Direct Manipulation of Free Form Deformation in Evolutionary Design Optimisation

Stefan Menzel, Markus Olhofer, Bernhard Sendhoff

2006

Preprint:

This is an accepted article published in International Conference on Parallel Problem Solving From Nature (PPSN). The final authenticated version is available online at: https://doi.org/[DOI not available]

Direct Manipulation of Free Form Deformation in Evolutionary Design Optimisation

Stefan Menzel, Markus Olhofer, and Bernhard Sendhoff

Honda Research Institute Europe GmbH,
Carl-Legien-Str. 30, D-63073 Offenbach/Main, Germany
{stefan.menzel, markus.olhofer, bs}@honda-ri.de

Abstract. The choice of an adequate representation plays a major role in any kind of evolutionary design optimisation. A flexible and preferably adaptive description of the shape is advantageous in many aspects as it guarantees a wide variety of geometry changes while keeping a low number of parameters. In the past free form deformation methods which are well known from the field of computer graphics have already been combined successfully with evolutionary optimisation. In the present paper general free form deformation (FFD) techniques are compared to the so-called direct manipulation (DM) of free form deformations method which promises several advantages in a design optimisation of complex systems. A general optimisation framework which allows the combination of these representations with evolutionary algorithms is described in detail. Its applicability is illustrated in a turbine blade test scenario to analyze the effects of the representations in evolutionary design optimisation.

1 Introduction

A crucial step when applying Evolutionary Algorithms to design optimisation problems is the definition of a representation of the design to be optimised by a set of parameter. Since the representation is highly problem specific various representations have been proposed for different application areas. Splines probably represent the most commonly used method for shape descriptions. A set of control points is used to define a spline curve or a spline surface which represents the cross-section or the surface of a geometry. This representation has been successfully used predominantly for the optimisation of relatively simple structures like turbine blades [1], aircraft wings or heat exchangers [2]. A major drawback of spline representations is the fact, that a large number of parameters is required to represent very complex shapes, i.e., the complexity of the representation is directly related to the complexity of the design. Additionally, if computational fluid dynamics calculations are required to determine the quality of new shapes, new grids have to be calculated. While automatic grid generation is possible for simple shapes, it is difficult or sometimes even impossible for complex designs. For these cases, the generation of the computational grid is a difficult and time consuming manual process. In particular, in the context of evolutionary algorithms, manual grid generation does not seem to be feasible.

An alternative representation that alleviates both problems (complex shapes and grid generation) is the free form deformation (FFD) method. In the FFD method

T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 352 – 361, 2006.

deformations of an initial design are described instead of the geometry itself. Therefore, the number of parameters is independent of the complexity of the shape. It is solely determined by the required flexibility of the deformation. FFD techniques have been introduced in the field of computer graphics and computer animation for object manipulation [3]. They have been applied to shape optimisation of aerodynamic problems by Perry et al. in [5] and together with evolutionary algorithms by Menzel et al. in [6]. It has been shown that FFD methods realize a good trade-off between shape flexibility and a low number of parameter which in turn results in a low dimensional search space for the optimisation. Furthermore, it has been demonstrated that the shape deformations defined by the FFD method can equally well be applied to deform a grid for computational fluid dynamics calculations. This way it is possible to omit the manual grid generation even for complex geometries. In many cases, design optimisation of complex shapes only becomes feasible when FFD methods are used for the representation.

An FFD system, which is described in Section 2 in more detail, is generally defined by a lattice of control points. A modification of the control point positions results in a deformation of the geometry inside the control volume. For the optimisation it is important to minimize the number of parameters. This is achieved by a proper initialisation of the control volume because the number and position of control points determines the shape flexibility.

One important aspect is to maximize the influence of the control points on the shape by placing control points close to the sensitive regions of the geometry. For the search process it is advantageous to generate a set of control points with which geometrical variations can be realized without the need for correlated changes of control points. Furthermore, the transformation should allow changes that are sufficiently local.

Therefore, this step of constructing an optimal control volume requires experience of the designer with the representation and more crucially with the problem at hand. Using an online adaptation of the grid of control points, which has been suggested in [6], can alleviate the problem of an optimal initial grid, however at the expense of an additional burden on the search process that is likely to result in additional evaluations and increased computing time.

Whereas the first problem addresses the dimensionality and flexibility of the representation, the second problem emerges purely due to the position of the control points in the design space. The position determines the structure of the search space and influences the optimisation process.

For both the dynamics of the search process and the design freedom, it would be desirable to *directly* specify shape variations instead of changing positions of control points in a grid that result in deformations of the "attached" shape as it is the case in standard FFD.

Direct Manipulation of Free Form Deformation (DMFFD) allows this. DMFFD decouples the control point grid of the FFD from the design modifications. Instead of defining the parameter of the transformation function, a set of "handles" to the design – usually surface points – are defined. The effect is a well defined influence of parameter on the shape. Furthermore, the search dimension is independent of the control volume, keeping all other attributes of the underlying free form deformation.

In this paper, we apply both methods, the standard free form deformation and the direct manipulation of Free Form Deformation, to a design optimization problem and compare the results with respect to quality, stability and convergence time. The optimization of a 2D transonic gas turbine stator blade does not represent the complex shape problem for which the FFD methods are actually most suitable. However, it indicates the strong and weak points of both representations and serves as a kind of intermediate benchmark problem.

The remainder of this paper is organized as follows. In Section 2, FFD and DMFFD are described in more detail. The combination of both representations with evolutionary computation is illustrated in the turbine blade test scenario in Section 3 and the effects of the representations are discussed. We conclude in Section 4.

2 Direct Manipulation of Free Form Deformations

2.1 Free Form Deformation (FFD)

The basic idea of the free form deformation is depicted in Fig. 1 a). The sphere represents the object which is the target of the optimisation. It is embedded in a lattice of control points (CP). Firstly, the coordinates of the object have to be mapped to the coordinates in the spline parameter space, a procedure which is called 'freezing'. If the object is a surface point cloud of the design or a mesh which originates from an aerodynamic computer simulation (as in our example in Section 3) each grid point has to be converted into spline parameter space to allow the deformations. For this calculation various methods have been proposed, e.g. Newton approximation or similar gradient based methods [4], [7]. After freezing the object can be modified by moving a control point to a new position. The new control point positions are the inputs for the spline equations and the updated geometry is calculated. Since *everything* in the control volume is deformed, a grid from computational fluid dynamics that is attached to the shape is also adapted. Hence, the deformation affects not only the shape of the design but at the same time the grid points of the computational mesh which is needed for the Computational Fluid Dynamics (CFD) evaluations of the proposed designs. The new shape and the corresponding CFD mesh are generated at the same time without the need for an automated or manual re-meshing procedure. This feature significantly reduces the computational costs and allows a high degree of automation [5], [6], [8]. Thus, by applying FFD the grid point coordinates are changed but the grid structure is kept.

As mentioned in the introduction the main disadvantage is the sensitivity of the FFD method to the initial placement of the control points. An inappropriate set-up increases the necessary size of the parameter set and therefore the dimensionality of the search space. One of the reasons is for example that the influence of a control point on an object decreases with the distance from the object. Even a small object variation requires a large modification of the control point if the initial distance between object and control point is large (this also violates the strong causality condition that is important in particular for Evolution Strategies). This in turn often modifies other areas of the design space which has to be compensated for by the movement of other control points. Hence, often correlated mutations of control points are necessary for a local change of the object geometry.

To reduce the influence of the initial positions of the control points direct manipulation is introduced as a representation for evolutionary optimisation which

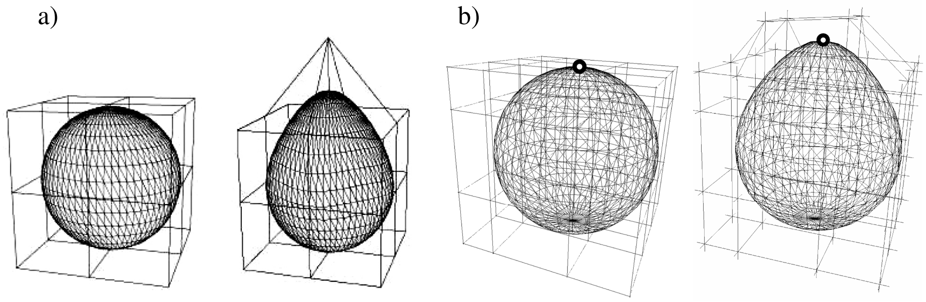

Fig. 1. a) Free Form Deformation [5]. The design is embedded within a lattice of control points. The modification of control points affects the shape as well as everything else inside the control volume. – b) Direct Manipulation of Free Form Deformations. The object point is chosen directly on the surface and the required movements of the control points to realize the target movement of the object point are calculated e.g. by the least squares method. The dotted control volume is invisible to the designer as s/he works directly on the object points; the control volume can be chosen arbitrarily.

allows to determine variations directly on the shape. Therefore local deformations of the object depend only on the so called object point.

2.2 Direct Manipulation of Free Form Deformations

Direct manipulation of free form deformations as an extension to the standard FFD has been introduced in [9]. Instead of moving control points (CP), whose influence on the shape is not always intuitive, the designer is encouraged to modify the shape directly by specifying so called object points (OP).

Although the initial setup of the control volume is similar to FFD, the control volume becomes invisible to the user and necessary correlated modifications are calculated analytically. In a first step, a lattice of control points has to be constructed and the coordinates of the object and the CFD mesh have to be frozen. But the control volume can be arbitrary, i.e., the number and positions of control points do not need to have any logical relationship to the embedded object, besides the fact that the number of control points determines the degree of freedom for the modification. In the next step, the designer specifies object points, which define handles to the represented object that can be repositioned. The shape is modified by *directly* changing the positions of these object points. The control points are determined analytically so that the shape variations (induced by the object point variations) are realized by the deformations associated with the new control point positions. In other words, the control points are calculated in such a way that the object points meet the given new positions under the constraint of minimal movement of the control points in a least square sense. Of course the object variations must be realizable by the deformations from the calculated new control point positions, i.e., if the number of control points is too small, some variations given by new object point positions might not be representable by a deformation.

In Fig. 1 b) an object point has been specified at the top of the sphere. The designer is able to move this object point upwards without any knowledge of the “underlying” control volume which can be initialized arbitrarily. The direct manipulation algorithm calculates the corresponding positions of the control points to mimic the targeted object point movement. The solution is shown in Fig. 1 b).

Direct manipulation of free form deformation has several advantages when combined with evolutionary optimization as compared to standard FFD. The construction of the control volume and the number and distribution of control points are not as important as in standard FFD. Furthermore, the number of optimisation parameters equals the number of object points.

3 FFD and Direct Manipulation of FFD in Evolutionary Design Optimisation

3.1 General Remarks on the Optimisation Set-Up

For a comparison of both representations a design optimisation test scenario has been set up. Four optimisation runs have been executed, one based on the standard free form deformation method and three using the direct manipulation technique.

All optimisations were based on an evolutionary strategy with covariance matrix adaptation (CMA-ES), an algorithm which combines fast convergence (few function evaluations) with high performance and small population sizes. This is especially significant for optimisations in which CFD calculations are required for fitness evaluation. There are mainly three features of the CMA-ES which distinguish it from standard evolutionary strategy algorithms. 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. 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 [1, 9, 10].

In all four optimisations the population size has been set to 32 individuals and an approximation model has been applied. In a pre-evaluation step all 32 individuals have been evaluated with a neural network and only the 16 most promising ones have been simulated with CFD to determine the individual fitness. The fitness values have also been used to train the neural network. From the 16 CFD results the best individual has been selected and considered as the parent for the next generation, so a (1,32(16))-strategy has been applied.

The details for each run can be found in Table 1. The number of parameters refers to the dimension of the search space. Their distribution on the design is depicted in Figure 3. The number of control points refers to the control point coordinates which can be modified in the FFD control volume. This is different from the total number of control point coordinates because points at the upper, lower and left border have to be constant due to CFD mesh consistency. Additionally, control points on the right edge of the control volume can be modified only in y-direction in order to fix the x-length of the design. For run 1 to run 3 the same FFD control mesh is used which is shown in Figure 3 in the upper left part for run 1.

Table 1. Type and number of parameters

Run	Type	Number of parameters	Number of control points
1	control points	10	10
2	object points	5	10
3	object points	13	10
4	object points	13	36

3.2 Turbine Blade Test Scenario

In this section, the results of a study on a turbine blade optimisation are presented, for details of the aerodynamic application and its parameters the reader is referred to [1].

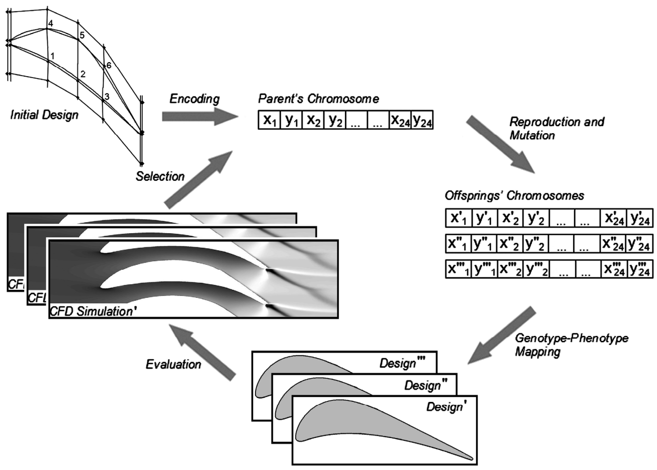

Fig. 2. The generation cycle in evolutionary design optimisation, see [6]

In Figure 2, the general workflow of the design optimisation is depicted and is now explained in more detail for run1, i.e., the standard free form deformation technique. A control volume consisting of 4x4 control points has been set up in which the turbine blade is embedded, see Figure 3. For easier visualization the CFD mesh is not plotted. However, we should keep in mind that during the deformation step the blade geometry *as well as* the CFD mesh are modified which allows the omission of the costly re-meshing process.

The control points CP_1-CP_4 can be freely moved in the x-y plane during the optimisation, while CP_5 and CP_6 are only allowed to move in vertical direction as stated above. After the encoding of these parameters (x and y coordinates of points CP_1 to CP_4 and the y-coordinate of CP_5 and CP_6) in the chromosome of the parent individual the control point positions are optimized. This includes the mutation of the control points, the deformation of the CFD grid based on the free form deformation algorithm and the execution of the CFD flow solver. As described in the previous section, the ES-CMA is used together with a neural network meta-model.

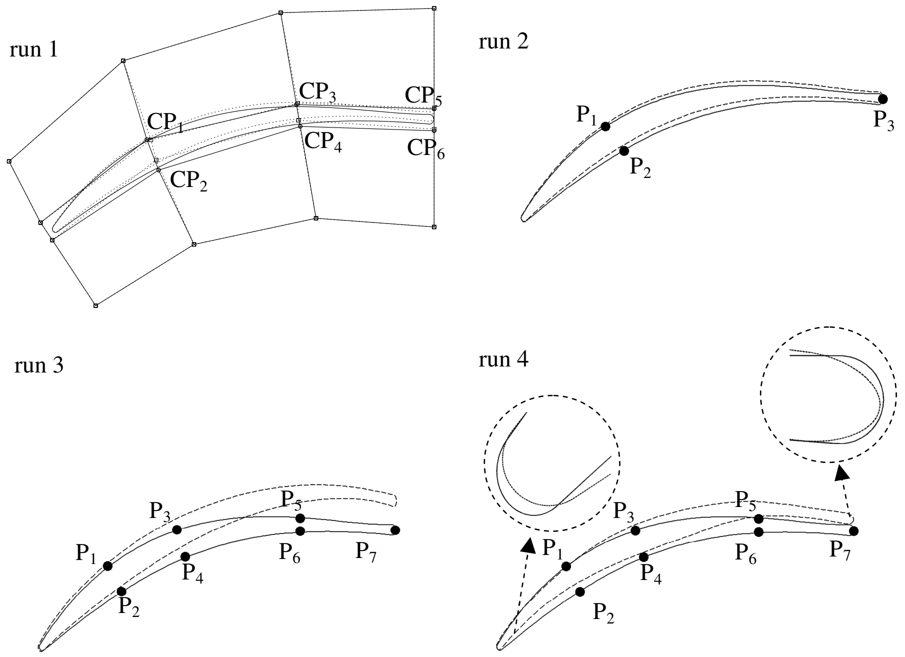

Fig. 3. Number and distribution of optimisation parameters. run1: 10 parameters (P_1-P_4: x, y; P_5, P_6: y); run2: 5 parameters (P_1, P_2: x, y; P_3: y); run3: 13 parameters (P_1-P_6: x, y; P_7: y); run4: 13 parameters (P_1-P_6: x, y; P_7: y). The closed line marks the initial designs, the dashed lines the optimized ones. The control volume is only drawn for run1. The control volumes for run2 and run3 are the same as for run1. Run4 has been modified in such a way that two rows and columns of control points have been inserted corresponding to a simple knot insertion algorithms as explained in [7]. For run4 the modifications at the leading and trailing edge are shown in a higher resolution to illustrate the occurring deformations. Initial circular or ellipsoid arcs are not kept after deformation because they turn out to be inferior to other leading and trailing edge geometries.

Run 2, 3 and 4 are identical besides that the direct manipulation of free form deformations is applied to modify the control points directly. The two major differences are the following:

1. The chromosomes contain object point positions (P_i) instead of control point positions (CP_i) as parameter sets.
2. The control points are calculated based on the encoded object points with the method for direct manipulation. Here the object points given in Figure 3 are used in the three runs. Based on the calculated control points the deformations of the design and CFD grid are updated in the same way as in run 1.

The results, i.e. the fitness progression, of this optimisation of all optimisation runs are summarized in Figure 4.

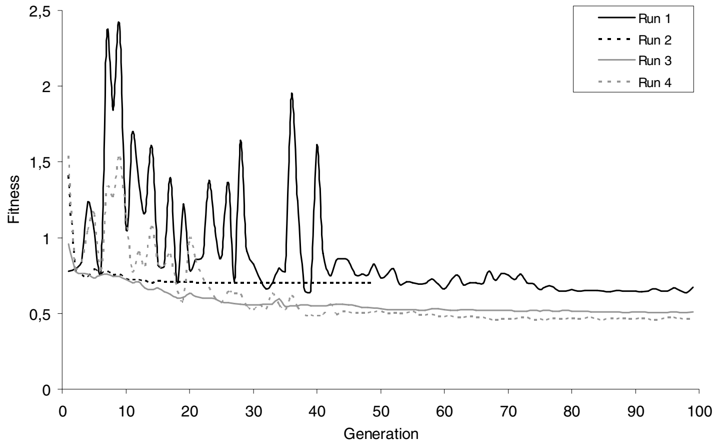

Fig. 4. The progress of the fitness of the runs 1 - 4

One major drawback of the direct manipulation method as presented in Section 2 is that the calculation of the control points is not unique in every case and constraints necessary for the deformation of the CFD grid are neglected. Especially negative volumes can emerge which can be described as loops in the design space.

This constraint is usually fulfilled by keeping the order of control points during the deformation step. However, when using direct manipulation the effect can occur that a targeted movement of an object point can only be achieved if large control point modifications are applied. These modifications can result in a change of the order of control points with the effect of producing grid cells with negative volumes. Methods for repairing and improving the structure of control points are therefore topic of our current research. To guarantee valid CFD meshes in the present optimisations the order of control points is checked after every mutation step. If the order of the control points is changed the mutation is repeated until a valid individual is generated.

3.3 Experimental Results

Figure 4 summarizes the results of the optimisations. Base run 1 that uses the standard FFD representation resulted in a converged fitness of 0.62 which means a 37% gain compared to the fitness of the initial turbine blade of 0.98.

According to Figure 3 three object points have been chosen for run 2. It resulted in a fitness of 0.7 but it needed less than half the number of generations and the optimization run is very stable. This is due to the reduced number of parameter which is only 5 (2 object points movable in x- and y-, 1 object point movable in y-direction) but it also shows that the flexibility of the design is limited by the choice of object points. This demonstrates that an optimisation using direct manipulation is limited by two factors. On the one hand a low number of object points restricts the flexibility of the design because these are the parameters which are optimized. On the other hand the number of control points limits the degree of realizable shape variations because the control points *actually* induce the targeted object point modifications through the defined deformations. If the number of control points is too low the targeted object points movements cannot be achieved.

In run 3 the number of object points (OP) has been increased to 7, i.e. 13 optimisation parameters (6 OP movable in x- and y-, 1 OP movable in y-direction) to improve the flexibility of the design. The fitness decreased to 0.5. This is an improvement compared to the optimisation run 1. This improvement is particularly interesting because the optimisation is based on the same control point grid as run 1. Even if the number of parameter for the optimisation is larger in run 3 the parameter for the deformations are identical because they are limited by the control point grid. Since the number and distribution of control points did not change between both runs the optimisation of run 1 must have converged to a local optimum. The structure of the search space seems to be changed by the direct manipulation in a way that the local maximum is circumvented in this optimisation run.

As a consequence, for this optimisation it can be seen that the usage of object points has been more successful. The fitness decreased faster and also at an earlier generation which is particularly important when dealing with time consuming evaluation functions like CFD simulations.

To analyze whether the performance could be even more increased by allowing more flexibility in the possible deformations two rows and columns of control points have been inserted into the control volume, resulting in 36 control points in run 4 while the number of object points was kept at 7. The fitness improved due to the control point insertion only slightly to 0.45. This is also a promising observation because the number of optimisation parameters is still 13 and the course of the fitness is quite similar to the one of run 3. Hence, the increase of flexibility by control point insertion did not affect the convergence behaviour.

4 Conclusions

In this paper, the standard free form deformation technique as well as direct manipulation of free form deformation have been combined with evolutionary optimisation. Both representations provide a fair trade-off between a low number of parameter and shape flexibility. To compare the performance a test scenario has been set up: a two dimensional turbine blade optimisation. The test scenario is a trade-off between a simple shape approximation benchmark problem and a truly complex shape optimization which is too computationally expensive for the comparison presented here. The expensive computational CFD simulations did not allow to perform several optimisations for averaging. By changing the representation and applying different numbers of object points the effect of the representation on the design optimisation has been studied.

In summary, we have shown that in this optimisation the usage of the new direct manipulation with free form deformation method has been advantageous.

If only 3 object points are chosen like in run 2 the convergence speed improved drastically and resulted still in a good performance index compared to the optimisation of the control points in run 1. This can be explained by the lower number of parameters in the optimisation. If the number of object points is increased like in run 3 and at the same time keeping the control points fixed, the fitness can be further improved although the possible transformations are kept constant in all three experiments. Here obviously the re-structuring of the search space by the introduction of the direct manipulation methods is beneficial.

Even an increase of control points in the control volume as it has been done in run 4 did not slow down the optimisation. This is a very promising result since the influence of the number of control points did not affect the convergence speed but the number of object points did. As a consequence one could argue to choose a high number of control points in the optimisation to achieve a high flexibility of the transformation and less constraints for the modification due to restrictions in the transformation. This definitely decreases the effect of the control point position and reduced the necessary prior knowledge about the optimisation problem while setting up the control volume.

Therefore, direct manipulation is a very promising representation because it provides the advantages of the standard free form deformation techniques while adding its own advantages as the arbitrary initialization of control volumes as well as the more direct impact of the object points on the geometry which improves the efficiency of the evolutionary algorithm.

References

[1] Sonoda, T., Yamaguchi, Y., Arima, T., Olhofer, M., Sendhoff, B., Schreiber, H-A.: Advanced High Turning Compressor Airfoils for Low Reynolds Number Condition, Part 1: Design and Optimization. Journal of Turbomachinery, 126: 350-359, 2004

[2] Foli, K., Okabe, T., Olhofer, M., Jin, Y., Sendhoff, B.: Optimization of Micro Heat Exchanger: CFD, Analytical Results and Multi-objective Evolutionary Algorithms. International Journal of Heat and Mass Transfer, vol. 49, no 5-6, pp. 1090-1099, 2006

[3] Sederberg, T. W., Parry, S. R.: Free-Form Deformation of Solid Geometric Models. Computer Graphics, 20(4):151-160, 1986

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

[5] Perry, E. C., Benzley, S. E., Landon, M., Johnson, R.: Shape Optimization of Fluid Flow Systems. Proceedings of ASME FEDSM'00. ASME Fluids Engineering Summer Conference, Boston, 2000.

[6] Menzel, S., Olhofer, M., Sendhoff, B.: Application of Free Form Deformation Techniques in Evolutionary Design Optimisation. 6th World Congress on Multidisciplinary and Structural Optimisation, Rio de Janeiro, 2005

[7] Piegl, L., Tiller, W.: The NURBS Book. Springer-Verlag Berlin Heidelberg 1995 and 1997

[8] Menzel, S., Olhofer, M., Sendhoff, B.: Evolutionary Design Optimisation of Complex Systems integrating Fluent for parallel Flow Evaluation. European Automotive CFD Conference, 2005

[9] Hsu, W. M., Hughes, J. F., Kaufman, H.: Direct Manipulations of Free-Form Deformations. Computer Graphics, 26:177-184, July 1992

[10] Jin, Y., Olhofer, M., Sendhoff, B.: Managing Approximate Models in Evolutionary Aerodynamic Design Optimization. Congress on Evolutionary Computation (CEC), vol. 2, pp. 592-599, 2001.

[11] Hansen, N., Ostermeier, A.: Completely Derandomized Self-adaptation in Evolution Strategies. Evolutionary Computation, vol. 9, no. 2, pp. 159-196, 2001