Coupled Evolutionary Shape Optimization and Reverse Engineering in Product Design and Virtual Prototyping

DAMIR VUČINA, ŽELJAN LOZINA, IGOR PEHNEC
FESB, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
University of Split,
R. Boskovica bb, 21000 Split,
CROATIA
vucina@fesb.hr, Web page: www.fesb.hr

Abstract: Product development in highly competitive times frequently implies very short time-to-market deadlines and product designs that need to deliver high performance at low investment and operation costs. These conflicting objectives can sometimes be accomplished by starting from existing designs and optimizing for better performance defined by some set of excellence criteria.
This paper develops a systematic procedure that starts by using a 3D scan of the existing design, defines an efficient shape parameterization for the part of the object that shall be redesigned, and develops an evolutionary shape optimization model. This is accompanied by developing a numerical workflow to handle this demanding process and an implementation that provides solutions in reasonable computer time. The workflow is developed based on existing specialized applications for geometric modeling, numerical analysis, finite-element simulations, and optimization, and therefore makes a rather heterogeneous system. It uses and runs different applications on the 'as-needed' basis and provides for corresponding synchronization and data mining. The procedure is in this paper applied to the optimum design of 2D airfoils for wind-turbine blades.

Key- Words: Shape optimization, reverse engineering, numerical workflow

1 Introduction

The ideal, somewhat utopistic objective of optimum engineering design is to design some engineering object 'from scratch' based on the technical specification of functionality only.

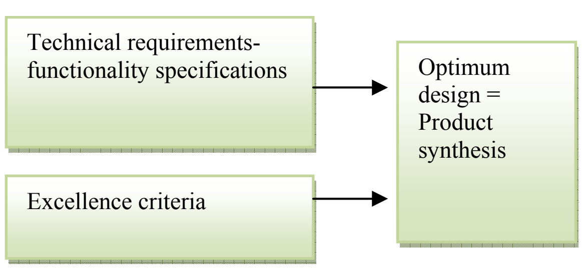


Fig.1, Idealized optimum design

In reality, the process in Fig.1 is not feasible directly, as many more intermediate steps are required. The problem in Fig.1 is characterized by multi-disciplinary, multi-objective and iterative elements and, by definition, belongs to inverse problems. An approach to this problem is developed in this paper.
The problem of optimum design [1], [2] is initially defined by the standard formulation,

$$\boldsymbol{x} = [\, x_1 \; x_2 \; .. \; x_n \,]^T \tag{1}$$

$$min \; \{ \boldsymbol{f}_i(\boldsymbol{x}) \} \quad , \; i = 1, k \tag{2}$$

$$g_j(\boldsymbol{x}) \leq 0 \quad , \; j = 1, p \tag{3}$$

$$h_j(\boldsymbol{x}) = 0 \quad , \; j = 1, r \tag{4}$$

with the optimization variables $\boldsymbol{x}$, k objective functions f, p inequality constraints g and r equality constraints h. For shape optimization problems, the variables in (1) are parameters that define the geometric shape of the object to be designed,

objective functions in (2) are shape- dependent excellence criteria, and (3) and (4) define technical requirements, more specifically constraints on shape related to required functionality.
Since the objectives and requirements depend on shape (1)-(4) which continuously changes during the course of optimization, efficient parameterization of shape is needed to provide for global and local representation of geometry of 2D or 3D objects with as few variables as possible.

Design optimization generally includes three building blocks [3], [4], [5]: topology optimization, shape optimization and dimensional optimization. While dimensional optimization is a mature process that found its way into the industry, shape and topological optimization are still avant-garde technologies. Unlike dimensional optimization where the approach is more or less just a matter of varying selected parameters, shape and topology optimization don't couple easily with standard simulation tools in the industry. The reason for this is the change in geometry and even topology which causes problems in communicating with FE-analysis and CAD packages.

This paper focuses on applying evolutionary optimization algorithms to multi-objective shape optimization, [6]- [10]. In particular, reverse engineering with optimization in the sense of product redesign (re-engineering) is considered here. The process is set-up to consist of high- density 3D-scanning of the object geometry, post-processing and parameterization of the geometric database, transforming the large geometric shape record into a small but sufficient set of shape variables, and finally shape optimization to achieve an improved design. The re-engineered design is expected to deliver better performance due to the improved shape of parts of the object, and geometric compatibility to other components of the system since parts of the object maintain their 'frozen' scanned geometry.

Different concepts of 2D and 3D geometry parameterizations can be found in the literature, [11], [12]. They will not be discussed here, but typically they include point sets outlining contours of objects, parametric mathematical surfaces, superposition of component shapes, parametric CAD formats with feature-based solid modeling [13]- [16], and other. None of those are perfect in terms of providing: compact sets of shape variables, easy data exchange with CAD and FEA packages, superior performance in geometric modeling, global and local shape control, detection of inconsistent geometries (invalid designs can potentially be derived in terms of: topology, shape, dimensions, constraints, mutual interference).

The need for efficient shape optimization is evident from the large variety of applications in different fields, which include airfoils and blades, machine elements, automotive body components, yacht component shape, path routing, die shape, casting shape, tool design, shape detection with inverse problems, and many other.

2 Numerical Procedure

The process will be designed in such a way that the evolutionary optimizer steers the values of optimization variables (1).

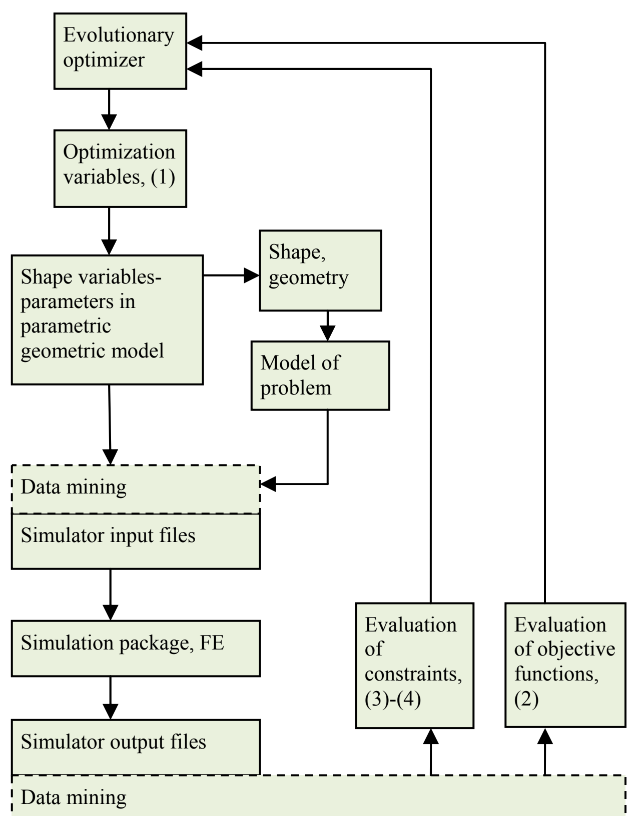


Fig.2, Shape optimization- numerical process

These variables are linked to the shape parameters, i.e. control points of parametric curves and surfaces in the geometric model of the object.
The shape optimization will be undertaken by applying genetic algorithms [6], [17]- [19]. Verification of the excellence criteria (2) and constraint conditions- technical requirements (3), (4) is provided for by initiating FE simulations from within the optimizer by launching the corresponding package, in this case the ADINA software [20], as shown in Fig.2. Depending on the particular application, evolutionary optimization can also be used with full value or cost fitness models [21], combined with gradient NLP search [22] methods, and combined with re-engineering based on local knowledge [23].

Communication and exchange of current data between the optimizer and the simulator applications is done via corresponding input and output files by applying relative data mining.
When evolutionary optimization is applied to the parameterized shape of the object, populations of virtual beings representing the 'species' of those objects will undergo virtual evolution process and be subject to operators of selection, cross-over, mutation and other.

Reverse engineering typically implies 3D scanning of an existing object in order to produce the corresponding CAD model. In this paper, the idea is to apply reverse re-engineering or re-design as part of the process. The objective is to scan an object using a high-resolution and high-accuracy optical 3D sensor, parameterize the geometry and subject it to re-design in such a way that it remains geometrically compatible with the system it belongs to while changing parts of it to improve performance. This implies that portions of the parameterized geometric model will remain 'frozen' while the shape of other portions will be numerically optimized using shape parameters as optimization variables.

The process starts with a 3D scan of the object, as shown in Fig.3. Our lab uses the instrument ATOS, an optical system based on the principle of triangulation, where fringe patterns are projected on the object's surface and recorded stereophotogrametrically. 3D coordinates for each camera pixel (two cameras) are calculated in high precision, and the points cloud and polygon mesh of the surface of the object are generated. The measurement resolution is 800k points, a single scan is complete in 0.8 seconds, and multiple scans are aligned by means of reference points which allows for moving of the camera and object between the individual scans and also superposition of individual (multi-view) scans into the overall geometric database.

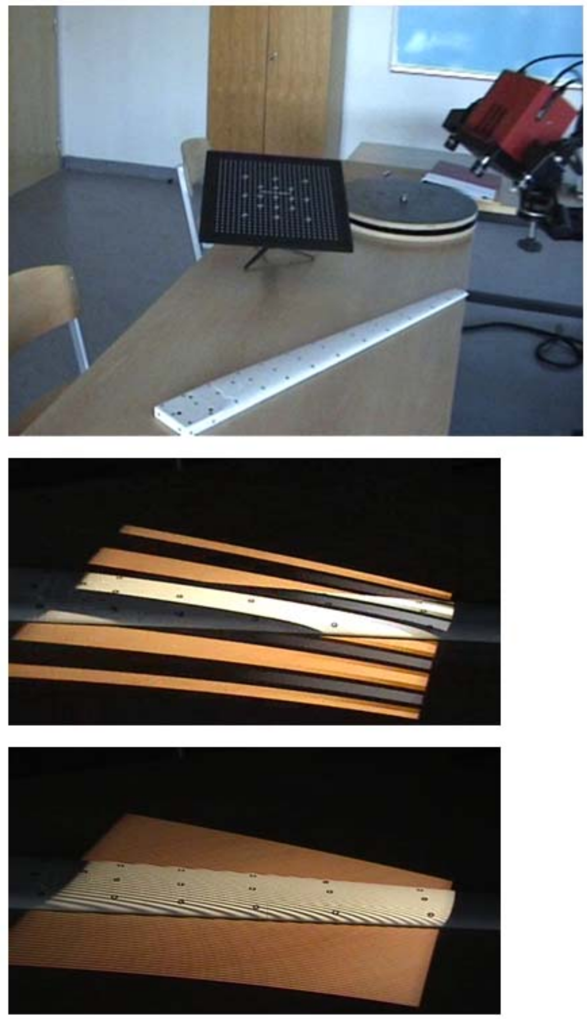


Fig.3, Optical 3D scan of the object

This huge geometric database is not directly suitable for the purpose of re-engineering. It is therefore necessary to transform the point cloud into parametric geometric objects, such as NURBS. The huge amount of scanned geometric data is represented by mathematical forms which are controlled by a small number of parameters, for example control points. A variety of commercial software is available for this process.

A full-scale reverse re-engineering is not yet implemented at this point. Instead, in the case of this paper, a simplified procedure is applied due to limited resources and different objectives of the paper. Instead of automatic parameterization into NURBS by an off-the-shelf application, we have developed a parameterization procedure based on chained Bezier curves and chained Bezier surfaces, [24]. Although in-house code is typically computationally less efficient, our parameterization allows for more flexibility and control, and is therefore well suited for the process in Fig.2.
In this paper, instead of applying the procedure to the full 3D geometry, a simpler and less numerically intensive application to 2D airfoils is presented here.

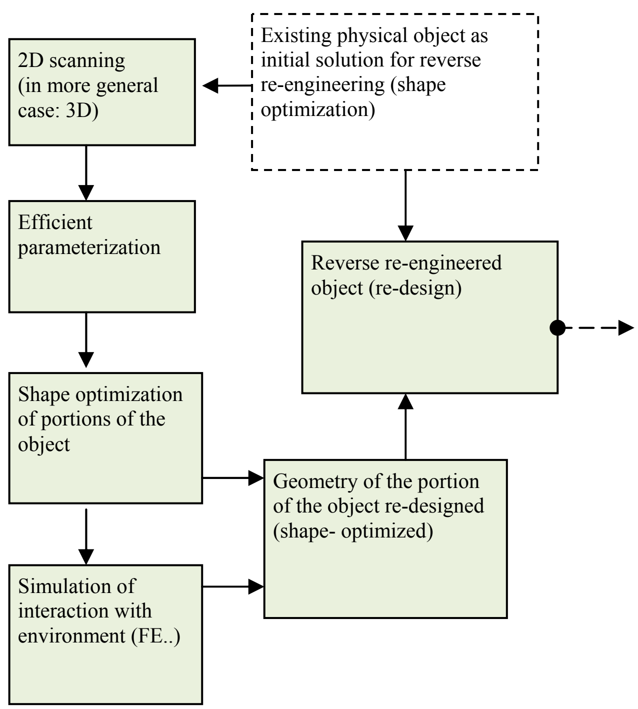


Fig.4, Reverse re-engineering, simple approach

The communication between the optimizer and the geometric modeling application can in general be achieved by using a dedicated program interface that makes it possible to make changes in the native format of the parametric modeling application and hence have the integrity of design be maintained by that application directly. Alternatively, standardized graphics exchange formats such as IGES could be used, in which case program scripts are necessary to propagate the changes to all entities affected and preserve overall integrity. IGES essentially keeps sequential records of individual graphic objects in text format, whereby standard individual entities-primitives are defined by the respective types of parameters and their structure, their respective numerical values and other graphic properties, in a variety of possible ways.

Finally, the process in Figs.1 and 2 is numerically very intensive, for a number of reasons. They include a relatively high number of design variables, changing geometry of the object domain under consideration, and relatively high workload on the simulator which in this case requires updates of the geometric definition of the object and frequently re-meshing in the FE discretization of the domain. It is therefore highly desirable to make the workflow layout parallelized in terms of the simulator calculations, which makes it possible to invoke many instances of the simulator with different candidate-solutions simultaneously.

3 Problem modeling

The contours- boundaries of the 2D object are in this paper represented by Bezier curves,

$$\mathbf{P}(u) = \sum_{i=0}^{n} \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \cdot \mathbf{P}_i \tag{5}$$

and by combining them into Bezier surfaces for 3D,

$$\mathbf{P}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u) \cdot B_{j,m}(v) \cdot \mathbf{P}_{i,j} \quad , \quad u,v \in (0,1) \tag{6}$$

where $\boldsymbol{P}_i$ and $\boldsymbol{P}_{i,j}$ are the control nodes and B the corresponding Bernstein polynomials. In order to have low-degree Bezier curves and/or surfaces and simultaneously be able to use a sufficient number of control points to represent complex shapes, a procedure for chaining piecewise Bezier curves and surfaces was developed. It consists of using piecewise curves/ surfaces and generating additional (dependent) control points in segments where the piecewise curves/ surfaces meet, in such a way that C^1 continuity is provided for and automatically imposed. As an example, a wind-turbine blade represented by a set of 2 x 2 chained surfaces is shown in Fig.5.

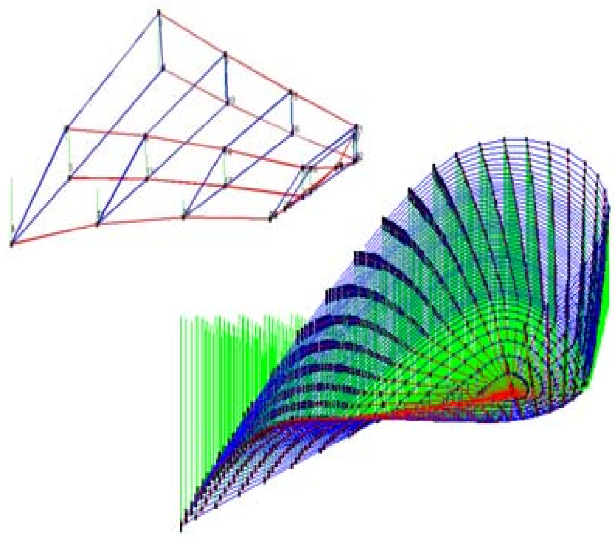


Fig.5, 2 x 2 chained Bezier patches representing the wind-turbine blade

In this paper, the in-house developed adaptive procedure [24] based on chaining piecewise Bezier curves (5) with C^1 continuity is applied, with

$$\boldsymbol{P}_{gen} = C(\boldsymbol{P}_l, \boldsymbol{P}_r) \tag{7}$$

where $\boldsymbol{P}_{gen}$ are points generated in the segment between $\boldsymbol{P}_l$ and $\boldsymbol{P}_r$ to impose the requested C^1 or higher continuity, and C is the operator that interpolates $\boldsymbol{P}_{gen}$.

Other parameterization concepts are also applied where instead of Bezier curves other basis curves can be chained. Firstly, cubic splines (Fig. 6) can be used.

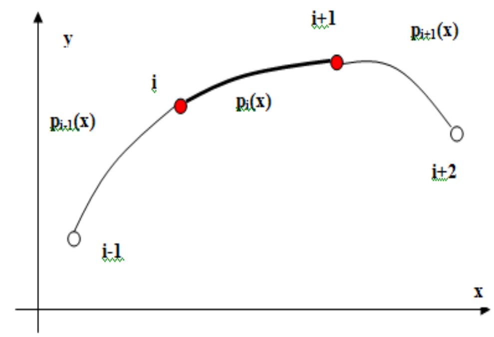


Fig.6, Cubic splines for 2D shape parameterization

Essentially, piecewise cubic polynomials defined in their respective segments are chained with C^2 continuity (curve itself, slope, curvature)

$$\left.\begin{array}{l} p_i(x_i) = y_i \\ p_i(x_{i+1}) = y_{i+1} \\ p'_i(x_{i+1}) = p'_{i+1}(x_{i+1}) \\ p''_i(x_{i+1}) = p''_{i+1}(x_{i+1}) \end{array}\right\} \tag{8}$$

The cubic piecewise polynomials are interpolated through a sequence of nodal points which represent the shape variables. Unfortunately, the splines do not possess locality properties, but this does not deteriorate the numerical efficiency much since there are typically few segments.

Yet another option is using low-degree polynomials and then blending them instead of chaining. This approach is demonstrated in Figure 7,

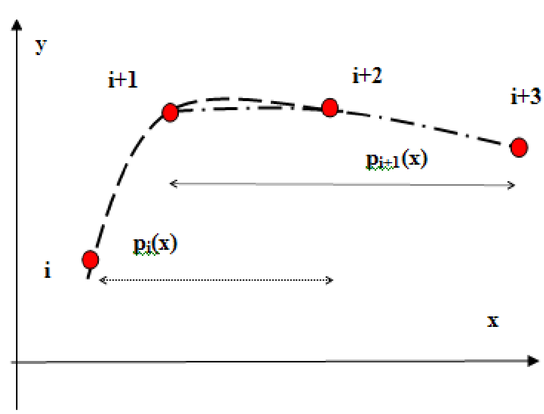


Fig.7, Curve blending for 2D shape parameterization

In this case, for example with 2nd degree polynomials

$$p_i(x) = a_i + b_i x + c_i x^2 \tag{9}$$

can be blended in the segment $(x_{i+1} < x < x_{i+2})$ where linear or other blending can be applied,

$$p(x) = (1-t)\cdot p_i(x) + t\cdot p_{i+1}(x)$$
$$t = (x - x_{i+1})/(x_{i+2} - x_{i+1}) \qquad t \in (0,1) \tag{10}$$

These parameterization procedures essentially act as lossy data compression procedures for geometric information defining 3D shape. A large number of 3D point coordinates obtained for example as a points cloud in the procedure of optical 3D scanning (Fig.3) can be reduced to a small set of control points by the procedure shown in Fig.4. Those control points can now act as optimization variables in the shape optimization process, Fig.4. The question remains how many control points should be used in the procedure. It is a obviously trade-off situation and the decision can be left to the analyst for each particular object, although it can also be automatized numerically, provided criteria for representation requirements are defined. Too many control points increase the dimensionality of the optimization search space in Fig.4 and (1), too few points do not provide sufficient design representation freedom to represent local changes in shape, i.e. have a filtering effect on the geometry.

A straight- forward criterion can be used when evaluating which parameterization to apply. Obviously, the objective is to be able to describe the shape with as few as possible geometric parameters, with satisfactory accuracy. Then the most acceptable parameterization would provide a satisfactory total square deviation (or average deviation) between the points cloud f_i of the scanned surface and the mathematical parametric surface p_i

$$S = S(a_j) = \sum_{i=1}^{m} (f_i - p_i)^2 \tag{11}$$

with the smallest number of parameters a_j, since this results in the smallest number of variables in the shape optimization search space.

4 Shape optimization of an airfoil

The 2D point cloud defining the airfoil contour, obtained by scanning (Fig.4), is given in the following table:

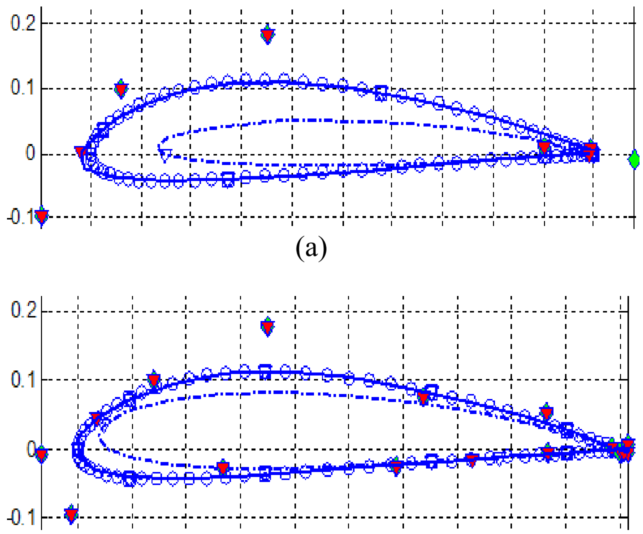
1	0
0.998459	0.00048
0.993844	0.001911
0.986185	0.004266
0.975528	0.007501
0.96194	0.011553
0.945503	0.01635
0.92632	0.021806
0.904508	0.027823
0.880203	0.034301
0.853553	0.04113
0.824724	0.0482
...	
0.761249	-0.011617
0.793893	-0.009878
0.824724	-0.008287
0.853553	-0.006843
0.880203	-0.005544
0.904508	-0.004385
0.92632	-0.003364
0.945503	-0.002478
0.96194	-0.001726
0.975528	-0.001108
0.986185	-0.000625
0.993844	-0.000278
0.998459	-0.00007
1	0

Table 1: 2D point cloud, scanning in Fig.4

Fig.8 demonstrates the application of the procedure to 2D airfoil geometry, which is convenient in this paper since the quantity of geometric data is not too large and can be handled more easily.

Several choices of free parameterization parameters were made, the number of chained curves and the degree of individual curves were varied.

Fig. 8 demonstrates the significance of efficient parameterization, it shows that only a few control points are sufficient and capable to represent the shape. To demonstrate the geometric capability of the shape representation, Fig.6 shows that the initial shapes shown could be drastically improved by optimizing the shape using the positions of respective control points as optimization variables, and best-fit least squares of deviations as the excellence criterion.

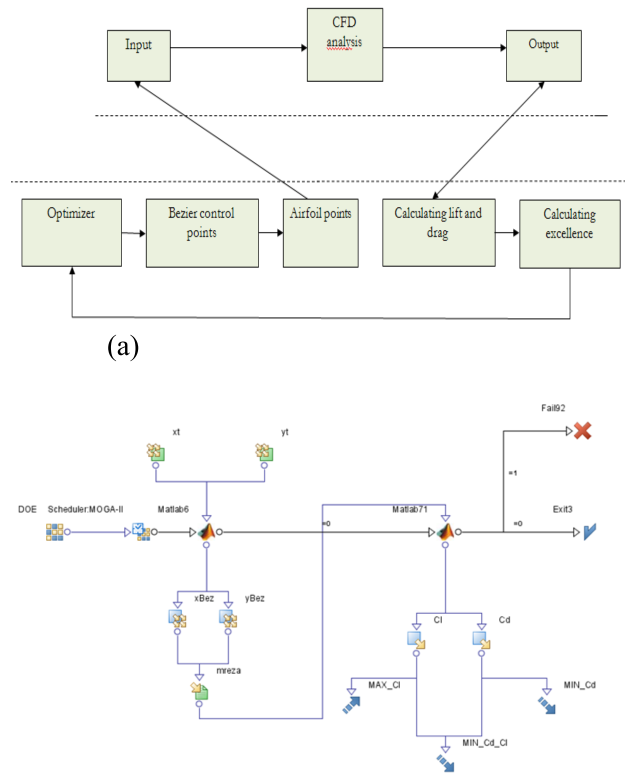


(a) (b)

Fig.8a and 8b, Parameterization of the airfoil contour based on given points cloud, (-.- initial shape, ___ fitted shape)

(a) 2 chained 3rd degree Bezier curves

(b) 5 chained 3rd degree Bezier curves



(a) (b)

Fig.9a, 9b, Shape optimization workflow

The next procedure referred to in Fig.4 is shape optimization of the airfoil using the control points structure of the selected parameterization as shape variables. The simulation of interaction with the environment, simulation of flow, is carried out using the ADINA-F finite element software [20], ModeFRONTIER optimizers [25], and in-house program scripts to control the workflow, Fig.9.

In the workflow developed here, the optimizer varies the shape variables (control points within constrained ranges) and communicates the updated geometry to the ADINA CFD simulator, which calculates the elementary forces on the airfoil. Piecewise curves are employed and continuity at joints is imposed by adding generated points. Initially, a population of 2D airfoils is generated randomly.

ADINA output files are read by an in-house developed script and, after data mining, the script evaluates the drag and lift which correspond to the current shape. The ADINA program calls and control scripts are encapsulated in the MATLAB nodes in the workflow. The program scripts are also in charge of updating the ADINA input files by applying corresponding data mining and writing, as well as generating boundary contours based on current control points. The data mining process is used to transfer the geometric data between the optimizer and ADINA. The batch-mode launch of ADINA and management of operations in ADINA (for example re-generating boundaries or re-meshing) from the calling MATLAB or C script are done using system calls such as “shell”, “sendkey”. The program scripts also generate delays between subsequent operations initiated on the other application, allowing them to complete.

Some simulation results are shown in Fig.10, where some of the successful and unsuccessful runs with corresponding shapes are presented.

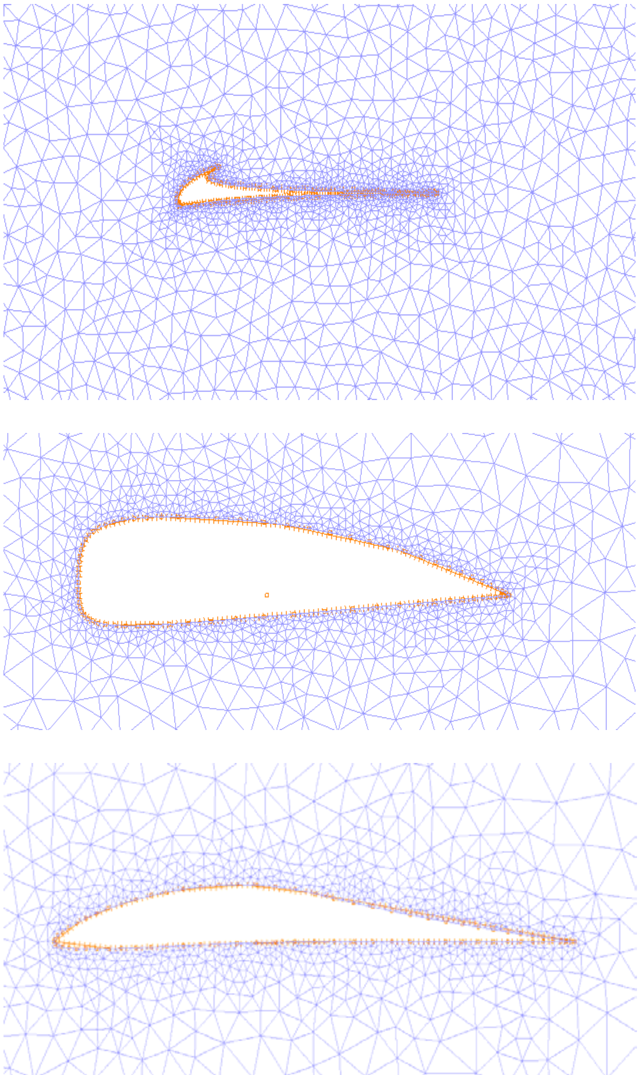


Fig.10, Some geometric shapes generated as candidate designs by the GA shape optimizer

Since an evolutionary optimizer is used, invalid geometric shapes are occasionally generated by the optimizer, which can cause ADINA to fail. This is detected by the calling script, by verifying the ADINA output file's existence and completeness. If there is no output file or there is an incomplete analysis file, the current ADINA run and the corresponding candidate design are dismissed. In some cases ADINA may even freeze due to invalid geometry, therefore the control script after a predefined time period triggers the rejection of the respective ADINA instance.

The approach in the developed workflow is specific in its very general layout and applicability. Any third-party simulation package (such as ADINA) can be integrated into the optimization environment since:

- no application-specific features such as return values are required
- the developed control & interface script uses only the application's output file to verify the completeness and validity of its execution
- data-mining in simulator I/O files is used for inter-application communication
- one application in the workflow (calling application, master) externally manages another application (called application, slave) via batch mode commands or via remotely simulated menu keystrokes
- the workflow layout is suitable for parallelization

The data mining is developed within the control scripts. Absolute position data mining is used both with input and output files of the FE simulator. Invalid shapes such as the one shown in Fig.10a. are detected either by non-existing FE simulator output files or recognized by their inappropriate size or structure. Such candidate designs are then simply disregarded and the corresponding instances of the FE simulator are dismissed.

Since evolutionary optimizers tend to be relatively slow and numerically inefficient if compared with the gradient algorithms of classical nonlinear programming, parallelization of the simulators was used in the subsequent workflows. Given the inherently parallel structure of the genetic algorithms, the numerical efficiency increases almost by the factor of the number of parallel computers running the FE simulators.

The parallel layout is outlined in Fig.11,

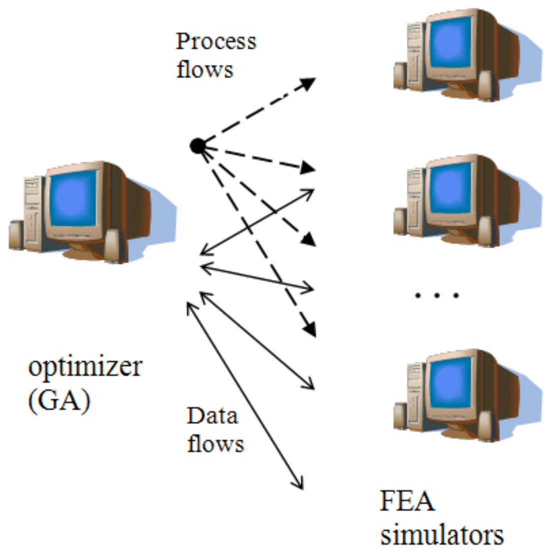


Fig.11: Parallel layout for shape optimization

This parallelization can be accomplished within the workflow under the assumption that a corresponding dedicated cluster is set-up.

The procedure developed in this paper can be compared to the numerically more efficient design sensitivity approach where the FE software is upgraded to additionally deliver sensitivity terms, derivatives with respect to shape variables $\boldsymbol{x}$,

$$K \cdot \frac{\partial u}{\partial x} = \frac{\partial F}{\partial x} - \frac{\partial K}{\partial x} \cdot u \qquad (12)$$

($\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{F}$ the load vector, $\boldsymbol{u}$ the displacements vector) in which case the pseudo-load on the right-hand side also requires the evaluation of derivatives of FE shape functions with respect to shape variables. Being a gradient-type formulation, this approach to shape optimization provides far better numerical efficiency. However, this formulation requires that the FE software be extended to provide the extra terms needed in (12). Moreover, the shape variables themselves are to be continuous and the stiffness matrices and loads are to be continuous functions of the shape variables.

On the other hand, the formulation and approach presented in this paper, while being far less numerically efficient, are also much less demanding. The variables can also be discrete, hybrid and discontinuous, and no changes are necessary in the standard FE software. Topology, shape and dimensional optimization can be combined and in some cases packaged in the same optimization variables vector and dealt with simultaneously. Convergence is ensured in the sense that the developed procedure is essentially a non-gradient search strategy that systematically scans the search space. Therefore, the decision which approach to select is an individual trade-off for the particular problem and depends on the problem at hand. The two approaches can also be combined, whereby the procedure presented here is very suitable for the early design stage, where there is still a lot of design freedom and only a very loose idea of an 'initial solution'.

4 Conclusion

A systematic general procedure for the optimized re-design of existing objects is presented. It includes 3D scanning of the existing object to import an initial solution for the geometry and 'freeze' the shape of sections which must not be changed. The rest of the object is re-engineered such that its shape is optimized for given excellence criteria and constraints. GA optimizers and FEA simulation tools are applied within the framework of a specifically developed custom workflow which uses inter-application calls, control scripts and data mining.

References:

[1] J. Arora: '*Introduction to Optimum Design*', McGraw-Hill, New York 1989.

[2] S.S. Rao, '*Engineering Optimization'*, Wiley Interscience, 1996

[3] M. Pourazady, Z. Fu, "An Integrated Approach To Structural Shape Optimization", *Computers &* Structures, Vol 60. No 2. pp. 279-289. 1996

[4] M. Zhou, N. Pagaldipti, H.L. Thomas, Y.K. Shyy, "An integrated approach to topology, sizing, and shape optimization", *Struct Multidisc Optim* 26, 308–317 (2004)
[5] K. Saitou, K. Izui, S. Nishiwaki, P. Papalambros, "A Survey of Structural Optimization in Mechanical Product Development". *Transactions of the ASME* 5, 214-226
[6] Goldberg D.E., *'Genetic Algorithms in Search, Optimization and Machine Learning*', Addison Wesley, 1989
[7] K. Deb and T. Goel. 'A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design', *First International Conference on Evolutionary Multi-Criterion Optimization*, 385-399. Springer-Verlag, Lecture Notes in Computer Science No. 1993, 2001
[8] K. Deb and T. Goyal. 'Multi-Objective Evolutionary Algorithms for Engineering Shape Design, *KanGAL report 200003*, Indian Institute of Technology, Kanpur, India, 2000
[9] D. Quagliarella and A. Vicini. 'GAs for Aerodynamic Shape Design II: Multiobjective Optimization and Multi-Criteria Design', in *Von Karman Institute Lecture Series 2000-07. Genetic Algorithms for Optimisation in Aeronautics and Turbomachinery*, May 2000.
[10] F. Cappello, A. Mancuso, "A genetic algorithm for combined topology and shape optimisations", *Computer-Aided Design* 35 (2003) 761–769
[11] Haftka, R.T.; Grandhi, R.V. 1986: 'Structural shape optimization– a survey', *Comp. Meth. Appl. Mech. Engrg.* 57, 1986, 91–106
[12] Samareh, J.A., 'A Survey Of Shape Parameterization Techniques', *CEAS/AIAA/ ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics* June 22-25, 1999/Williamsburg, VA,
[13] Lee, K., *'Principles of CAD/ CAM/ CAE Systems*', Addison Wesley, 1999
[14] Dai, L., Gu, Y., Zhao, G., Guo, Y., ' Structural Shape Optimization based on Parametric Dimension-Driving and CAD Software Integration', *6th World Congress on Structural and Multidisciplinary Optimization* Rio de Janeiro, 2005, Brazil
[15] G. Farin, '*Curves and Surfaces for Computer Aided Geometric Design (2nd edn)*'. Academic Press: New York, 1993.
[16] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics.* McGraw-Hill, New York (1976).
[17] M. Gen, R. Cheng, '*Genetic Algorithms and Engineering Optimization*', Wiley-Interscience, 2000
[18] T. Back, D.B. Fogel, T. Michalewicz, '*Evolutionary Computation 2, Advanced Algorithms and Operators*', Institute of Physics Publishing, 2000
[19] K. Deb, *'Multi-Objective Optimization using Evolutionary Algorithms*', Wiley and Sons, 2001
[20] Adina, *ADINA^{TM} 8.3 User interface and command reference manual*, Adina R&D Inc, 2005
[21] D.Vucina, Z. Lozina, 'Design Optimization with GA Fitness Functions based on Total Lifecycle Value or Cost', *7th WSEAS International Conference on EVOLUTIONARY COMPUTING (EC '06)*, Cavtat
[22] T.Y. Chen, Y.L. Cheng, 'Global Optimization Using Hybrid Approach', *7th WSEAS International Conference on Simulation, Modelling and Optimization*, Beijing, China, 2007
[23] M.A. Rosenman, N. Preema, 'Plastic Surgery and Genetic Re-Engineering in Evolutionary Design', *8th WSEAS International Conference on Evolutionary Computing*, Vancouver, 2007
[24] modeFRONTIER, www.esteco.com
[25] D. Vučina, Z. Lozina, I. Pehnec, 'A Compact Parameterization for Shape Optimization of Aerofoils', *International Conference of Computational Intelligence and Intelligent Systems*- London 2008