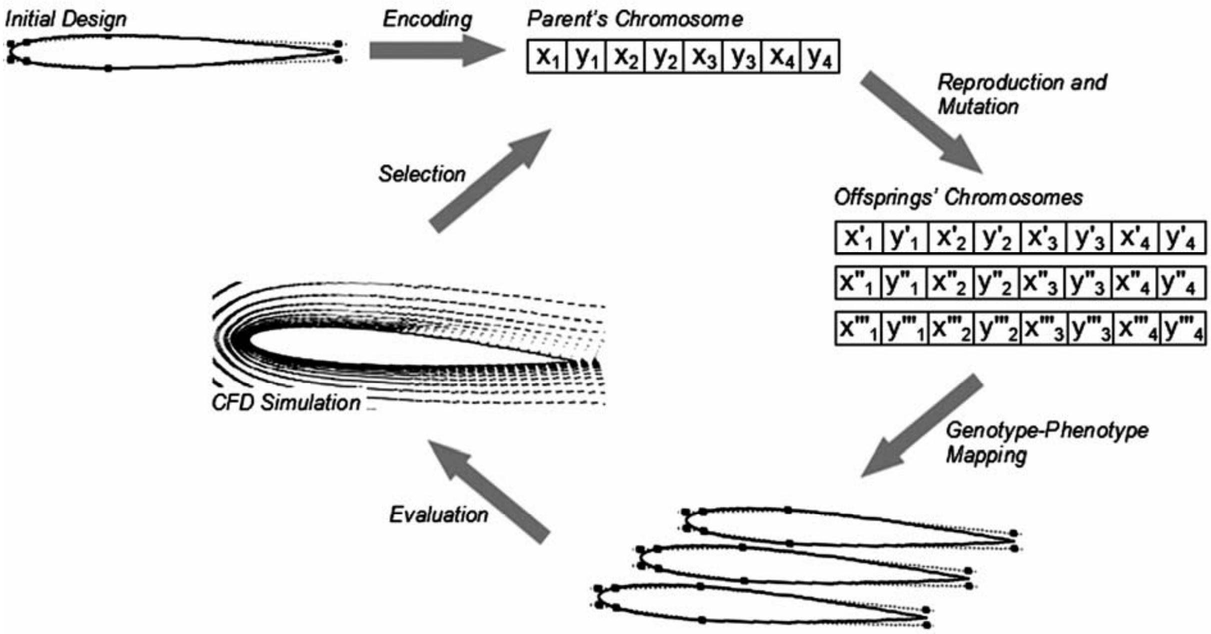

Figure 2. The generation cycle in evolutionary design optimisation

After the mutation of the offspring the genotype is mapped to the phenotype, i.e., the actual designs are created based on the encoded parameter set and prepared for the evaluation. In this phase a fitness value is assigned to each design which determines the performance with respect to realistic conditions. When optimizing a fluid dynamical system, typically a CFD calculation is performed and a selected result, e.g. a force, pressure or temperature, is taken as the controlling fitness value. Based on the calculated fitness the best offspring is picked according to the chosen selection method. For Evolutionary Strategies two different kinds of selection operators exist, one is called (μ+λ)-selection, the other (μ,λ)-selection [3]. They differ in the way of choosing the offsprings which form the parent generation of the next generation. In a (μ+λ)-strategy the new parents are selected from the sum of the parents and the offsprings of the current generation whereas in a (μ,λ)-strategy only the offsprings are considered. In the first case a constant increase of the performance index is guaranteed because the best individual is kept in every generation but likewise it includes the danger that the evolution gets stuck in a local optimum. In the airfoil optimisation a (μ,λ)-strategy is applied, i.e. the selection takes place only on the offsprings. After this selection the process starts with a new reproduction cycle until the algorithm terminates when a certain stop criterion, e.g. a convergence threshold for the step size, is fulfilled.

2.2 Covariance Matrix Adaptation (CMA-ES)

For the performance of Evolution Strategies a crucial aspect is the adaptation of the strategy parameters which highly influence the behaviour of the search and finally the convergence speed of the algorithm. As explained in the previous section the strategy parameters mainly define the variance(s) of the normally distributed random vector which is added to the object parameter during the mutation phase. In the simplest case, one single strategy parameter determines the variance of all elements of the random vector. The strategy parameter itself can be adapted by a self adaptation process. Therefore, the same process of evolution is applied to it as to the object parameter. Instead of one single strategy parameter the CMA algorithm adapts the covariance matrix of the distribution of the random vector. Furthermore, the path of the evolution is observed in order to predict favourable mutations based on the history of the optimisation. The theoretical concepts of the CMA-ES and its successful application in test scenarios as well as in a CFD optimisation environment have been discussed in various papers [2], [4], [5]. A detailed description of the algorithm can be found in [6] and an implementation in a software library in [7]. To summarize, there are mainly three features of the CMA-ES which are important. Firstly, the stochastic influence in the mutation step is reduced by introducing only one stochastic source which is used for modifying both, the object as well as the strategy parameters. The actual mutation of a selected and successful individual which is responsible for the mutation of the object parameter is used for the adaptation of the strategy parameter. Secondly, the so-called cumulative step-size adaptation is applied which extracts information from past generations to speed up and stabilize the adaptation of the strategy parameter. Thirdly, an adaptation of the full covariance matrix of the probability density vector takes place instead of independent variances for each single parameter. Therefore correlated mutations can be realized which can significantly increase the convergence speed of the algorithm [3], [6].

2.3 Free Form Deformation as Representation

The choice of an adequate shape representation, i.e., the set of object parameters, is crucial for the optimisation for various reasons. First, the representation defines the properties of the map from the genotype space (the object parameter space) to the phenotype space (the space of actual designs). The properties of this map are important because they can change the difficulty of the search process; one frequently quoted such property is strong causality [3]. Another one is completeness which is closely coupled to compactness. As mentioned previously, the representation should be very flexible, allowing to encode any feasible design. At the same time, the dimensionality of the search space should be kept minimal, i.e. the representation should be compact. Lower dimensionality will – in general – lead to faster convergence while only completeness guarantees that the optimal solution (whatever it might look like) can be *represented* by the chosen parameterization or representation. To overcome this trade-off between compactness and completeness we propose an adaptive extension of free form deformations which is explained in Section 2.5.

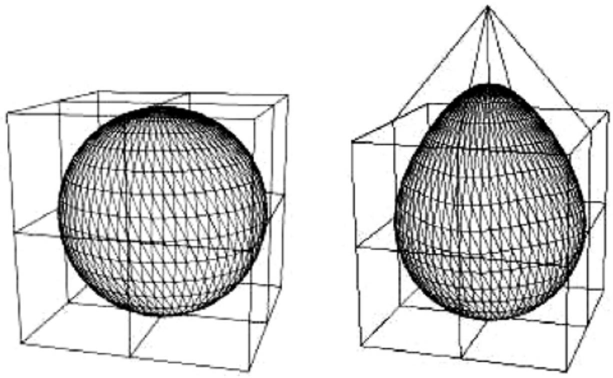

Figure 3. Free Form Deformation [10]

The concept of free form deformation has been introduced by Sederberg et al. [8] in the field of solid and surface modelling and has been extended and generalized by Coquillart [9]. Geometries are embedded within an arbitrary control volume which is defined by a set of control points. By modifying these control points deformations can be applied to the geometry and the resulting shape is calculated based on a system of trivariate Bernstein polynomials or B-splines respectively as mathematical foundation. Additionally, if this kind of representation is combined with evolutionary optimisation which requires CFD evaluations for determining the performance of designs another advantage can be identified. Because of the possibility to represent not only the surface mesh of a body but also all grid knots within the control volume, shape designs and CFD grids can be evolved simultaneously so that a costly mesh generation process can be omitted [10]. This property is especially important for complex systems because the CFD mesh can be directly deformed while keeping its structural composition.

In the context of design optimisation two aspects of a free form deformation representation should be noted. On the one hand, the initial chromosome which is needed for the start up of the optimisation loop has to be prepared by encoding the initial geometry. On the other hand, directly before the evaluation phase the chromosomes have to be extracted and mapped to the geometric definition (genotype-phenotype mapping).

Concerning the first aspect a lattice of control points has to be constructed which encloses either the whole object or the part of the object which will be modified during optimisation. In the second and most important step the geometry and grid coordinates have to be transferred into the parameter space of the lattice, a procedure which is also called “freezing”. When freezing an object the u, v and w coordinates of the geometry in spline parameter space are calculated. This is usually done by Newton approximation which is regarded to be the fastest approach but it can also be done by similar gradient based methods [9], [11]. Finally, the positions of the control points which are chosen for the deformation of the object are encoded in the parent’s chromosome as optimisation parameters. Practically, when this method is combined with CFD, at first the initial design is meshed. In a second step the part of the design which is subject to optimisation and its surrounding part of the CFD mesh is embedded within an adequate control volume. The arrangement of the control points should be chosen carefully to keep the number of parameters as low as possible. Additionally, the objective control points should be fixed close to the surface of the design to maximize their influence. After the control points have been selected the coordinates in parameter space can be calculated by freezing the geometry and the CFD mesh.

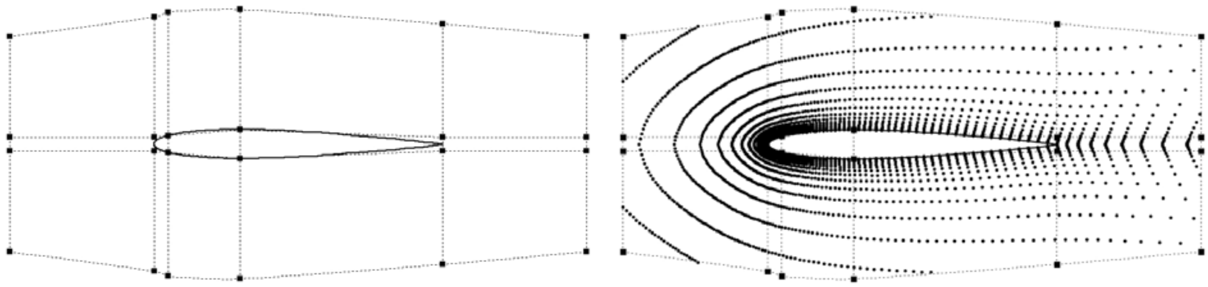

Figure 4. Set-up of the control volume

After the mutation step the new coordinates of the control points are extracted from the chromosome and the deformations are transferred to the design geometry as well as to the CFD mesh. By solving the B-spline equations using the new spatial coordinates of the control points the x, y and z coordinates of the design surface and the grid knots are updated. In case of combining this method with FLUENT as CFD solver the new grid coordinates are calculated and the case-file is modified by replacing the original grid coordinates with the new ones. This guarantees that the mesh is modified while the structural composition is kept the same. Via the possibility of journal files the whole evaluation with FLUENT is fully automized which is the subject of Section 2.4.

2.4 Evaluation using Fluent as parallelized Flow Solver in a Cluster Environment

While working with evolutionary design optimisation a high degree of process automation is desired to reduce manual interaction which can be error prone. In order to assign performance indices to each design, in every generation each of the λ offspring has to be evaluated. If CFD simulations are used the design has to be meshed, the case-file has to be produced, the numerical analysis has to be performed and a file containing the results has to be processed. These procedures have to be carried out for each design in the population, i.e., the set of designs. The inherently parallel structure of evolutionary algorithms allows the straightforward coarse-grained parallelization of the optimization process. All individuals can be evaluated in parallel using the Parallel Virtual Machine (PVM) framework in a master/slave configuration [12], [13]. In the pre-evaluation phase the meshes are modified using the free form deformation technique and the case-files for each offspring are produced. For each case a FLUENT journal file is automatically generated which contains all required information about path', case-setups and data storage. In the evaluation phase, each single computer node is notified by the master process to start-up a FLUENT solver and to initiate the numerical calculations using the corresponding FLUENT journal file. When the flow results are available the master process filters the fitness value from the transcript file and assigns it to each offspring so that the selection phase can be started. In the present airfoil design optimisation a total number of 377 generations à 14 offspring has been evaluated. Additionally 8 subpopulations à 14 offspring have been calculated in each of the two adaptation phases which are the subject of Section 2.5 resulting in 7238 flow solver calls.

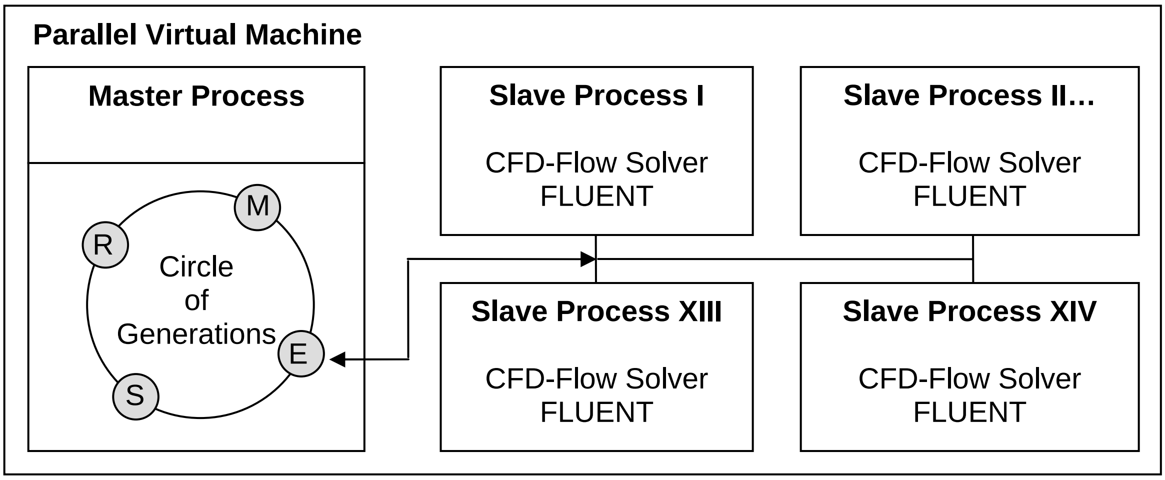

Figure 5. Parallelized computing in a cluster environment. We distinguish the Mutation (M), Evaluation (E), Reproduction (R) and Selection (S) phases.

2.5 Adaptation of the Representation

In order to solve the previously mentioned trade-off between compactness and completeness of the representation of the design, we will introduce in this section an adaptive representation [14], i.e., a representation which can change during optimization. During optimization new control points are introduced, thereby increasing the dimensionality and the design variability step by step during search. At the same time, the self-adaptation process has the tendency to decreases the strategy parameters, the variances of the normal distribution. Thus, while increasing the dimensionality of the search space with the adaptive representation, the search also becomes more and more local due to the decreasing strategy parameters. Therefore, we can interpret the process in the following way: Firstly, the whole design space is searched for promising areas. Secondly, the dimensionality is increased, thereby opening up new *variation opportunities*. However, due to the decrease of the "old" strategy parameters, the new degrees of freedom are restricted to the previously identified promising region. In other words, previously optimized control points will only experience slight changes whereas newly added control points will change more dramatically. This way a search in sub-spaces is organized by the search process itself. While in practice old strategy parameters nearly always decrease, theoretically – if necessary for the success of the search – even old strategy parameters can increase again. Thus, the whole process is much more flexible compared to a more straightforward approach to simple keep old parameters fixed when new ones are introduced.

Having this behaviour in mind one can argue that only a rough description is necessary in the first phase of the optimisation. This can be realised by only a few parameters. Later on, the description should be more and more precise and focus on a region near a found sub-optimum. Here a high number of parameters is needed but the search space is limited due to the local search.

Free form deformations can be extended by adding additional control points without changing the actual shape in the design process. Thereby, we can change the design space and the representation of each solution without changing its shape and quality. In biology, this effect that a change in genotype space (the parameter space) does not lead to a change in phenotype space (the shape space) is called neutrality and the corresponding change a neural change. Neutrality is a very important property of the spline representation and therefore also of the free form deformation. Neutrality allows to change the design space without running the risk that the performance of the shapes is decreased.

The choice in which direction to increase the dimensionality of the design space or in other words where to insert an additional control point is crucial for the success of the adaptive representation. It can be done manually, e.g. using expert knowledge to determine where an increase of variability might be particularly beneficial for the optimization process. An alternative is to integrate the adaptation process into the evolutionary search. First, a number of different variations of the representation of an individual are generated by introducing additional points at random positions. In a second step all of these individuals are optimized in sub-populations in order to determine which search space extensions offer the most promising quality increase. The best changes of the representations are kept and integrated again into the overall search process. The target of this sub-population framework is to determine the potential of representational changes not necessarily to increase the overall performance.

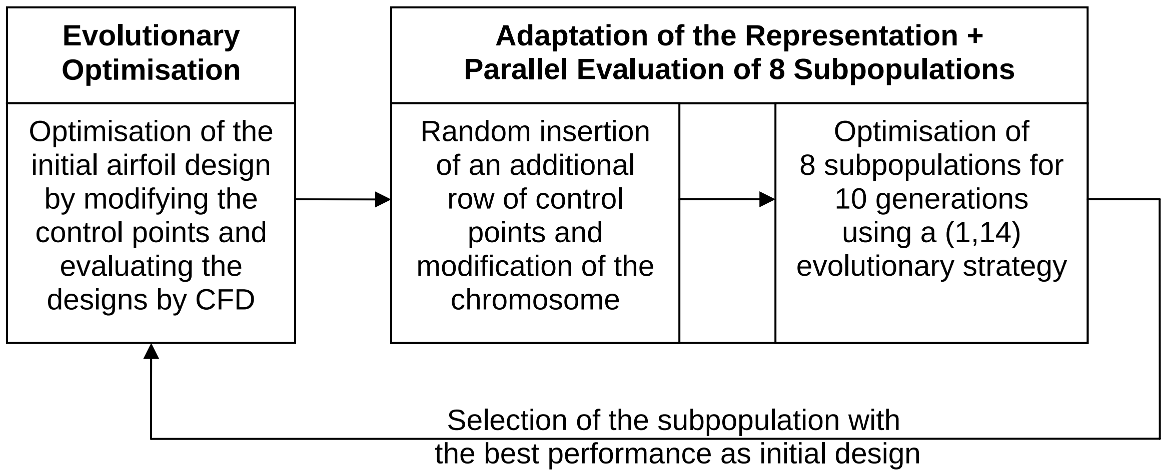

Figure 6. Workflow of an evolutionary strategy with adaptation

In the present airfoil optimisation the representation is adapted as follows. At first the initial design is optimized until the step size converged in generation 100. The resulting design D100 with a corresponding set of control points which is stored in the parent's chromosome is used for the adaptation process. To refine the representation eight sub-populations are set up which differ by the position of the inserted control points. In each sub-population a random number determines the position in the control volume where a new row of control points is inserted. The coordinates of these control points are calculated by a linear interpolation between the two neighbouring control points. Based on the extended control volume the coordinates of the geometry D100 are calculated in parameter space so that the current best design is unaltered. To proceed with the optimisation the chromosome of the parent is modified by adding the coordinates of the new control points. After these modifications took place the optimisation of each sub-population is continued for ten generations and the best design is determined by a comparison to the current performance index. The best one is selected and is the subject of the continuing optimisation process. Using this method the most promising design candidate is selected because the insertion of the new row of control points resulted in the biggest performance increase and hence these control points seem to have a major influence on the design. The adaptation phase is followed by another 120 generations before a second adaptation phase is carried out using the same pattern as described above. This optimisation method results in an optimized design which will be presented in section 3.

3. Test Scenario: 2D Airfoil Optimisation

The test scenario which is selected for illustrating the described evolution strategy is based on the example 3 in the FLUENT Tutorial Guide "Modelling External Compressible Flow" [1]. In Section 3.1, a short description of the configuration, constraints and assumptions is given, for more detailed informations on the set-up of the CFD calculation please see also [1]. The two-dimensional airfoil has been chosen because the relatively low number of cells and grid points limits the computation time needed to evaluate each design in the population which is the most critical factor in an optimisation involving CFD simulations. An average calculation took about 4 minutes using a PIII Xeon, 2.0 Ghz node, initialized with a pre-converged flow field. Therefore, the two-dimensional task provides a good middle course between showing the workflow of the optimisation and a resource friendly example. The proposed methods can be straightforwardly extended to three-dimensional design optimization tasks with more complex constraints. Additionally, we should note, that the final result of the optimisation is more or less artificial and does not claim to be of any practical relevance; it merely serves us as a demonstration of the feasibility of our optimization framework.

3.1 Problem Description and Constraints

Subject to the optimisation is the airfoil whose cross-section can be found in Figure 1. The velocity of the air is given by a mach number of 0.8 and the angle of attack is 4°. Before starting the evolutionary optimisation the wing is embedded in a control volume as shown in Figure 4 and Figure 7. In order to increase the influence of each control point the ones numbered 1, 2, 3 and 4 are positioned close to the geometry. These four points, i.e. their x and y coordinates are taken as object parameters and hence, they are encoded in the chromosome of the parent. The remaining four points shown in Figure 7 are kept constant because a movement of these points implies a strong deformation of the leading edge of the airfoil. If these points would be optimized too, the whole position of the wing would dramatically change and consequently the angle of attack. Therefore, the final result could hardly be compared to the initial settings.

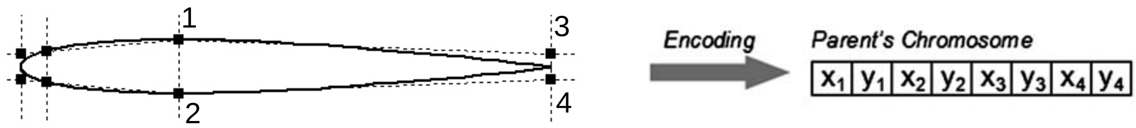

Figure 7. Definition and encoding of the initial control points

As optimisation criteria the total lift of the wing is chosen which initially equals 15500 N. As additional constraints for the new evolved designs a minimum cross-section area and a minimum thickness have been introduced to prevent the airfoil from becoming too thin.

3.2 Optimisation Results

Based on the initial design a total number of 377 generations have been calculated. The course of the fitness and the global step size can be found in Figure 8. As stated above the whole optimisation falls into three sub-optimisations. Based on the initial definition 100 generations have been calculated which lead to an increase of the fitness by 10700 N. As a good indicator about the state of the calculation the global step size can be analyzed. A larger step size (strategy parameters) indicates a large mutation of the object parameters whereas a lower one implies small mutations. It can be seen that in the beginning the step size increases fast because of the high rate of fitness changes. When the plateau is reached the step size converges towards zero indicating that an optimal solution has been identified, which we termed D100 and which is shown in Figure 9. The width of the design increased and the trailing edge moved upwards.

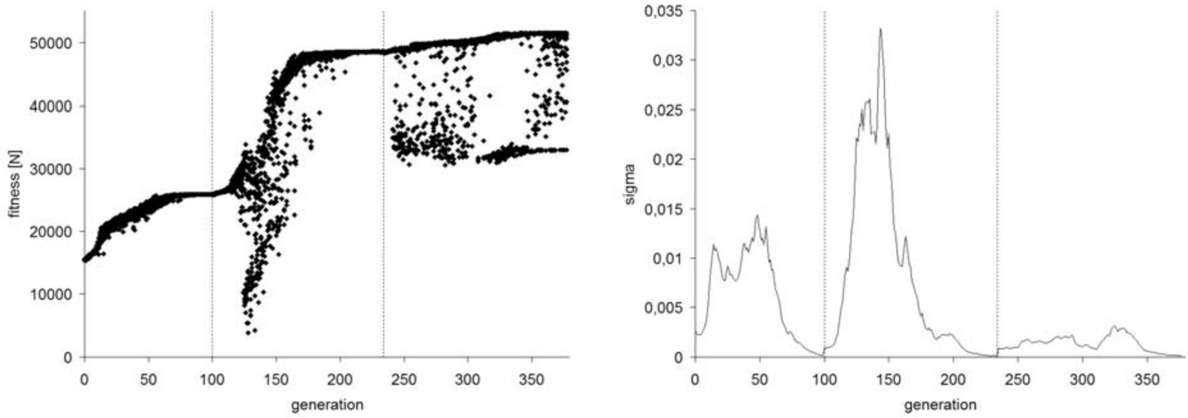

Figure 8. Course of fitness and step size during optimisation

By adapting the representation, i.e. inserting new control points, a higher degree of flexibility is achieved. New control points allow local deformations to increase the overall performance. As a consequence the lift increases to 48500 N and a strong correlation between the fitness increase and the increase of the step size can be observed. Simultaneously, to the increase of the global step size we observe a high variance in the fitness values which reflects the increased search dimension and variability. The resulting design is similar to the final result D377 in Figure 9. Both designs feature a bend in the middle section which is only possible because of the insertion of new control points in this region. After the second adaptation the fitness increase is comparably small, approx. 3000 N. Therefore, the adaptation process has been stopped and the optimisation loop has been terminated.

Initial Design D0 | Design D100 | Design D377

Figure 9. Designs D0, D100 and D377

4. CONCLUSIONS

In the present paper, an evolution strategy has been combined with an extended free form deformation technique. This method has – especially for a design optimisation of fluid dynamical systems – various advantages. For many complex systems the number of parameters can be drastically reduced using free form deformation. Especially, the required additional process of grid generation which can take several days for complex shapes can be integrated into the optimization framework. Since the mesh is deformed together with the shape no re-meshing is needed. For evolutionary optimisation a high degree of automation and parallelization is needed which can easily be realized by running FLUENT via journal files in a cluster environment and via the exchange of computational grids by modifying case files.

The method to adapt the representation allows an increase of flexibility during the optimisation process which is integrated in and governed by the process itself. As shown in the test scenario the performance of the design can be increased by introducing new control points which increase the dimensionality of the design space while allowing a higher degree of locality. However, there are also some limitations when using free form deformations with CFD which have to be considered carefully. With respect to the computational grid large modifications of the coordinates of the control points imply consequently large deformations of the cells. Therefore, it has to be guaranteed that the CFD mesh is still valid after deformation. The ratios of cell width to cell height have to be observed and changes in the order of the control points have to be prevented. In case of very large deformations the design has to be re-meshed and the optimisation has to be re-started based on the updated mesh.

Improving the adaptive representation will be one of our future research focus, in particular the question when to increase the variability of the design during the optimization process requires additional analysis.

5. REFERENCES

[1] FLUENT 6.2 Tutorial Guide. Example 3. Modelling External Compressible Flow. Fluent Inc. 2005-01-07

[2] Toyotaka Sonoda, Yoshihiro Yamaguchi, Toshiyuki Arima, Markus Olhofer, Bernhard Sendhoff and Heinz-Adolf Schreiber. Advanced high turning compressor airfoils for low reynolds number condition, Part 1: Design and optimization. In Proceedings of the ASME Turbo Expo, 2003

[3] Ingo Rechenberg. Evolutionsstrategie ’94. Friedrich Frommann Verlag · Günther Holzboog, Stuttgart-Bad Cannstatt 1994

[4] Yaochu Jin, Markus Olhofer and Bernhard Sendhoff. A Framework for Evolutionary Optimisation With Approximate Fitness Functions. In *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 5, pp. 481-494, October 2002

[5] Yaochu Jin, Markus Olhofer and Bernhard Sendhoff. Managing Approximate Models in Evolutionary Aerodynamic Design Optimization. In *Congress on Evolutionary Computation (CEC)*, volume 2, pages 592-599, Seoul, Korea, May 2001. IEEE Press.

[6] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. In *Evolutionary Computation*, Vol. 9, No. 2, pp. 159-196, 2001

[7] Shark – a modular C++ library for the design and optimization of adaptive systems. Open Source Software: http://sourceforge.net/projects/shark-project

[8] Thomas W. Sederberg and Scott R. Parry. Free-Form Deformation of Solid Geometric Models. In *Computer Graphics*, 20(4):151-160, 1986

[9] Sabine Coquillart. Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. In *Computer Graphics*, 24(4):187-196, August 1990

[10] Ernest C. Perry, Steven E. Benzley, Mark Landon and Richard Johnson. Shape Optimization of Fluid Flow Systems. In *Proceedings of ASME FEDSM'00*. 2000 ASME Fluids Engineering Summer Conference. Boston, Massachusetts, 2000

[11] Les Piegl and Wayne Tiller. The NURBS Book. Springer-Verlag Berlin Heidelberg 1995 and 1997

[12] Al Geist et al. PVM: Parallel Virtual Machine – A Users’ Guide and Tutorial for Networked Parallel Computing. Cambridge, Ma: The MIT Press, 1995

[13] PVM: Parallel Virtual Machine. http://www.csm.ornl.gov/pvm/pvm_home.html

[14] Markus Olhofer, Yaochu Jin and Bernhard Sendhoff. Adaptive encoding for aerodynamic shape optimization using evolution strategies. In *Congress on Evolutionary Computation (CEC)*, volume 2, pages 576-583, Seoul, Korea, May 2001, IEEE Press.