

UNIVERSITY OF SPLIT  
FACULTY OF ELECTRICAL ENGINEERING, MECHANICAL ENGINEERING  
AND NAVAL ARCHITECTURE

**Ivo Marinić-Kragić**

**EFFICIENT PARAMETERIZATION METHODS FOR  
GENERIC SHAPE OPTIMIZATION WITH  
ENGINEERING APPLICATIONS**

DOCTORAL THESIS

Split, 2017.



UNIVERSITY OF SPLIT  
FACULTY OF ELECTRICAL ENGINEERING, MECHANICAL ENGINEERING  
AND NAVAL ARCHITECTURE

**Ivo Marinić-Kragić**

***Efficient Parameterization Methods for Generic Shape  
Optimization with Engineering Applications***

DOCTORAL THESIS

Split, 2017.

The research reported in this thesis was carried out at Department of Mechanical Engineering and Naval Architecture, University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture.

Supervisor: Prof dr. sc. Damir Vučina, FESB, University of Split, Croatia

Disertation number: xxx-xxx

---

#### BIBLIOGRAPHIC INFORMATION

Keywords: shape optimization, engineering optimization, shape parameterization, B-spline, CFD

Scientific area (znanstveno područje): Technical science (Tehničke znanosti)

Scientific field (znanstveno polje): Fundamental technical science (Temeljne tehničke znanosti)

Scientific branch (znanstvena grana): Technical mechanics, mechanics of rigid and deformable bodies (Tehnička mehanika, mehanika krutih i deformabilnih tijela)

Institution of PhD completion: University of Split, Faculty of electrical engineering, mechanical engineering and naval architecture

Supervisor of the thesis: Prof dr. sc. Damir Vučina

Number of pages: 198

Number of figures: 30

Number of tables: 2

Number of references: 71

---

Committee for assessment of doctoral dissertation:

1. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
2. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
3. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
4. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
5. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town

Committee for defence of doctoral dissertation:

1. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
2. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
3. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
4. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town
5. XXXX. dr. sc. XXXX Yyyyy, Institution name and City/Town

Dissertation defended on: xx.month.20xx.

# Efficient Parameterization Methods for Generic Shape Optimization with Engineering Applications

## Abstract

Engineering optimization very often includes complex and computationally expensive numerical simulations with time requirements ranging from few seconds to months per single simulation, depending on the simulated physics and available computational resources. Such optimization tasks should therefore contain no more than a small, yet sufficient number of optimization variables. Additional difficulties arise when the underlying physics has nonlinear character modeled by nonlinear partial differential equations. The most common cases of engineering optimization involving computationally very expensive nonlinear simulation include problems in computational fluid dynamics. Furthermore, technical objects considered in fluid dynamics (ship hulls, wind-turbines or fan blades,...) are almost always complex three-dimensional shapes that cannot adequately be described with only a few shape variables. The most commonly used tools for generic shape representation are B-spline and NURBS curves and surfaces. When a shape is too complex to be represented by a single surface, mutually connected surface patches can be used. In recent years, a generalization of B-splines called T-splines is also increasingly used.

The first objective of this doctoral thesis is the development of genetic shape parameterization methods for solving selected engineering optimization problems. This includes integration of all needed elements (shape parameterization, numerical analysis and optimization) in an integrated numerical workflow. The second objective is the development of more universal advanced shape parameterization methods that can aid in the reduction of the overall time required for shape optimization.

**Keywords:** shape optimization, engineering optimization, shape parameterization, B-spline, CFD

# Metode učinkovite parametrizacije za generičko optimiranje oblika s inženjerskim primjenama

## Kratki sažetak

Inženjerska optimizacija često sadržava složene i računalno skupe numeričke simulacije s računalnim vremenskim zahtjevima od nekoliko sekundi do nekoliko mjeseci po simulaciji, ovisno o simuliranoj fizici i dostupnim računalnim resursima. Ovakvi optimizacijski zadaci stoga ne mogu sadržavati nego mali broj optimizacijskih varijabli. Dodatne teškoće nastaju kada je problem opisan nelinearnim parcijalnim diferencijalnim jednadžbama. Primjer inženjerske optimizacije s računski zahtjevnim simulacijama i nelinearnim modelima vrlo često uključuju računalnu dinamiku fluida. Uz sve navedeno, tehnički objekti u dinamici fluida (trup broda, lopatice vjetro-turbine ili ventilatora,...) često su kompleksni trodimenzionalni oblici koji se ne mogu adekvatno opisati s malim brojem varijabli oblika. Najčešće korišteni alati za zapis generičkih krivulja i ploha su B-spline i NURBS plohe. Kada je oblik previše složen da bi se opisao s pojedinačnom plohom, može zapisati i pomoću više ploha spojenih po dijelovima. Posljednjih godina se u te svrhe koristi i generalizacija B-spline, takozvani T-spline.

Prvi cilj ovog doktorskog rada je razvoj metoda generičke parametrizacije oblika za rješavanje odabranih problema inženjerske optimizacije. Ovo uključuje i razvoj numeričkih radnih tokova koji integriraju sve potrebne elemente (parametrizacija oblika, numerička analiza i optimizacija). Drugi dio doktorskog rada je razvoj naprednih metoda parametrizacije koje su primjenjive za opći slučaj parametrizacije oblika i omogućavaju ubrzanje sveukupnog procesa inženjerske optimizacije oblika.

**Ključne riječi:** optimizacija oblika, inženjerska optimizacija, parametrizacija, B-spline, računalna dinamika fluida

Blank page



# Contents

Abstract.....	iv
Kratki sažetak .....	v
Contents .....	vi
List of Figures.....	x
List of Tables .....	xii
Abbreviations.....	xiii
1. INTRODUCTION .....	1
1.1. Motivation and hypothesis .....	1
1.2. Research methodology and scientific contributions.....	3
1.3. List of complementary papers .....	5
1.4. Thesis overview.....	6
2. ELEMENTS OF NUMERICAL OPTIMIZATION WORKFLOW .....	7
2.1. Numerical optimization.....	7
2.1.1. Genetic algorithms .....	8
2.1.2. Sensitivity analysis .....	10
2.2. Shape parameterization .....	11
2.2.1. Bezier Patches .....	13
2.2.2. B-spline and NURBS .....	14
2.2.3. T-Splines .....	15
2.2.4. Multi-patch parameterizations .....	17
2.3. Computational fluid dynamics .....	18
2.3.1. Fluid flow equations .....	20
2.3.2. RANS equations .....	22
2.3.3. Computational domain discretization .....	23
2.4. Test examples of engineering optimization .....	24
2.4.1. Centrifugal roof fan .....	24
2.4.2. Ship hull .....	26
2.4.3. Wind turbine .....	27
3. PARAMETRIC SHAPE FITTING .....	30

3.1.	Enhanced fitting method .....	31
3.1.1.	Projection to rectangular domain .....	32
3.2.	Adaptive fitting .....	35
3.2.1.	Relaxation surface.....	35
3.2.2.	Adaptive re-mapping .....	36
3.2.3.	Flowchart of the procedure .....	37
3.3.	Engineering application of parametric shape fitting .....	37
3.3.1.	Parameterization testing by enhanced shape fitting.....	38
3.3.2.	Boat hull adaptive fitting using B-spline and T-spline .....	38
3.3.3.	Sensitivity based re-parameterization .....	40
3.3.4.	Application to multi-patch parameterizations.....	41
4.	SUMMARY OF COMPLEMENTARY PAPERS.....	43
4.1.	Paper 1: Multi-regime shape optimization of fan vanes for energy conversion efficiency using CFD, 3d optical scanning and parameterization .....	43
4.1.1.	PhD candidate contribution .....	43
4.2.	Paper 2: 3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise related excellence criteria and constraints .....	43
4.2.1.	PhD candidate contribution .....	43
4.3.	Paper 3: Numerical models for robust shape optimization of wind turbine.....	44
4.3.1.	PhD candidate contribution .....	44
4.4.	Paper 4: Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes.....	44
4.4.1.	PhD candidate contribution .....	44
4.5.	Paper 5: Reverzno inženjerstvo i dvo-stupanjska optimizacija broskog vijka i sapnice pomoću B-spline ploha.....	45
4.5.1.	PhD candidate contribution .....	45
4.6.	Paper 6: Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape .....	45
4.6.1.	PhD candidate contribution .....	45
4.7.	Paper 7: Numerical analysis of energy efficiency performance and noise emissions of building roof fan .....	46
4.7.1.	PhD candidate contribution .....	46

4.8. Paper 8: Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting.....	46
4.8.1. PhD candidate contribution .....	46
5. CONCLUSION .....	47
5.1. Ongoing and future work .....	49
BIBLIOGRAPHY .....	50
APPENDIX A - Multi-regime shape optimization of fan vanes for energy conversion efficiency using CFD, 3D optical scanning and parameterization .....	55
APPENDIX B - 3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise-related excellence criteria and constraints .....	71
APPENDIX C - Numerical models for robust shape optimization of wind turbine .....	93
APPENDIX D - Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes.....	109
APPENDIX E - Reverzno inženjerstvo i dvo-stupanjska optimizacija broskog vijka i sapnice pomoću B-spline ploha.....	121
APPENDIX F - Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape optimization workflow....	129
APPENDIX G - Numerical analysis of energy efficiency performance and noise emissions of building roof fan .....	145
APPENDIX H - Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting (submitted for publication).....	155
ŽIVOTOPIS .....	183
BIOGRAPHY .....	184

## List of Figures

Figure 1. Typical multidisciplinary shape optimization workflow with CFD simulator.	3
Figure 2. Pareto front.	8
Figure 3. Flowchart of genetic algorithm.	9
Figure 4. Bezier curve of degree 4 with 5 control points	14
Figure 5. Knot lines for basis function $PB_i(u,v)$ .	16
Figure 6. T-mesh created from a regular B-spline mesh	17
Figure 7. Chaining 3 <sup>rd</sup> degree Bezier surfaces with C1 continuity [24].	18
Figure 8. Schematic representation of a simple optimization problem involving a single parameter and a single objective. The objective function shows two minima. The exact objective function is represented by the solid line while the inaccurate and more realistic case of CFD-based evaluation of the objective function is shown as a dashed line.	19
Figure 9. Finite volume schematic representing nodes and faces.	24
Figure 10. Shape parameterization using generic B-spline surface, used for both vane and domain parameterization.	25
Figure 11. Example of optimization workflow.	26
Figure 12. Developed shape parameterization method.	27
Figure 13. Wind turbine blade parameterization methods: a) Airfoil-based wind turbine blade shape parameterization: T – airfoil translation, L – airfoil scaling and $\alpha$ – airfoil rotation and b) Generic wind turbine blade shape parameterization using B-spline surface.	28
Figure 14. Vertical-axis wind turbine shape parameterization.	29
Figure 15. Determining control points polygon for a known data set.	30
Figure 16. DTMB half of ship hull.	32
Figure 17. Lines j-j, k-k and l-l illustrated in: a) physical space b) parametric space.	32
Figure 18. Point cloud: internal points, boundary points and corner points.	33
Figure 19. Point cloud projected to rectangular parametric domain.	33
Figure 20. Definition of angle $\alpha$ and angle-related distance.	34
Figure 21. Flow chart of the adaptive fitting parametric surface.	37
Figure 22. Results of fitting using the developed adaptive re-mapping for: a) B-spline with 15x20 CPs grid, b) B-spline with 15x9 CPs grid	38
Figure 23. Fitting results.	39
Figure 24. Comparison of error distributions for: a) B-spline with 15x9 CPs grid, b) B-spline with 15x20 CPs grid and b) T-spline with 15x20 CPs grid and additional control points	39
Figure 25. Topologically optimized structure a) distribution of material density b) magnitude of the sensitivity field gradient.	40
Figure 26. Density field described by B-spline surface: a) solution with constant x-y control-point coordinates and b) solution with constant x-y control-point coordinates and adaptive redistribution in x-y direction.	40

Figure 27. Single-patch B-spline surface describing a half-sphere. ....41

Figure 28. Half-cube described by: a) multi-patch parameterization with 5 B-spline surfaces and b) Single-patch B-spline surface obtained by adaptive fitting.....41

Figure 29. Two-bladed propeller described by: a) multi-patch parameterization (only two out of many B-spline patches are shown) and b) Single-patch B-spline surface obtained by the adaptive fitting. ....42

Figure 30. Crossover of single-patch surfaces: a) crossover of half-sphere and half-cube at chromosome mid-point and b) two-point crossover of half-sphere and half-cube c) crossover of half-sphere and boat propeller at chromosome mid-point and d) two-point crossover of boat propeller and half-cube .....42

## List of Tables

Table 1. Practical limits of CFD optimization for a modern PC. ....	20
Table 2. Shape parameterization in some VAWT optimization studies.....	28

## **Abbreviations**

CFD – computational fluid dynamics  
CV – control volume  
GA – genetic algorithm  
HAWT – horizontal axis wind turbine  
IGA – isogeometric analysis  
PDE – partial differential equation  
SO – shape optimization  
VAWT – vertical axis wind turbine  
WT – wind turbine  
MOO – multi-objective optimization  
DOF – degrees of freedom  
NURBS – non uniform rational B-spline  
CP – control point

Blank page



## 1. INTRODUCTION

The introduction section first presents the motivation of the doctoral thesis and the respective hypothesis. Next the scientific methodology used to confirm the posed hypothesis is presented together with the scientific contributions. The contribution is based on papers papers [1]–[8] that are listed in a separate subsection. An overview of the remaining doctoral thesis is given at the end of the introduction section.

### 1.1. Motivation and hypothesis

Shape optimization (SO) is a rather complex undertaking which involves many challenges but nevertheless becomes a necessity in several industries [9]. SO is a part of computational mechanics and can be subdivided in three branches: sizing optimization (for example thickness distribution), shape optimization itself and topology optimization [10]. This doctoral thesis will focus on engineering application of shape optimization itself, but variable shape topology will also be considered.

In SO, the system subject to optimization is usually described by partial differential equations (PDEs). In recent years, the field of optimization of systems based on PDEs has received a large impulse with a variety of research projects being funded by national and international agencies [11], [12]. In engineering application of SO, the optimization problems usually contain multiple mutually conflicting objectives functions, modeled by very different computational methods. This causes difficulties in engineering applications of methods developed specially for solving a specific type of PDEs. This doctoral thesis gives a review of methods required for generic engineering SO of complex engineering systems.

A common example of complex engineering optimization tasks includes computational fluid dynamics (CFD), a system modeled by (nonlinear) PDEs. Concerning CFD, the first applications of optimization are found for aeronautical problems, in particular to improve wing profile and flight properties (typically, reduce drag) [13]. CFD models are a major concern as appropriate turbulence models and domain discretization exhibit a major influence on the simulation results and require substantial computing resources. These tasks also include complex 3D shapes such as ship hulls, wind-turbine blades, fan vanes, etc. Computational modeling of the respective geometry is correspondingly difficult as modeling of both global and local variations of shape is required. Furthermore, only a modest-size data-set of shape parameters has to be used, since otherwise the dimensionality of the subsequent optimization space is very high. The most often solution of these problems is to apply local SO while keeping most of the 3D shape fixed i.e. ignoring the global character. In any case, an integrated optimization workflow must be constructed to contain (at least) a geometry modeler and an engineering simulation node controlled by the optimizer.

The objective of this doctoral thesis is the investigation of existing and development of improved methods that can increase the efficiency of engineering SO problems. Particularly, consideration is given to the application in complex numerical simulations (such as CFD) within

an integrated optimization workflow. Methods for both global and local variations of shape will be investigated. The major part of the doctoral thesis is concerned with geometry modeling since much research is currently directed towards geometry modeling methods (Bezier, B-spline, multi-patch parameterizations,...) and an universal approach does not exist. It is also important to understand the physical principles and engineering models used in the engineering simulation node of the workflow. Without an efficient numerical simulation, parameterization method will not be able to show its full potential no matter how well selected. The optimizer controlling the shape variations within the numerical workflow is also a major concern. Depending on the optimization problem, a different optimization algorithm (genetic algorithm, gradient method,...) may be appropriate. For example, a gradient method will not be able to improve the solution in the case of a “noisy” objective function. An opposing case would be the case of local optimization with a smooth objective function where a genetic algorithm will eventually obtain a solution but the application of a gradient method can save orders of magnitude of computational time.

The thesis hypotheses are as follows:

- It is possible to reduce the number of shape variables by application of B-spline surfaces and advanced shape parameterization methods while keeping the shape generality during the shape optimization. Since the time required for conducting an engineering optimization task depends on the number of variables, the number of variables can be considered sufficiently low if the optimization can be conducted in a reasonable amount of time (order of magnitude of several days). The shape generality means that the search space contains a wide array of possible shapes that can be expected for a given optimization task.
- After the previous hypothesis is shown to be correct on several test examples, various algorithms to aid the shape optimization procedure could be developed. A method of adaptive fitting based on scalar field could be used for two different applications. First, it can be used to aid the preparation of shape parameterization based on existing solutions. Here, the scalar field would use a geometric error between the shape parameterization and the existing solutions. This results in better distribution of the parametric surface control points, meaning that the control points will be clustered at geometrically complex areas while moving away from the flat regions. The adaptive fitting procedure could also be used during the optimization quasi-time. Here, a scalar field such as acting forces or sensitivity with respect to an objective function would be used for control-points re-distribution. Another application of adaptive fitting would be fitting of one type of shape parameterization (phenotype) to a different type thus allowing crossover between multi-patch parameterizations with a different number of partitions and variables (genotypes).

## 1.2. Research methodology and scientific contributions

The first part of the research is the development of efficient shape parameterization using integral surfaces on several different engineering examples and conducting the shape optimization. Thus, several shape parameterization methods for specific applications must be developed. This requires development of various numerical workflows which can be used for testing different shape parameterizations as a part of the engineering optimization. Numerical integration of the required elements in the form of a workflow is not trivial since it needs to encapsulate both process flows with synchronization of processes and the necessary data-mining. The data transfers within the workflow include results for all candidate designs. On the input side of the engineering simulations (CFD,...), the current shapes of the candidate designs need to be communicated to the simulators ('data-burying'). As the engineering object generally functions across a range of operating regimes, a good definition of the excellence criteria is also crucial. Moreover, the design variables may include additional parameters beyond those which control the shape, including discrete variables. Some of the variables can be given only as statistical distribution, making the problem even more difficult. A schematic of a general workflow is illustrated, Figure 1. Corresponding data mining is represented with vertical direction while the coordination of process is represented by the horizontal direction.

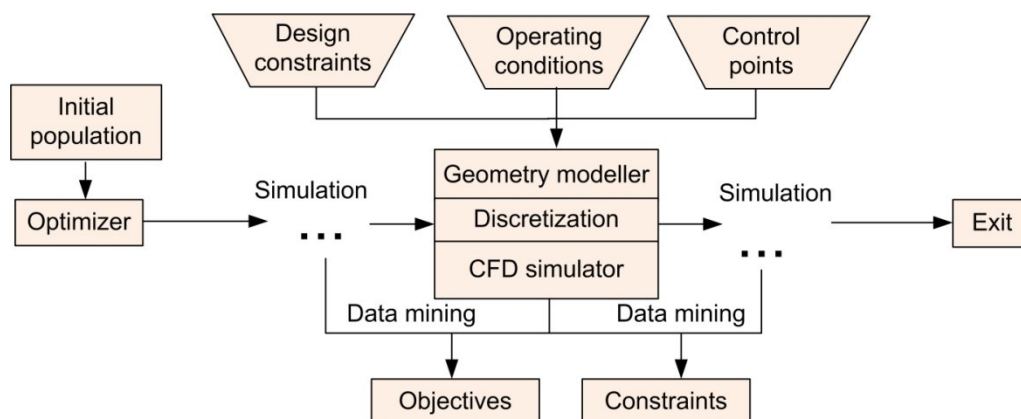


Figure 1. Typical multidisciplinary shape optimization workflow with CFD simulator.

A generic engineering SO task has the following elements:

1. For an engineering object and corresponding operating conditions the objectives and constraints have to be defined;
2. Numerical simulations are conducted (and developed) to confirm that the computer model is able to replicate reality for the required operating conditions. Typically multiple test cases are conducted and compared with experimental results;
3. Fine tuning of the numerical simulation to improve the computational efficiency and keep the required accuracy;
4. Real-life solutions of a similar problem already exist, but they are not optimal for a particular optimization objective. The shape of the pre-existing solutions can be used as

- a starting point of SO task. If the geometry file is not available, 3D optical scanning can be used to provide the point cloud representing the initial shape;
5. Parameterization of the 3D point cloud into computational geometry entities (NURBS, T-spline,...) provides for optimization parameters. Fitting procedures can be used to evaluate the performance of parameterization and in this way the most appropriate shape parameterization procedure can be selected;
  6. Definition of the excellence criteria and objective functions for the optimizer such that the fitness functions for the candidate designs can be evaluated. In the case of multi-regime operation, numerical sample of operating regimes has to be generated in order to evaluate excellence;
  7. Definition of the optimization constraints. A common example is setting the bounds for the selected parameterization control point mobility. Further constraints can be related to non-shape variables (mass, stress, temperature, revolution velocity, etc.).
  8. Initial solution or a set of initial solutions based on the selected parameterization is generated;
  9. Launching the optimizer with corresponding operators and parameter values. Most often a genetic algorithm-based optimizer is selected. This requires a setup of selection operators, cross-over operators and probabilities, mutation, elitism, fitness scaling, etc;
  10. Linking an array of engineering simulators to provide values of excellence and constraints for all candidate designs represented by corresponding shape parameterization.
  11. Iteratively shape-optimizing the engineering object within the numerical cycle embedding the optimizer, shape modeler and engineering simulators (CFD, etc.).

Elements of the engineering SO tasks (3), (5) and (9) have the most potential to be improved and are the topics of interest in this doctoral thesis. The focus was mainly put on step (5) since new emerging shape parameterization and fitting procedures are not yet sufficiently tested in the context of generic shape engineering optimization, hence leaving room for improvement. Since various different engineering optimization problems will be conducted, a general conclusion regarding the application of the integral parametric surfaces can be obtained.

The second part of research is the development of an adaptive fitting procedure which allows easier testing of different shape parameterization methods without a time-consuming optimization. This adaptive fitting method should be based on an adaptive control-points redistribution based on various scalar fields which can be selected based on the optimization objective. In shape fitting, the scalar field is a geometric error between the shape parameterization and the pre-existing solutions. When using the adaptive fitting procedure during the optimization quasi-time, a scalar field such as sensitivity with respect to an objective function should be used for control-points re-distribution. Finally, a concept for a procedure of fitting different shape parameterizations to each other can be developed. This opens possibilities of application of genetic algorithm operators such as crossover to mutually different genotypes.

Scientific contributions of the doctoral thesis are:

- It was shown that by using an integrated B-spline surface it is possible to reduce the number of shape variables while keeping the shape generality in an engineering optimization task, i.e. more efficient parameterization methods were developed.
- Several original optimization workflows for specific engineering shape optimization problems were developed.
- An adaptive parameterization method which can enable testing of different shape parameterizations before the actual optimization procedure was developed. The same method could be used for dynamic control-point redistribution during the optimization quasi-time. Various scalar fields can be used such as the sensitivity field with respect to an objective function.
- It was shown that adaptive fitting can be used to convert different parameterization types (patch topologies) to the same single-patch parameterization. This allows for usage of variable length chromosomes and multi-patch parameterization in the same optimization. It was shown that a smooth crossover between various shapes of different patch topologies is possible.

### 1.3. List of complementary papers

This doctoral thesis presents the basis and offers a short survey for the papers [1]–[6], [8], [7] along with overall conclusions which can be obtained from them. The following is the list of papers:

- [1] Z. Milas, D. Vučina, and I. Marinić-Kragić, “Multi-regime shape optimization of fan vanes for energy conversion efficiency using CFD, 3D optical scanning and parameterization,” *Eng. Appl. Comput. Fluid Mech.*, vol. 8, no. 3, pp. 407–421, 2014.
- [2] I. Marinić-Kragić, D. Vučina, and Z. Milas, “3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise-related excellence criteria and constraints,” *Eng. Appl. Comput. Fluid Mech.*, vol. 10, no. 1, pp. 210–228, 2016.
- [3] D. Vučina, I. Marinić-Kragić, and Z. Milas, “Numerical models for robust shape optimization of wind turbine blades,” *Renew. Energy*, vol. 87, pp. 849–862, 2015.
- [4] I. Marinić-Kragić, D. Vučina, and Z. Milas, “Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes,” in *European Wind Energy Association Annual Conference and Exhibition 2015*, 2015.
- [5] I. Marinić-Kragić, P. Bagavac, and I. Pehnc, “Reverzno inženjerstvo i dvo-stupanjska optimizacija brodskog vijka i sapnice pomoću B-spline ploha,” in *Sedmi susret hrvatskog društva za mehaniku*, 2016, pp. 121–126.
- [6] I. Marinić-Kragić, D. Vučina, and M. Ćurković, “Efficient shape parameterization

method for multidisciplinary global optimization and application to integrated ship hull shape optimization workflow,” *Comput. Des.*, vol. 80, pp. 61–75, Nov. 2016.

- [7] I. Marinić-Kragić, Z. Milas, and D. Vučina, “Numerical analysis of energy efficiency performance and noise emissions of building roof fan,” in *9th International Exergy, Energy and Environment Symposium (IEEEES-9)*, 2017.
- [8] I. Marinić-Kragić, M. Ćurković, and D. Vučina, “Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting,” *Submitted for publication*

#### **1.4. Thesis overview**

The introduction section first presented the motivation of the doctoral thesis and the respective hypothesis. Next the scientific methodology used to confirm the posed hypothesis was shortly explained. More detail about the elements of the used tools and methods are presented in section 2 and section 3. Section 2 presents the methods used for generic shape optimization which are required for the first part of the hypothesis. The section 3 presents the methods for shape fitting and shape manipulation that are required for the second part of the hypothesis. This introduction section has already presented the list of the scientific contributions while the papers on which the contribution is based are reviewed in section 4. The contribution of the author is presented separately for each paper. The final sections are the conclusion section, recommendation for future work and the bibliography.

## 2. ELEMENTS OF NUMERICAL OPTIMIZATION WORKFLOW

All of the components of the optimization task can be integrated in a single numerical workflow controlled by the optimizer based on the selected optimization methods. After the initial solution is selected (based on earlier designs or randomly generated), the direction of the shape modification is controlled by the selected optimization algorithm. The result of the engineering optimization task is of course primarily dependent on the selected optimization method. This section will briefly describe important optimization methods and when is it appropriate to apply one or the other since no single superior approach exists. Further improvements that could improve the convergence speed of numerical optimization are also briefly described, for example a surrogate model can be introduced in the optimization workflow.

Engineering SO problems usually belong to class of continuous nonlinear multi-objective optimization (MOO) although discrete variables can be present. Methods for solving continuous MOO problems can be roughly divided in methods relying on standard optimization engines (single-objective optimization methods) such as gradient methods, and other approaches such as genetic algorithms [14]. When discrete variables are present in a MOO problem, genetic algorithms represent an appropriate and robust but computationally more demanding method.

### 2.1. Numerical optimization

Most engineering optimization problems involve multiple excellence criteria. Single-objective optimization is a special case with number of objective functions equal to one. A general multi-objective problem can be posed:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{Minimize}} \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_k(\mathbf{x})]^T \\ & \text{Subject to} \begin{cases} g_j(\mathbf{x}) \leq 0, & j = 1, 2, \dots, m \\ h_l(\mathbf{x}) = 0, & l = 1, 2, \dots, e \end{cases} \end{aligned} \quad (1)$$

where  $m$  is the number of inequality constraints, and  $e$  is the number of equality constraints,  $k$  is the number of objective functions,  $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^k$  is a vector of objective functions  $F_i(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$ .  $F_i(\mathbf{x})$  are the objective functions, whereby an individual function is a mapping from the design space to single objective space. The gradient of a function  $F_i(\mathbf{x})$  with respect to  $\mathbf{x}$  is written as  $\nabla_{\mathbf{x}} F_i(\mathbf{x}) \in \mathbb{R}^n$ . The point  $\mathbf{x}_i^*$  minimizes the objective function  $F_i(\mathbf{x})$ . The feasible design space  $\mathbf{X}$  is defined as the set  $\{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m; \text{ and } h_i(\mathbf{x}) = 0, i = 1, 2, \dots, e\}$ . Each point in the design space maps to a point in criterion space, but every point in the criterion space does not necessarily correspond to a single point in design space.

In comparison to single-objective optimization, the solution of the multi-objective problems solutions cannot be defined uniquely. The optimal solution can be regarded more as a concept than as a definition. The principal concept in defining an optimal point is that of Pareto optimality, which is defined as follows:

A point,  $\mathbf{x}^* \in \mathbf{X}$ , is Pareto optimal if there does not exist another point,  $\mathbf{x} \in \mathbf{X}$ , such that  $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$  and  $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$  for at least one function.

In case of maximization problems, the Pareto optimal designs are illustrated in the objective space in Figure 2. In this case, the objective is usually multiplied by minus one so that the earlier definitions do not need to be modified.

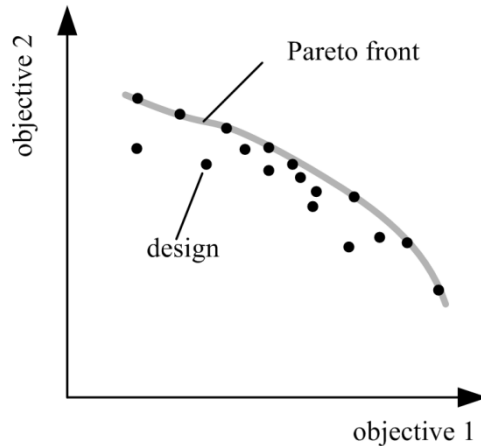


Figure 2. Pareto front.

### 2.1.1. Genetic algorithms

Several modern heuristic tools exist for solving optimization problems that are difficult (or even impossible) to solve using classical methods. These tools include simulated annealing, tabu search, particle swarm, evolutionary computation, etc. These techniques are finding popularity within research community as design tools and problem solvers because of their flexibility and ability to optimize in complex highly non-linear search spaces with multiple local minima. GA can be viewed as a general-purpose search method. It is based loosely on Darwinian principles of biological evolution, reproduction and “the survival of the fittest”. GA maintains a set of candidate solutions called the population and repeatedly modifies them by application of genetic operators.

The initial population is usually generated randomly and its size is selected based on experience from similar optimization problems. From the initial (and subsequent) generations, the next generation is generated in several steps by application of selection, reproduction and mutation operators. In first step, selection operator creates temporary clones of the selected individuals. By preferring more fit individuals, the selection operator implements “the survival of the fittest” principle. Children are obtained by combining features of randomly selected clones, from two parents two children are created by reproduction operator. The third step is applying small amount of random modifications to children i.e. mutation. In some cases it is



desirable to keep selected individuals so they are cloned without modifications. Over successive generations, the population evolves toward the optimal solution since the individuals with lower function values have higher fitness value and are more likely to be selected for reproduction (survival of the fittest). The GA is well suited to and has been widely applied to solve complex engineering optimization problems. GA can handle both continuous and discrete variables, nonlinear objective and constraint functions without requiring gradient information. Genetic algorithms are global optimization techniques, which means that they generally converge to the global solution rather than to a local solution. However, this distinction becomes unclear when working with multi-objective optimization where Pareto optimal solutions are obtained. The defining feature of GA multi-objective optimization methods is that not just local but global Pareto solutions are determined by the procedure [14]. The flowchart of genetic algorithm is illustrated in Figure 3.

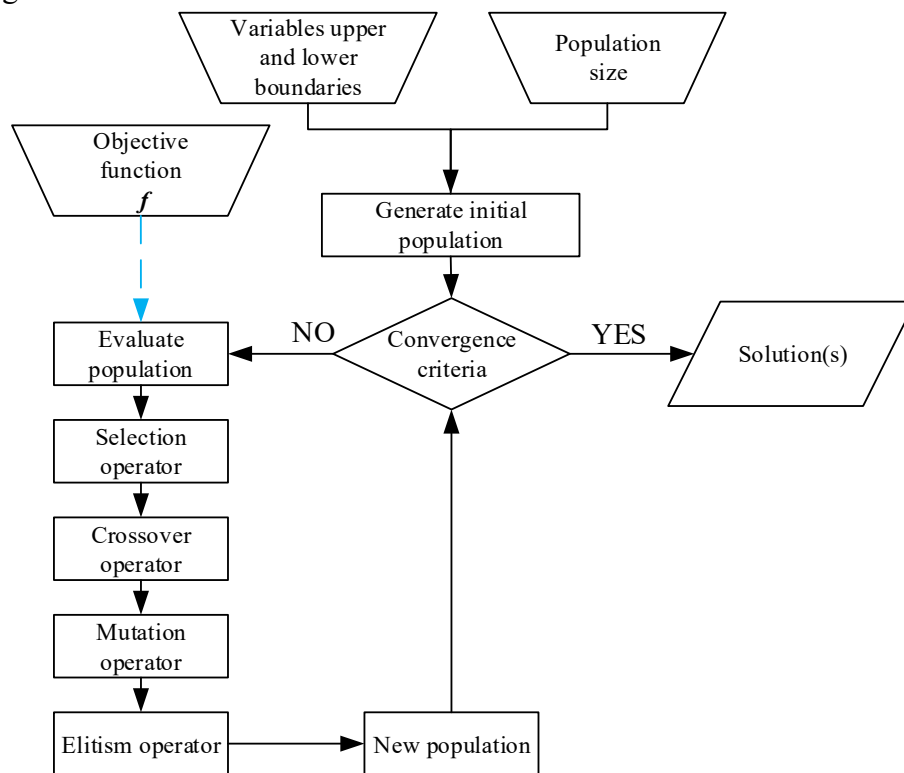


Figure 3. Flowchart of genetic algorithm.

In optimization which involves simultaneous topology and shape optimization, various approaches exist. When the level-set method [15] is applied method, the same parameterization can be used while allowing the topology variations. In the cases when the level-set method is not applied, a different approach to topology optimization is required. One of the alternate approaches could be the application of variable length chromosomes [16]. For such problems, the number of variables for which a solution is searched does not need to be known in advance. This type of variable length chromosomes could contain information of multiple set of objects, their shapes, locations, etc. The chromosomes would grow in size adding more genes if more objects are added during the optimization procedure. Various similar problems (and solutions)

regarding variable length chromosomes exist and a good review of the methods is given in [17]. A genetic algorithm with variable length chromosome could be applied for multi-patch parameterization since the topology of the optimal solution is not known in advance.

### 2.1.2. Sensitivity analysis

Design sensitivity analysis plays an important role in several areas including numerical optimization. Sensitivity information is also valuable on its own for estimation of robustness of the obtained solution with respect to the optimization variables. It is possible to determine how the value of a given response function (optimization objective), changes with respect to a given change in the model parameters. For small perturbations  $|\Delta \mathbf{x}|$ , this information is obtained through a first-order Taylor series expansion. If the response function is the stress at a point of a structural system, aerodynamic performance measure, or some other quantity evaluated through a computationally expensive numerical analysis, then it can be predicted how the response function value varies for small perturbations in the model parameters without performing a re-analysis. Much work has been performed in the field of design sensitivity analysis. Most notably, perhaps, is the work in structural mechanics with applications to structural optimization [18].

The simplest method to obtain sensitivity information is the finite difference method. In the finite difference method, a simple Taylor series expansion is used to approximate to the derivative:

$$f(\mathbf{x} + \Delta \mathbf{x}) = f(\mathbf{x}) + \sum_i \frac{\partial f(\mathbf{x})}{\partial x_i} \Delta x_i + o(\Delta x_i^2) \quad (2)$$

where the derivatives are approximated with finite differences for every design variable independently. This method thus suffers from computational inefficiency and possible errors. More often used is the adjoint method whose basics are explained briefly in continuation. The adjoint method originates in the theory of Lagrange multipliers in optimization [19]. As an example, consider the minimization of a function  $J$  from  $\mathfrak{R}^n$  into  $\mathfrak{R}^1$  (single-objective), under the equality constraints  $h$  from  $\mathfrak{R}^n$  to  $\mathfrak{R}^m$ .

$$\text{Minimize } J(\mathbf{x}) \text{ such that } h(\mathbf{x}) = 0 \quad (3)$$

Introducing the Lagrange multiplier  $\lambda$  in  $\mathfrak{R}^m$  and the Lagrangian  $\mathcal{L}$ :

$$\mathcal{L}(\mathbf{x}, \lambda_1, \lambda_2, \dots, \lambda_m) = J(\mathbf{x}) + \sum_{k=1}^m \lambda_k h_k(\mathbf{x})$$

the optimality condition for (3) (under some qualification conditions) is the stationary value of the Lagrangian, namely

$$\nabla \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0$$

The adjoint method is an extension of this approach in the framework of optimal control theory. In this context, the variable  $\mathbf{x}$  is the union of a state variable  $\mathbf{y}$  and a control (shape) variable  $\mathbf{u}$ , while the constraint  $h(\mathbf{y}, \mathbf{u}) = 0$  is the state equation (governing equations). In such a case the Lagrange multiplier  $\lambda$  is called the adjoint state. Denote by  $\mathbf{u}$  in  $\mathbb{R}^k$  a control variable.

The state of the system is denoted by  $\mathbf{y}$  in  $\mathbb{R}^n$  and is defined as (for finite dimensional case) the solution of the following state equation:

$$\mathbf{A}(\mathbf{u})\mathbf{y} = \mathbf{b} \quad (4)$$

where  $\mathbf{b}$  is a given right hand side in  $\mathbb{R}^n$  and  $\mathbf{A}(\mathbf{u})$  is an invertible  $n \times n$  matrix. Since the matrix depends on  $\mathbf{u}$ , so does the solution  $\mathbf{y}$ . The goal is to minimize, over all admissible controls  $\mathbf{u}$ , an objective function  $J(\mathbf{u}, \mathbf{y})$  under the constraint (4). The difficulty of the problem is that  $\mathbf{y}$  depends nonlinearly on  $\mathbf{u}$ . It is assumed here that  $\mathbf{u}$  is a continuous variable. To numerically minimize the objective function  $J$ , the most efficient algorithms are those based on derivative informations. Therefore, a key issue is to compute the gradient of objective function  $\nabla J(\mathbf{u}, \mathbf{y}(\mathbf{u}))$  which is dependent both on the shape  $\mathbf{u}$  and the solution  $\mathbf{y}$  (depends nonlinearly on  $\mathbf{u}$ ). The gradient  $\nabla J(\mathbf{u}, \mathbf{y}(\mathbf{u}))$  is computed using the adjoint method. There are at least two ways for introducing an adjoint in a computer code. Either, a so-called analytic adjoint, is implemented, or a program for the adjoint is obtained by automatic differentiation of the code solving the state equation (4). The drawback of the whole adjoint approach is that it requires development of an adjoint solver (no general purpose adjoint solver exist for CFD). This can be a labor intensive task which requires good knowledge of the implemented numerical tools [19].

Optimization with aid of the solution to the adjoint system of equations is an active area of research with applications in both structural optimization and computational fluid dynamics, particularly for aeronautical applications [20]. In fluid dynamics, the first use of adjoint equations for design was by Pironneau [21] for minimization of drag in stokes flow. Application of the adjoint approach is popular in the optimal design of structures for topology optimization problems. In those cases, the number of optimization variables describing the system is so large that it is the only viable approach. In the context of topology optimization for mechanical structures, the number of design variables is even larger since any cell of a “hold-all” computational domain is potentially a design variable: either it is full of material or empty.

The sensitivity approach is helpful mainly in the context of gradient-based optimization and such optimization has its own limitations. Firstly, it is only appropriate when the design variables are continuous. For design variables which can take only integer values (e.g. the number of vane of the fan, or number of blades of wind turbine) stochastic procedures such as GA are usually implemented since the gradient does not exist in those cases. Secondly, if the objective function contains multiple minima, the gradient approach will generally converge only to the nearest local minimum. This is usually solved by multi-start from various locations in search space if one wants to find the global minimum using gradient methods. Still, if the objective function is known to have multiple local minima, and possibly discontinuities, a stochastic search method such as GA may be more appropriate [20].

## 2.2. Shape parameterization

This doctoral thesis considers engineering SO as a combination of fully generic 3D shape modeling, complex engineering numerical simulation (3D CFD), and global optimization combined. Hence a heavy computational effort of the respective numerical workflow can be expected. This requires an efficient shape parameterization with as much geometric modeling capability with a small number of optimization variables. Surveys of parameterization methods

for engineering applications have been made but they are usually limited to a specific object, for example [22], [23] give a survey of parameterization methods on 2D airfoil.

Wide variety of shape parameterization methods exist, as there is no single representation of surfaces that satisfies all of the needs for engineering SO problems. The simplest shape representation method is the mesh points method, which requires a large number of design variables but there are no restrictions on attainable geometry. Since it requires a large number of variables, it is not very useful for generic SO. While CAD models are often used in engineering, they consist of many interconnected partitioned geometric entities with corresponding parameters, relationships and constraints. This makes them unappropriated for SO and therefore other parameterization methods need to be considered. Complex 3D shape might in the most elementary approach be represented by a single polynomial surface, requiring high-degree polynomials. High degree polynomials have issues of lacking local control and they exhibit oscillatory behavior likely to introduce numerical difficulties. The first shape parameterization method that will be presented in the doctoral thesis is the Bezier surface (single-patch) parameterization. The Bezier single-patch surface offers a more intuitive shape parameterization method, explained in more detail in continuation of this section. A generalization of Bezier patches is B-spline surface, one of the most often used surfaces for shape parameterization. Furthermore, their generalizations are the non rational uniform B-splines (NURBS) and T-splines. All of the mentioned generalizations of the Bezier surface can be regarded as single-patch parameterizations. In order to achieve further generalizations, multi-patch surfaces are required. Often used multi-patch surfaces in CAD are subdivision surfaces, but any single patch surface can be connected (with varying degree of smoothness) to create a multi-patch surface. Regarding the application in engineering optimization, still, no universal solutions exist. So an appropriate selection amongst a wide variety of existing methods and development of new parameterization methods offers a lot space for improvement.

Before the actual shape parameterization methods are presented, three types of shape representation will be discussed. Two commonly used surfaces in shape modeling are parametric and implicit surfaces but sometimes explicit surfaces are also appropriate. The explicit surface can be defined with single expression  $z = S(x, y)$ . The parametric surface required three expressions similar to the explicit surface expression, what can be written as:

$$\begin{bmatrix} x(\mathbf{u}) \\ y(\mathbf{u}) \\ z(\mathbf{u}) \end{bmatrix} = \mathbf{S}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^2 \quad (5)$$

where:

- $\mathbf{S}$  is a mathematical function, mapping  $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ , i.e. from parametric space to model space. For two-dimensional shapes that exist in three dimensional space, but in general case  $\mathbb{R}^n \rightarrow \mathbb{R}^m$ ;
- $\mathbf{u}$  is a point in 2D parametric space.

The implicit surface is defined by:

$$I(\mathbf{x}) = 0, \quad \mathbf{x} \in \mathbb{R}^n \quad (6)$$

where:

- $I$  is a mathematical function, mapping  $\mathbb{R}^n \rightarrow \mathbb{R}$ , but in general case  $\mathbb{R}^n \rightarrow \mathbb{R}^1$ . For example  $n=2$  gives a implicit definition of curve. Algebraic, trigonometric, exponential, logarithmic etc. functions can be used, as well as any function described in the continuation of this section (most often with RBF-s);
- $\mathbf{x}$  is a point in 3D space;
- $n$  is the dimensionality of space in which surface is located, for real objects  $n=3$ .

Comparably, explicit surfaces have the least shape generality as they cannot represent infinite slopes (if polynomials are used) or closed and multi-valued surfaces. The advantage is that they are easy to construct and display. The explicit surface can easily be converted to implicit  $S(x, y) - z = 0$  but reverse can be achieved only for simple cases. Implicit surfaces are difficult and non-intuitive to manipulate and not easy to construct (display). Curves can be constructed only in 2D. The implicit surfaces compared to parametric surfaces offer some advantages such as: any topology can be represented by a single mathematical function; easy representation of intersection; easy point classification for internal ( $S(\mathbf{x}) < 0$ ) and external points ( $S(\mathbf{x}) > 0$ ). However, parametric surfaces are easy to construct and enable intuitive manipulation what is very important and makes parametric surfaces the most used type in engineering SO applications. The following sections will chiefly presume that parametric surfaces are used although with little modification the same mathematical functions can be used to define implicit surfaces.

### 2.2.1. Bezier Patches

Bezier curves and surfaces (5) have a property which is convenient for shape parameterization: they pass through the end control points, the end slopes are defined by the respective control points next to the ends, and analogously for higher-order derivatives. These properties are used in this paper to impose inter-segment continuity for piecewise chained curves. Parametric curve can be defined by the  $n$ -th degree Bezier curve  $P(u)$  for  $(n+1)$  control points) as [24]:

$$\begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \mathbf{S}(u) = \sum_{i=0}^n B_{i,n}(u) \cdot \mathbf{Q}_i \quad , \quad 0 \leq u \leq 1 \quad (7)$$

where  $B$  (Bernstein polynomial) is defined recursively (non-recursive and more efficient definition also exists) as:

$$B_{i,n}(u) = (1-u) \cdot B_{i,n-1}(u) + u \cdot B_{i-1,n-1}(u) \quad (8)$$

*with:  $B_{0,0}(u) = 1, B_{i,n}(u) = 0$  for  $i < 0, i > n$*

Vector  $\mathbf{Q}_i$  is the vector of control points (Figure 4) coordinates and  $u \in [0,1]$  is the curvilinear coordinate parameter.

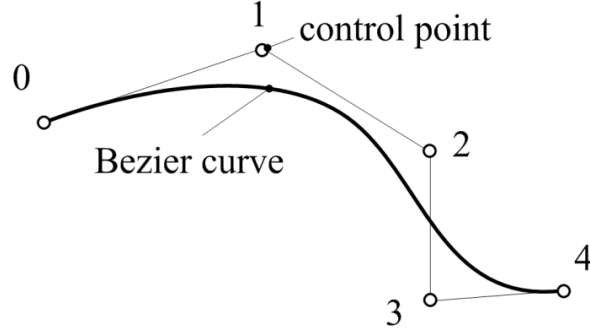


Figure 4. Bezier curve of degree 4 with 5 control points

Bezier curves can be extended to 3D Bezier surfaces (and bodies) for describing parametric surfaces (and bodies). This is achieved by combining Bezier curves, whereby control points of a Bezier curve are replaced by Bezier curves in the orthogonal direction,

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) \cdot B_{j,m}(v) \cdot \mathbf{Q}_{i,j} \quad , \quad u, v \in (0,1) \quad (9)$$

where  $\mathbf{Q}_{i,j}$  are the control points of the control polyhedron. Bezier curves do not possess the property of locality and directly link the number of control nodes with the degree of the respective curve.

### 2.2.2. B-spline and NURBS

B-splines are a generalization of Bezier curves (and a special case of NURBS) where the degree of the curve is independent of the number of control points and where the change of one of the control points only affects  $k$  segments. A parametric surface described by B-splines can be defined as:

$$\mathbf{S}(u, v) = \sum_{i=0}^{n_0} \sum_{l=0}^{n_1} N_{i_0, d_0}(u) \cdot N_{l, d_1}(v) \cdot \mathbf{Q}_{i_0 l} \quad , \quad u, v \in [0,1] \quad (10)$$

with blending functions defined recursively as

$$N_{i,0}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad , \quad 0 \leq i \leq n+d \quad (11)$$

$$N_{i,j}(t) = \frac{t-t_i}{t_{i+j}-t_i} N_{i,j-1}(t) + \frac{t_{i+j+1}-t}{t_{i+j+1}-t_{i+1}} N_{i+1,j-1}(t) \quad , \quad 1 \leq j \leq d, 0 \leq i \leq n+d-j$$

$$0 \leq i \leq n+d+1 \quad (12)$$

The equations (10)-(12) define the B-spline. As the individual shape functions  $N_{i,j}(t)$  are non-zero just for the  $[t_i, t_{i+j+1})$  interval, while amounting to zero for  $t < t_i$  and  $t \geq t_{i+j+1}$ , the property of local control is ensured. As a result, the surface is locally formed exclusively by a small number of adjacent control points,  $\mathbf{Q}_{ij}$ . The first and the last blending function in both  $u$

and  $v$  directions are equal to unity at the ends while the rest of the blending functions are equal to zero making the surface pass coincidentally through the end curves.

The shape of the surface is controlled by modifying the control points and the knot vectors. The properties of local support, partition of unity and non-negativity add to the numerical stability of the subsequent optimization procedure. B-spline and NURBS surfaces are flexible enough and provide sufficient degrees of freedom (DOF) to represent the necessary shape for ship hull representation. These integral shape parameterizations are also scalable as the number of control points and the degrees of the basic polynomials can be varied. The NURBS curves and surfaces are a generalization of the B-spline. NURBS use the same blending functions as B-splines but in  $n+1$  dimensional space where  $n$  is the actual space dimensionality ( $n=2$  for plane,  $n=3$  for 3D space). The additional coordinate adds the ability to increase and decrease the impact of an individual control point on the resulting shape thus enabling better control. Furthermore, with additional coordinate analytical shapes (conics) can be exactly represented. For NURBS curve, the fourth coordinate is:

$$h = \sum_{i=0}^n h_i \cdot N_{i,d}(u) \quad (13)$$

where  $h_i$  denotes respective weights for  $i$ -th control point. The shape is projected to the original space by dividing with the fourth coordinate. Thus the NURBS for parametric curve:

$$\mathbf{S}(u) = \frac{\sum_{i=0}^n h_i \cdot N_{i,d}(u) \cdot \mathbf{Q}_i}{\sum_{i=0}^n h_i \cdot N_{i,d}(u)} \quad (14)$$

For parametric surfaces, extension of NURBS curves to surfaces is defined by:

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^{n_0} \sum_{j=0}^{n_1} h_{i,j} \cdot N_{i_0,d_0}(u) \cdot N_{j_1,d_1}(v) \cdot \mathbf{Q}_{i_0j_1}}{\sum_{i=0}^{n_0} \sum_{j=0}^{n_1} h_{i,j} \cdot N_{i_0,d_0}(u) \cdot N_{j_1,d_1}(v)} \quad (15)$$

### 2.2.3. T-Splines

B-splines and NURBS belong to a type of tensor-product surfaces that use a rectangular grid of control points. If one wants to insert a control point (CP) on a B-spline surface at a desired location, this cannot be made without inserting a whole new row and/or column of CPs. To solve this problem, an adequate generalization of the B-spline surface called the T-spline surface was developed. To define the T-spline, it is necessary first to describe a surface whose CPs have no topological relationship with each other whatsoever (as opposed to regular grids). This surface is called a PB-spline, because it is point-based instead of grid-based. The equation for a parametric surface using a PB-spline is:

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^n PB_i(u, v) \cdot \mathbf{Q}_i}{\sum_{i=0}^n PB_i(u, v)} \quad (16)$$

where  $\mathbf{Q}_i$  are control points and  $PB_i(u, v)$  are basis functions given by:

$$PB_i(u, v) = N_{i,3}(u)N_{i,3}(v) \quad (17)$$

and  $N_{i,3}(u)$  and  $N_{i,3}(v)$  are the cubic B-spline basis functions associated with the knot vectors:

$$\begin{aligned} \mathbf{u}_i &= [u_{i0}, u_{i1}, u_{i2}, u_{i3}, u_{i4}] \\ \mathbf{v}_i &= [v_{i0}, v_{i1}, v_{i2}, v_{i3}, v_{i4}] \end{aligned} \quad (18)$$

as illustrated in Figure 5. Thus, to specify a PB-spline, one must provide a set of knot vectors for each CP.

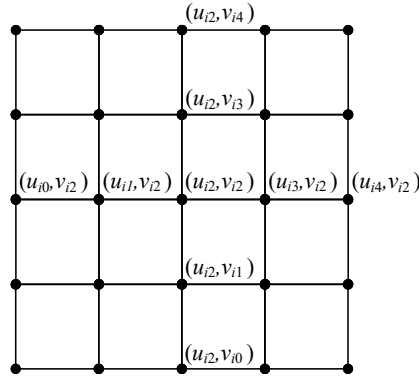


Figure 5. Knot lines for basis function  $PB_i(u, v)$ .

Now with the definition of a single PB-spline basis function, one can proceed to define a T-spline surface which is composed of multiple PB-spline basis functions. A T-spline is a PB-spline for which order has been imposed on the CP mesh. The simplest T-spline is the one where PB-spline basis functions are set in an ordered grid. This results in a parametric surface equivalent to the B-spline and this was also used as the initial T-spline grid in this paper.

An example of a more complex control-point mesh for a T-spline is illustrated in Figure 6 where each circle represents the center of a  $PB_i$  basis function (knot vector components  $u_{i2}$  and  $v_{i2}$ ). To complete the definition of each basis function, the remaining knot vector components for  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are obtained from the nearest edges in straight-line horizontal and vertical directions respectively.

After creating the initial CPs grid, additional CPs can be added (grey filled circles) in accordance to the T-spline rules:

- In Figure 6  $u_i$  and  $v_i$  designate parametric coordinates of the respective edge while  $e_i$  and  $f_i$  designate the knot intervals i.e. distance in parametric domain ( $e_1 = u_2 - u_1$ , etc). The first rule is that the sum of the knot intervals on opposing edges must be equal. Thus,  $e_5 + e_6 = e_3$  and  $f_5 + f_4 = f_2$ .
- If a T-junction (intersection of three edges) on one edge can be connected to a T-junction on an opposing edge without violating the previous rule, that edge must be included in the T-mesh.
- A new control point (basis function  $PB_i$ ) can only be inserted on a horizontal edge if knot vector  $\mathbf{v}$  is equal for  $PB_i$  and the four neighboring basis functions in the horizontal direction. In the case of inserting  $PB_3(Q_3)$  this would be the



basis functions  $PB_1, PB_2, PB_4$  and  $PB_5$ . This means that control point  $Q_3$  can be inserted since the respective knot vectors  $\mathbf{v}$  are equal. Meanwhile,  $Q_6$  (x mark) cannot be added since the vectors  $\mathbf{v}$  for  $PB_0$  and  $PB_1$  are not equal. If the control point is to be inserted on a vertical edge the same is valid for the  $u$  knot vectors.

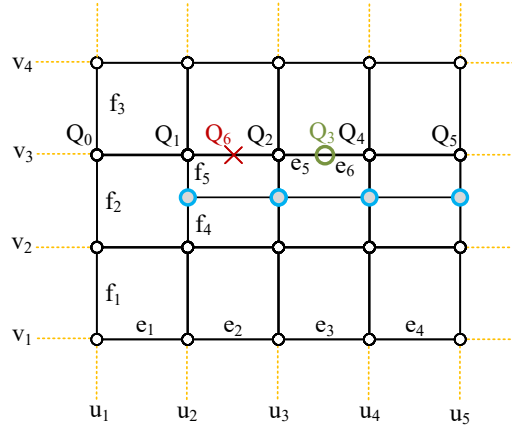


Figure 6. T-mesh created from a regular B-spline mesh

#### 2.2.4. Multi-patch parameterizations

Any of the mentioned single patch shape parameterization methods can be used to generate a multi-patch parameterization by imposing the required continuity conditions on the boundaries. If the single patch shapes are not adjacent in parametric domain, first the intersection has to be calculated. When using T-splines multi-patch parameterization may even not be necessary i.e. are only necessary if topological changes are required.

Application of multi-patch parameterization can be illustrated on chained Bezier surfaces. In the paper [24], chained piecewise Bezier curves and surfaces are developed to provide for locality and low-degree curve. Bezier patches are piecewise by definition, with each defined only within a partial segment of the entire domain. However, while reducing the problems of locality and possible oscillations, this raises an additional request of imposing and ensuring adequate continuity between the piecewise segments. In the framework of design optimization, chaining of piecewise curves should also not increase the parameterization complexity, as increasing the number of control points would drastically increase the numerical effort due to higher dimensionality of the search space. The proposed procedure in [24] is based on subdividing the domain into patches and chaining piecewise low-degree Bezier curves and surfaces into complex shapes. The procedure starts by subdividing the problem domain into patches, for each of an individual approximation curve or surface with corresponding control points is assigned. In segments (between adjacent original control points) where chained surfaces are to be joined with  $C1$  continuity, additional control points are interpolated as illustrated in Figure 7 using the local original points. The number of optimum design variables (original control points) is thereby not increased since the generated points depend on the original ones.

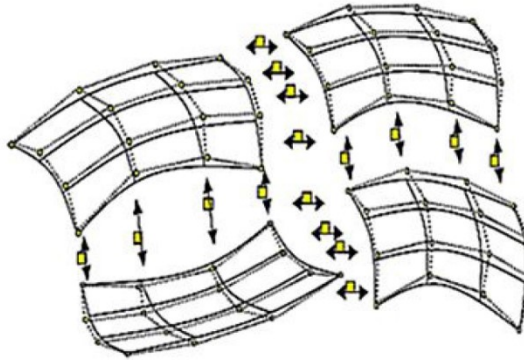


Figure 7. Chaining 3<sup>rd</sup> degree Bezier surfaces with  $C1$  continuity [24].

### 2.3. Computational fluid dynamics

Since most engineering optimization cases used in this thesis contain a computational fluid dynamics (CFD) simulation, this section presents the properties of CFD that should be considered in shape optimization procedure. CFD is undergoing significant expansion regarding its engineering applications, number of researchers active in the field, and also the number of courses offered at universities. For solving fluid flow problems, numerous software packages exist but still the market is not quite as large as the one for structural mechanics codes. While for structural problems, the finite element method is well established, CFD problems are, in general, more difficult to solve. However, CFD codes are slowly being accepted as design tools by the industry [25]. Even if accurate CFD modeling is possible, the issue when using CFD as part of an optimization workflow is the respective computational time required for simulation.

The simplest optimization problem would be single variable, single objective optimization. These kind of optimization problems are usually easy to solve. Still, several problems can appear. Even in an ideal CFD simulation multiple minima are possible, so when starting from two different initial points, two different local minima might be found as illustrated Figure 8. Furthermore, a realistic CFD simulation will not yield in ideally smooth objective function curve, but will have some “noise” added to the ideally smooth function. This makes the optimization even more difficult since even when starting from near the global minimum, the optimization procedure will soon end trapped in a local noise-induced optimum (illustrated in Figure 8 by triangles). This illustrates that even in the simplest single objective case, the optimization is not as trivial as it might seem. The gradient methods that are by far most efficient in optimization of smooth functions are in this case not directly applicable. Thus, when incorporating engineering simulations in an optimization workflow, all components of the workflow (optimization method, shape parameterization and engineering simulation - CFD) need to be considered.

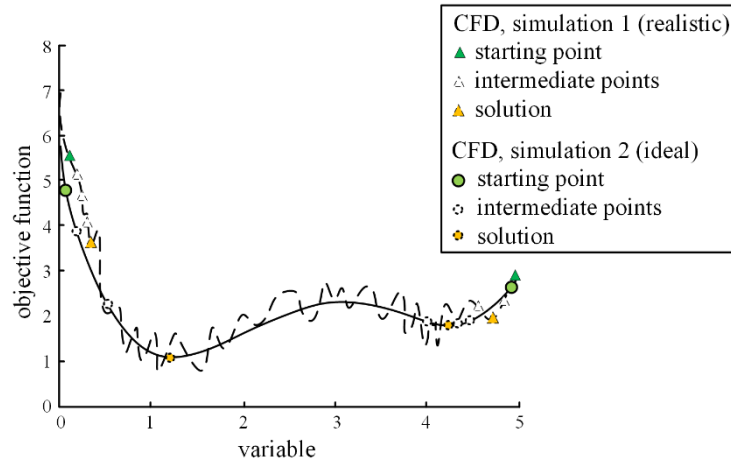


Figure 8. Schematic representation of a simple optimization problem involving a single parameter and a single objective. The objective function shows two minima. The exact objective function is represented by the solid line while the inaccurate and more realistic case of CFD-based evaluation of the objective function is shown as a dashed line.

Real optimization problems found in engineering as well as in fundamental research will generally involve several (often too many) shape variables and several objectives. It is important to know the practical limits of optimization based on CFD. The most limiting factor is the computing time requirement for a single evaluation. Admittedly, each CFD evaluation with shape modification will usually lead to slightly different numerical costs but the variations are normally negligible relative to the overall computational cost. This means that the computational cost depends mostly on the type of simulation that is conducted, for example, on a modern PC the following CFD simulations could be conducted:

- one flow simulation with included thermal analysis for a photovoltaic panel takes typically about 10 hours of computing time and 4 GB of computer memory [26];
- one simplified simulation of the turbulent channel flows requires less than 5 seconds of computing time and 10 Megabytes (MB) of computer memory;
- one transient 3D simulation of Darrieus wind turbine requires 16 Gigabytes (GB) of computer memory and about 30 days of computing time;

By looking at the computing requirements for these few examples, it is evident that the computational time can vary by 5 orders of magnitude and the computer memory by 3 orders of magnitude. This is typical of what will be found in practice: simple CFD problems can be solved within seconds on a standard PC; high-end CFD problems can require weeks of computing times and Terabytes (1 TB=1000 GB) of memory on supercomputers. In such cases, the limits of optimization when using CFD are clear in principle: CFD based optimization is possible for simple CFD problems and almost impossible for high-end CFD configurations. To somewhat quantify the possibilities, considering again the earlier examples:

- only 100 evaluations corresponding have been conducted in [26] and this already at an extremely high computational cost;

- on the other hand, more than 5000 evaluations have been easily carried out for the simplified turbulent channel flow in [13];
- for the transient 3D simulation of Darrieus wind turbine lasting 30 days, no practical optimization could be conducted.

As a rule of thumb for engineering practice, it therefore seems appropriate to state: CFD based optimization is possible when the duration of a single CFD computation does not exceed a few hours at most [13]. Practical limits of CFD optimization are illustrated in Table 1 for a modern PC.

Table 1. Practical limits of CFD optimization for a modern PC.

Evaluation cost	Optimization cost for $\leq 2$ parameters	Optimization cost for $\geq 3$ parameters
low cost (1 min CPU, 100 MB memory)	very easy	easy up to 20 parameters
medium cost (1h CPU, 1000 MB memory)	easy	still possible up to 20 parameters by parallelization
high cost (20h CPU, 10 GB memory)	still possible	almost impossible

In order to achieve improvements in optimization speed, the sensitivity with respect to the objective function can be calculated by adjoint methods [20], [21]. While this is easily done (and well documented in the literature), e.g. for the Euler equations, the task will become much more difficult when considering complex, multiphysics problems. Furthermore, the adjoint approach requires a full knowledge of the intermediate approximations on a way to the full solution of the system of equations solved by CFD. In simple words, this means that the adjoint approach, while greatly reducing the number of evaluations (and computational time), will also lead to a huge increase of the requested computer memory which will again become a major problem for complex, three-dimensional flows involving many unknowns at each discretization point [13]. The adjoint approach (by default) used gradient optimization methods and is accompanied with all of the disadvantages of gradient methods such as finding only the nearest local minimum. For a generic optimization task with ambition to find a global solution this approach can be helpful but it cannot be used exclusively. This doctoral thesis aims mostly at investigating efficient shape parameterization techniques that could be used in any engineering simulation (structural, multiphysics...) involving SO. But CFD is used since it is widely applied in combination with SO, and it illustrates many problems occurring when using engineering simulations. The following sections will briefly layout the basics of CFD.

### 2.3.1. Fluid flow equations

Fluids are substances whose molecular structure offers no resistance to external shear forces: even the smallest force causes deformation of a fluid particle. For most engineering applications, a fluid is regarded as a continuous substance (continuum). Flow equations can be given by multiple conservation law. For mass, which cannot be created or destroyed, the conservation equation can be written (for a control mass, CM):

$$\frac{Dm}{Dt} = 0 \quad (19)$$

where  $m$  stands for mass and  $t$  for time,  $\frac{D}{Dt}$  is the material derivative (later  $D$  will be used for rate of strain tensor). As opposed to mass the amount of change momentum is not equal to zero. Newton's second law of motion states that:

$$\frac{d(m\mathbf{v})}{dt} = \sum \mathbf{f} \quad (20)$$

where  $\mathbf{v}$  is the velocity, and  $\mathbf{f}$  is the forces acting on the control mass. The left hand side of equations (19) and (20) can be generalized for conservation of an intensive property  $\phi$ . This generalization can be written as an integral equation:

$$\frac{D}{Dt} \int_{\Omega_{CM}} \rho \phi d\Omega = \frac{d}{dt} \int_{\Omega_{CV}} \rho \phi d\Omega + \int_{S_{CV}} \rho \phi (\mathbf{v} - \mathbf{v}_b) \cdot \mathbf{n} dS \quad (21)$$

where  $\Omega_{CM}$  stands for volume occupied by the control mass,  $\rho$  is density,  $\Omega_{CV}$  is the control volume (CV) volume,  $S_{CV}$  is the surface enclosing CV,  $\mathbf{n}$  is the unit vector orthogonal to  $S_{CV}$  and directed outwards,  $\mathbf{v}$  is the fluid velocity and  $\mathbf{v}_b$  is the velocity with which the CV surface is moving. For mass conservation  $\phi = 1$ , for momentum conservation  $\phi = \mathbf{v}$ . For common case of a CV fixed in space,  $\mathbf{v}_b = 0$  and the first derivative on the right-hand side becomes a local (partial) derivative. This equation states that the change of the amount of property in the control mass  $\Omega_{CM}$  is the rate of change of the property within the control volume plus the flux of it through the CV boundary caused by the fluid motion relative to CV boundary. The flux is called the convective flux of  $\rho \phi$  through the CV boundary. The integral form of the mass conservation (continuity) equation in case of fixed control volume follows directly from (21) by setting  $\phi = 1$ :

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \int_S \rho \mathbf{v} \cdot \mathbf{n} dS = 0 \quad (22)$$

again for a fixed control volume, the momentum conservation equation is obtained by setting  $\phi = \mathbf{v}$  and combining with (20). The resulting equation is:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{v} d\Omega + \int_S \rho \mathbf{v} \mathbf{v} \cdot \mathbf{n} dS = \sum \mathbf{f} \quad (23)$$

Continuity equations can also be written in coordinate-free differential form by application of Gauss divergence theorem to the second term and reducing the volume to infinitesimally small. To express the right-hand side in terms of intensive properties, surface forces and body forces have to be considered.

The surface forces are from the molecular point of view, the microscopic momentum fluxes across a surface, macroscopically represented by pressure and stresses. If these fluxes

cannot be expressed by density and velocity, the system of equations is not closed since there would be fewer equations than dependent variables. By making assumptions that the fluid is Newtonian (it is in most practical engineering applications), a closed system of equations can be assured. For Newtonian fluids, the stress tensor  $T$ , which is the molecular rate of transport of momentum, can be written (using index notation):

$$T_{ij} = - \left( p + \frac{2}{3} \mu \frac{\partial u_j}{\partial x_j} \right) \delta I_{ij} + 2\mu D_{ij} \quad (24)$$

where,  $\mu$  is the dynamic viscosity,  $I$  is the unit tensor,  $p$  is the static pressure and  $D$  is the rate of strain (deformation) tensor:

$$D_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (25)$$

The body forces per unit mass can be represented by force vector  $\mathbf{b}$ , so the integral form of the momentum conservation equation now becomes:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \mathbf{v} d\Omega + \int_S \rho \mathbf{v} \mathbf{v} \cdot \mathbf{n} dS = \int_S \mathbf{T} \cdot \mathbf{n} dS + \int_{\Omega} \rho \mathbf{b} d\Omega \quad (26)$$

The equation (26) with (24) and (25) included is the most general form of momentum conservation equation for Newtonian fluids. All fluid properties (density,...) can vary both in space and in time. However in many applications the fluid density is practically constant. Constant density may be assumed for most liquid flows, but also for gases in case of low Mach number ( $<0.3$ ). Such flows are called to be incompressible. However this simplification is generally not of a great value, as the equations are similar to the original ones regarding difficulty of obtaining a solution. However, assuming constant density and viscosity does help in numerical solution and it should be implemented if the problem allows it. In addition to simplification to incompressible flow, various other simplifications are used depending on the conducted simulation. It is important to know what simplifications can reasonably be used to lower the computational time but still keep sufficient simulation physics so that the correct optimal solution can be obtained in a reasonable computational time.

### 2.3.2. RANS equations

The continuity and momentum equations in Reynolds averaged differential form can be summarized, in tensor notation and Cartesian coordinates as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho \bar{u}_j)}{\partial x_j} = 0 \quad (27)$$

$$\frac{\partial \rho \bar{u}_i}{\partial t} + \frac{\partial \rho \bar{u}_i \bar{u}_j}{\partial x_j} = - \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \mu \left[ \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right] + \frac{\partial}{\partial x_j} [R_{ij}] + \rho f_i \quad (28)$$

$$R_{ij} = -\overline{\rho u_i u_j} = \mu_t \left[ \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right] - \frac{2}{3} \rho k \delta_{ij} \quad (29)$$

where  $\rho$ ,  $p$  and  $\bar{u}_i$  are the mean flow density, pressure and the Cartesian velocity components respectively;  $x_i$  is the Cartesian coordinate;  $\mu$  is the molecular viscosity; The specific body force  $f_i$  includes external body forces such as gravity. Turbulent viscosity  $\mu_t$  and turbulence kinetic energy  $k$  are obtained from a selected turbulence model. The continuity and momentum equations in some cases have to be solved together with energy conservation equation:

$$\frac{\partial \rho e}{\partial t} + \frac{\partial}{\partial x_j} [(\rho e + p) \bar{u}_j] = \frac{\partial}{\partial x_j} \left[ (k' + k'_t) \frac{\partial T}{\partial x_j} \right] + S_h \quad (30)$$

Where  $h$  is enthalpy;  $k'$  thermal conductivity;  $k'_t$  is the thermal conductivity due to turbulence and is defined by turbulence model;  $T$  is the temperature ;  $S_h$  is volumetric heat source; and  $e$  (specific total internal energy) can defined by:

$$e = h - \frac{p}{\rho} + \frac{\bar{u}^2}{2} \quad (31)$$

### 2.3.3. Computational domain discretization

Most often in CFD, the finite volume method is implemented to discretize the computational domain. It uses the integral form of the conservation equation as the starting point:

$$\int_S \rho \phi \mathbf{v} \cdot \mathbf{n} dS = \int_S \Gamma \text{grad}(\phi) \cdot \mathbf{n} dS + \int_{\Omega} q_{\phi} d\Omega \quad (32)$$

The usual approach is to define control volumes (CVs) by a suitable grid and assign the computational node to the CV center, see Figure 9. Nodes on which boundary conditions are applied are shown as full circles in this figure. The derivatives that appear in conservation equations are approximated by finite differences. Domain discretization methods are very important since computational time can be reduced by an order of magnitude while keeping good computational physics if appropriate methods are used. Also, mesh generation can take a considerable amount of overall time of a shape generated by the optimizer.

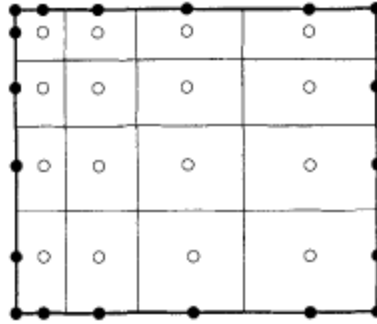


Figure 9. Finite volume schematic representing nodes and faces.

The result of the discretization process is a system of algebraic equations, which are linear or non-linear according to the nature of the partial differential equations from which they are derived. In the non-linear case, the discretized equations must be solved by an iterative technique that involves guessing a solution, linearizing the equations at the current solution, and improving the solution; the process is repeated until a converged result is obtained. So, whether the equations are linear or not, efficient methods for solving the linear systems of algebraic equations are needed. The selection of the method is an important step, since orders of magnitude in computational time can be lost if an inappropriate method is chosen. Various guidelines (for example ANSYS Fluent help [27]) exist but the appropriate method is problem-dependent.

After the solution is obtained, the last step is post-processing of the flow field. This usually includes integrating surface forces (or forces\*distance to obtain torque) on boundaries that represent the object of optimization. This value is finally sent to the optimizer and the whole process is repeated till the convergence criteria are met.

## 2.4. Test examples of engineering optimization

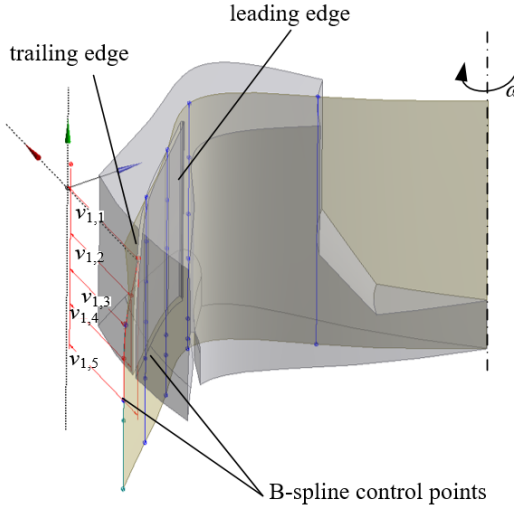
The overall numerical optimization problem with all included components is usually integrated in an optimization workflow. The workflow represents the final product of integrating all the required components: engineering simulations, parameterization methods and optimization algorithms. Several examples of engineering optimization were conducted as a part of this doctoral thesis: centrifugal fan, ship hull, vertical axis wind turbine and horizontal axis wind turbine. This section briefly presents the developed methodology and the results for each test case while the respective detailed description can be found in papers [1]–[6].

### 2.4.1. Centrifugal roof fan

The part of the centrifugal fan considered for optimization is the vane shape. In previous research [28], most common shape variables were geometric parameters such as the leading edge angle, the trailing edge angle and the vane length. In the simple optimization case, the shape of the vane is evaluated by a single CFD simulation for a single flow regime. However, multi-regime aspects should be considered in order to obtain a robust solution. In the papers [1], [2], procedures for multi-regime optimization were developed for centrifugal fan vanes. The



developed workflow is capable of fully generic 2D and 3D shape optimization of a centrifugal roof fan vane by manipulating the control points of parametric surfaces as illustrated in Figure 10. This parameterization method is an upgrade version of the one developed in [1] for 2D optimization.



*Figure 10. Shape parameterization using generic B-spline surface, used for both vane and domain parameterization.*

Additional variables used were rotation speed and the discrete number of vanes. The excellence formulation was based on design flow efficiency, multi-regime operational conditions and noise criteria for various cases including multi-objective optimization. Multiple cases of optimization demonstrate the ability of customized and individualized fan design for the specific working environment according to the selected excellence criteria. Noise analysis was also considered in an multi-objective optimization workflow as an additional decision making tool. The workflow for the multi-regime operational conditions is illustrated in Figure 11. In the figure, the horizontal direction represents the process flow while the vertical direction represents simultaneous data flow. Corresponding data mining and coordination of the process flow were implemented using commercial software modeFRONTIER [29]. The computational workflow needed to implement the procedure encapsulates geometric modeling (computer-aided design, CAD), simulation software (computational fluid dynamics, CFD) and calculators (statistical mean calculation) coupled with the evolutionary numerical optimization. The numerical coupling needs to include both the process executions and their mutual synchronization as well as data flows between the individual applications. The workflow is in this example controlled by MOGA-II, a version of multi-objective genetic algorithm. Detailed description of input and output quantities is available in [2]. Here the workflow is illustrated just to present that it is indeed possible to construct a working numerical workflow that integrates all the necessary applications and procedures. The results of the optimization process have shown that the B-spline shape is applicable to the centrifugal fan vanes. It was shown that

both global and local optimization method should be used to improve the convergence of the optimization procedure.

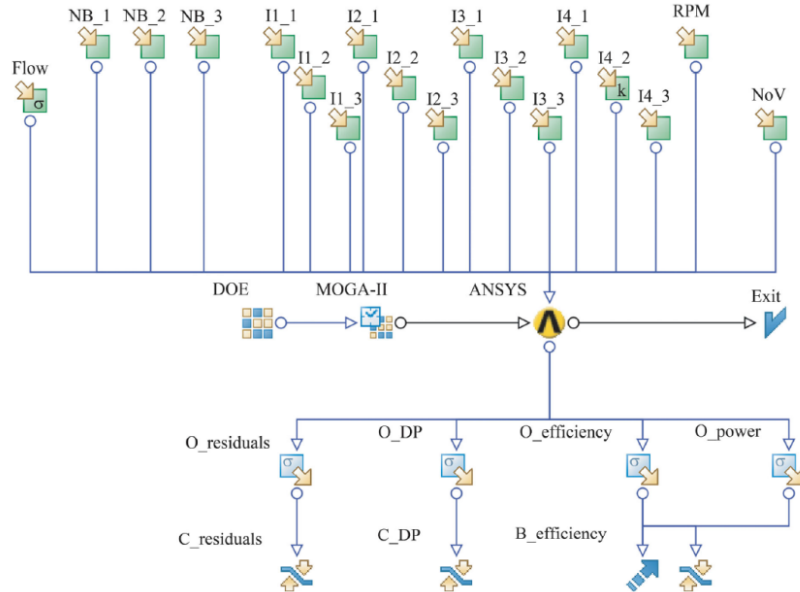


Figure 11. Example of optimization workflow.

#### 2.4.2. Ship hull

Ship hull shape optimization is a research topic with a large number of research papers [30]–[38]. This research often includes multi-disciplinary optimization [31], [33] which requires both fluid and structural mechanics. Paper [39] gives a review of ship hull parameterization methods. Regarding the application of B-spline and NURBS surfaces, it was concluded that their application is limited by the demand for a regular control-points grid and fixed topology. T-spline and sub-division surfaces are considered as more efficient methods but they require an additional procedure for control-point insertion. This might pose a problem during the optimization in which the location where the control-point insertion is required, is not known in advance. This opens a space for development of optimization algorithms for adaptive insertion and removal of the control-points. However, a better stability of the optimization procedure would be obtained by a fixed control-points grid.

Thus, a parameterization method would ideally be capable of describing complex shapes with a fixed number of control-points. As a part of this doctoral thesis, a method for efficient global optimization using a single-patch B-spline surface was developed in [6]. This method can describe different complex shapes which are even topologically different using a fixed set of control-points. The parameterization method is illustrated in Figure 12. The parameterization is achieved by using a B-spline surface and a symmetry plane which is used to trim a part of the surface for which  $y < 0$ . This means that not all of the control-point degrees of freedom are required which can potentially reduce the number of shape variables in comparison to full 3D shape parameterization.

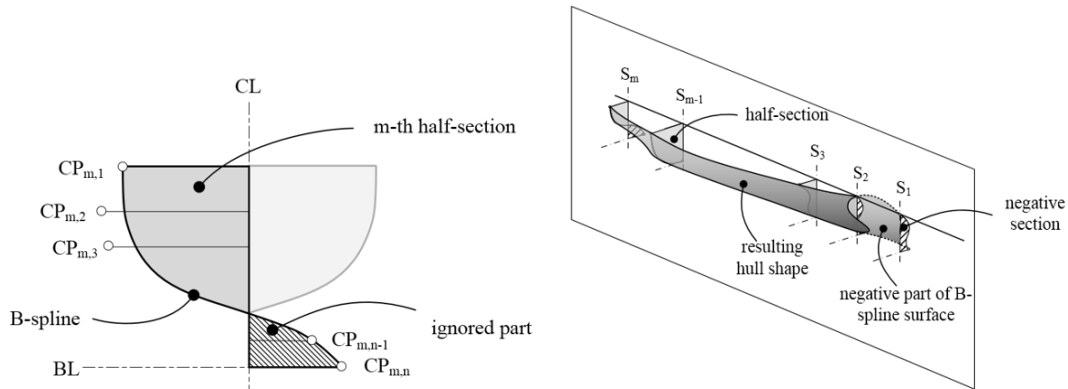


Figure 12. Developed shape parameterization method.

This parameterization method was used as a part of multi-disciplinary shape optimization workflow. The workflow included a geometry modeler, structural requirements calculator and a CFD simulator. Single case of shape optimization procedure was conducted. The result was a Pareto set of solutions which are a compromise between structural weight and hydrodynamic drag. It was shown that the proposed parameterization method can be used in complex multi-disciplinary shape optimization.

### 2.4.3. Wind turbine

Wind turbines can be divided into two categories: horizontal-axis wind turbines (HAWT) and vertical-axis wind turbines (VAWT). HAWTs typically have an airfoil-based blade shape which can be constructed by scaling and rotation of a selected airfoil shape. Meanwhile, VAWTs appear in a diverse array of shapes. A large amount of papers regarding wind turbine optimization exist and different parameterization methods have been used.

In the earliest papers [40] regarding VAWT optimization (and still in newer papers [41], [42]) the wind turbine solidity (ratio of surface taken by the blade and the total swept surface area) was used as an optimization variable. However, this parameter by itself does not allow for sufficient generality required for optimization. Better shape control is obtained by using a parameterized air-foil. The most commonly used [43], [44] airfoils are the NACA 4-digit series where the optimization variables are the maximum width, maximum camber and location of the maxima. In [45], [46] the shape is described by using parametric curves (Bezier and NURBS) where the variables are the respective control-points. These 2D shape parameterizations require 16 to 19 optimization variables. If this is combined with 3D CFD as a part of a numerical optimization workflow, 19 is already a large number of variables and only 2D shape is the shape that can be obtained. Also, researchers have shown that height-wise angle [47] and chord [48] distribution also have influence on the wind turbine performance. For HAWT optimization, shape parameterization usually involves constant airfoil with chord and twist angle distribution as variables, but span-wise translation of airfoil sections can also aid improvements [49]. Generic B-spline and cubic splines parameterizations have also been developed for HAWT and similar shape optimization problems [50]–[52]. Nevertheless, the VAWT optimization task requires a specialized parameterization method which permits a very wide range of different

shapes with small yet sufficient number of shape variables. An overview of shape parameterization methods is given in Table 2.

Table 2. Shape parameterization in some VAWT optimization studies

Considered variables	shape	Number of shape variables	Performance prediction method	Reference
Rotor solidity		1	MST,CFD	[40]–[42]
NACA 4-series airfoils		4	CFD	[43], [44]
2D Bezier parametric curve control points		16	2D unsteady panel method [53]	[45]
2D NURBS control points		19	CFD	[46]
Helical angle, chord length, rotor diameter, pitch angle, blade thickness ratio		5	CFD	[47]

The first wind turbine optimization tests conducted as a part of this doctoral thesis were HAWT optimization tests. First, the numerical CFD model was tested based on NREL experiments [54] for 10 meter diameter wind turbine with S806 airfoil. The airfoil performance was also compared with experimental data from [55], [56]. After it was confirmed that the CFD can predict the wind turbine performance with a satisfactory accuracy, shape parameterization methods were developed on the basis of the NREL wind turbine. The first shape parameterization was airfoil-based. A fixed airfoil with variable rotation, translation and scaling was used along the blade. The airfoil itself was described by a B-spline curve so it can be fixed or variable as illustrated in Figure 13a. The second proposed parameterization is based on a generic B-spline surface with 2 degrees of freedom per control points as illustrated in Figure 13b for the upper half of the blade.

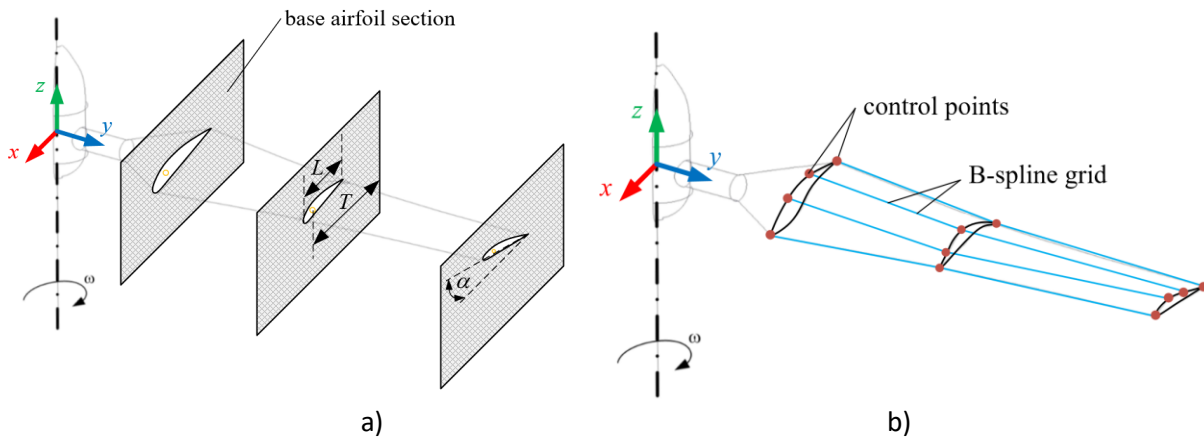


Figure 13. Wind turbine blade parameterization methods: a) Airfoil-based wind turbine blade shape parameterization:  $T$  – airfoil translation,  $L$  – airfoil scaling and  $\alpha$  – airfoil rotation and b) Generic wind turbine blade shape parameterization using B-spline surface.

The next step is integration of the geometry modeler with CFD simulator in an optimization workflow. The initially selected objective function was maximum power

coefficient at a selected wind speed. Finally, the annual energy production was used, which means that the wind speed was described by a statistical distribution. So, several CFD simulations were required in order to calculate the annual energy production. For this case, in addition to the shape variables, the blade rotation (“pitch”) as a function of wind speed was also used.

For the VAWT optimization case, the problem is even more complex. In addition to wind speed described by a statistical distribution, each VAWT design has to be evaluated by a transient CFD simulation for several rotations and the VAWT characteristics are obtained by averaging. The following pseudocode illustrates the required number of CFD simulations in the optimization procedure:

```

while (convergence criteria)
  shape = generate new shape
  for i = 1 do number of operating regimes
    for j = 1 do 360
      aerodynamic power = fCFD(shape,i,j)
    end
  end
end
end

```

In addition to numerical complexity, VAWTs can appear in various mutually very different shapes such as the Savonius [57] and Darrieus [58] types. This means that the usage of the same shape parameterization as for HWATs is not appropriate. The proposed parameterization method is an airfoil-based parameterization using a B-spline surface as shown in Figure 14. A single airfoil is described by a set of B-spline control points in 2D as shown in Figure 14a, but if all 2D degrees of freedom are used there would be 24 variables related only to a single airfoil. The shape variables have to be chosen as low as possible but still the parameterization has to be capable of describing both the Savonius and the Darrieus shapes. The number of variables was reduced to only 8 by relating all of the control points to supplementary variables as shown in Figure 14a. Only the control point  $cp_1$  was allowed for independent 2D freedom of shape. This 2D parameterization defines a single “row” of B-spline surface control points, designated as  $profile_i$  in Figure 14b. For full blade description, the base control point row was translated and rotated along the height direction. The detailed description of the method and the obtained results are shown in [4].

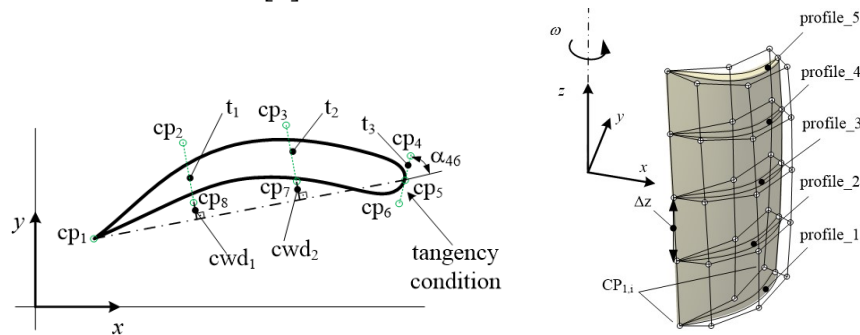


Figure 14. Vertical-axis wind turbine shape parameterization.

### 3. PARAMETRIC SHAPE FITTING

The section 2.2 described how to generate surfaces from a known control point (CP) net. The inverse problem is also of interest; i.e., given a known set of data on a surface, determine the control net that best approximates that data. This is known as shape fitting. The parametric shape fitting is important in different research areas and there is abundant literature dealing with it [59]. This thesis deals mostly with surface fitting which can be divided into techniques that do not [60], [61] and ones that do [62], [63] require mapping of 3D shape to planar domain. It was shown that the second group is more appropriate for the purposes of this doctoral thesis so they will be the topic of this section.

By using shape fitting of parametric surfaces to the already existing solutions of the optimization problem at hand, a preliminary result of a parameterization method efficiency can be obtained. This makes a valuable tool since numerically expensive full optimization is not required in order to test a parameterization method. Also, adaptive parameterization method (changing the parameterization methods during the optimization procedure) could be developed with the aid of parametric shape fitting.

The problem of fitting is shown for illustration in Figure 15. The problem is determining a control polygon that generates a B-spline curve to approximate a set of known data points.

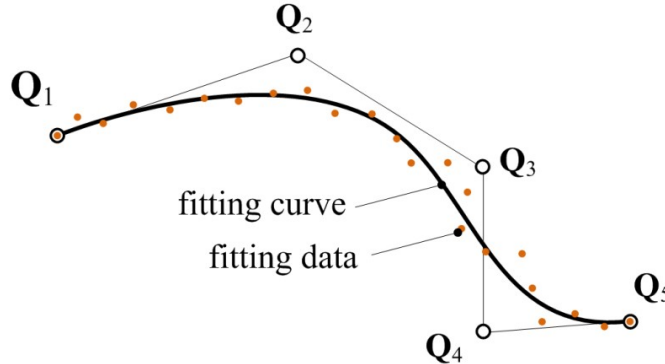


Figure 15. Determining control points polygon for a known data set.

If objective is that the curve passes through all data points, the following equations must be satisfied:

$$\sum_{i=0}^n B_i(u_j) \cdot Q_i = P_j \quad \text{where } i = 1, \dots, n \text{ and } j = 1, \dots, m \quad (33)$$

where  $B_i(u)$  are the basis functions (Bezier, B-spline, NURBS, ...)  $P_j$  are the data points,  $Q_i$  are the control points. The system of equations is more compactly written in matrix form as:

$$[\mathbf{B}][\mathbf{Q}] = [\mathbf{P}] \quad (34)$$

If  $n=m$ , the problem is interpolation and simple inversion  $[\mathbf{Q}] = [\mathbf{B}]^{-1} [\mathbf{P}]$  can be used to obtain the control points. For  $n < m$ , the following can be conducted by using the pseudoinverse:

$$\begin{aligned} [\mathbf{B}]^T [\mathbf{B}][\mathbf{Q}] &= [\mathbf{B}]^T [\mathbf{P}] \\ [\mathbf{Q}] &= \left([\mathbf{B}]^T [\mathbf{B}]\right)^{-1} [\mathbf{B}]^T [\mathbf{P}] \end{aligned} \quad (35)$$

In general case, the problem is solved by minimization of the following least square error function:

$$E(\mathbf{Q}) = \left| \sum_{i=0}^n B_i(u_j) \cdot \mathbf{Q}_i - \mathbf{P}_j \right|^2 \quad (36)$$

For surfaces, the least square error function can be written as:

$$E(\mathbf{Q}) = \left| \sum_{i=0}^n B_i(u_j, v_j) \cdot \mathbf{Q}_i - \mathbf{P}_j \right|^2 \quad (37)$$

The continuation of this section will present fitting methods that were used in this doctoral thesis.

### 3.1. Enhanced fitting method

This section presents the enhanced fitting method which was used in [6] as a part of the doctoral thesis. The enhanced fitting method uses a non-linear fitting procedure that allows for B-spline control points crowding and aggregation at locations of certain geometric features thus enabling good shape fitting while keeping low numerical complexity and high stability [64][65]. The enhanced fitting method is crucial since the linear fitting cannot replicate the mentioned control point aggregation that would accrue in the case of real optimization. The fitting method is composed of a projection of a point cloud  $\mathbf{P}$  to a rectangular domain and obtaining an initial solution by linear fitting of a B-spline to the projected point cloud. After the linear fitting, a gradient method and genetic algorithm are used for improving the solution. As opposed to classical B-spline fitting, the parametric coordinates ( $u$  and  $v$ ) of the (point cloud) individual points are additional fitting variables that allow spline control points movement towards locations of certain geometric features such as the sharp edges. In that case, the error function subjected to minimization is:

$$E(\mathbf{U}, \mathbf{V}, \mathbf{Q}) = \frac{1}{2} \sum_{j0=0}^{m0} \sum_{j1=0}^{m1} \left| \sum_{i0=0}^{n0} \sum_{i1=0}^{n1} N_{i0,d0}(u_{j0j1}) \cdot N_{i1,d1}(v_{j0j1}) \cdot \mathbf{Q}_{i0i1} - \mathbf{P}_{j0,j1} \right|^2 \quad (38)$$

where  $\mathbf{P}$  is point cloud matrix representing the hull shape;  $\mathbf{U}$  and  $\mathbf{V}$  are matrices of parametric values:

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix} u_{00} & \dots & u_{0m_1} \\ \dots & \dots & \dots \\ u_{m_0 0} & \dots & u_{m_0 m_1} \end{bmatrix} \\ \mathbf{V} &= \begin{bmatrix} v_{00} & \dots & v_{0m_1} \\ \dots & \dots & \dots \\ v_{m_0 0} & \dots & v_{m_0 m_1} \end{bmatrix} \end{aligned} \quad (39)$$

### 3.1.1. Projection to rectangular domain

For the initial solution, the  $U$  and  $V$  values are fixed and only the control points coordinates are obtained. Before obtaining the initial solution, an ordered structured point cloud is required such that it can be written in matrix form. To obtain the required ordered distribution for  $U$  and  $V$  in the case of the full 3D spline fitting, a projection of shape from physical space to parametric space is required. The most-simple method is to use parallel cross-sections which can individually be projected to the respective location in the parametric domain. However, this is not appropriate for complex shapes such as for example the DTMB hull [66] illustrated in Figure 16.

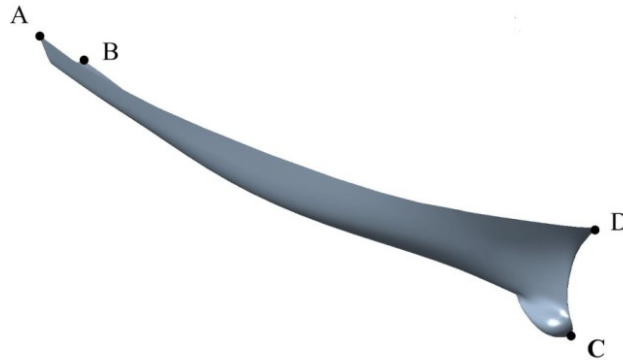


Figure 16. DTMB half of ship hull.

Figure 17 illustrates the necessity of projection to rectangular domain. The line  $j-j$  which is the straight deck line with constant height can easily be projected to  $u-v$  domain by projection in  $y$ -axis direction. Similarly, the line  $k-k$  also appears that it can be easily be projected but in a general case, line  $k-k$  is a 3D curve. The line designated  $l-l$  is obviously a 3D curve and it cannot be projected to the parameter space in a simple matter as the line  $j-j$ . Various methods of projection of 3D surfaces to rectangular domain exist. Another possibility is applying the spring analogy to the meshed point cloud in order to achieve a regular square shape in parametric  $u$  and  $v$  coordinates [67] and applying linear interpolation to generate the required ordered distribution of points. Here, a procedure developed in [64] is implemented to achieve the required matrix topology since it provides for much better initial guess.

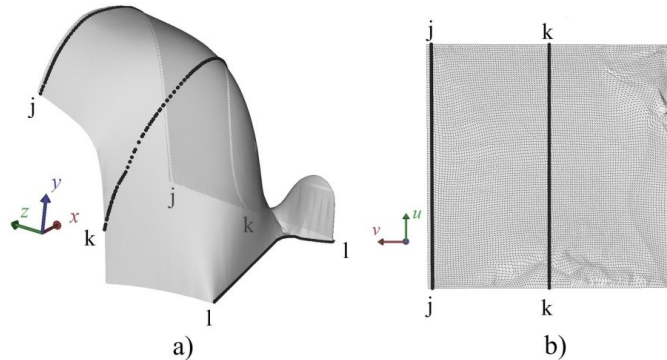


Figure 17. Lines  $j-j$ ,  $k-k$  and  $l-l$  illustrated in: a) physical space b) parametric space.



The procedure is illustrated on a half of the DTMB hull, scaled to the unit cube. The starting point is a triangulated surface with an unordered point cloud  $\mathbf{p}^U$  with physical coordinates, scaled to the unit cube as illustrated in Figure 18. The unordered points  $\mathbf{p}^U$  can be divided into two sets of points, the first set being the boundary points  $\mathbf{p}^B$  and the second set of points the remaining internal points  $\mathbf{p}^I$ .

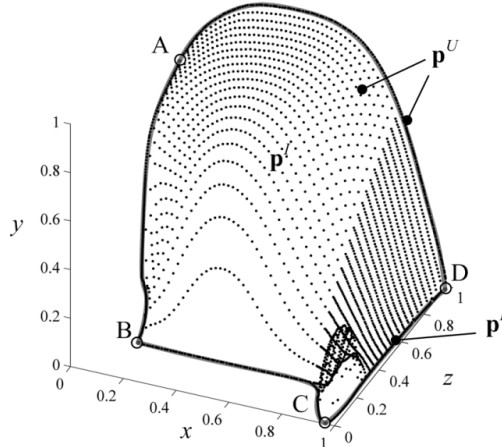


Figure 18. Point cloud: internal points, boundary points and corner points.

The first step is selecting four points on the geometry boundary  $\mathbf{p}^B$ , illustrated as small circles A,B,C and D in Figure 18. These points will be pre-set in the corners of the square  $u-v$  parametric domain as illustrated in Figure 19. The second step is moving the rest of the  $\mathbf{p}^B$  in-between the pre-set corner points, thus making a subset of projected boundary points  $\mathbf{p}'^B$  as illustrated with shaded line in Figure 19. Spacing of the  $\mathbf{p}'^B$  points in the  $u-v$  domain between the respective corner points is linearly correlated to the physical distances between the points.

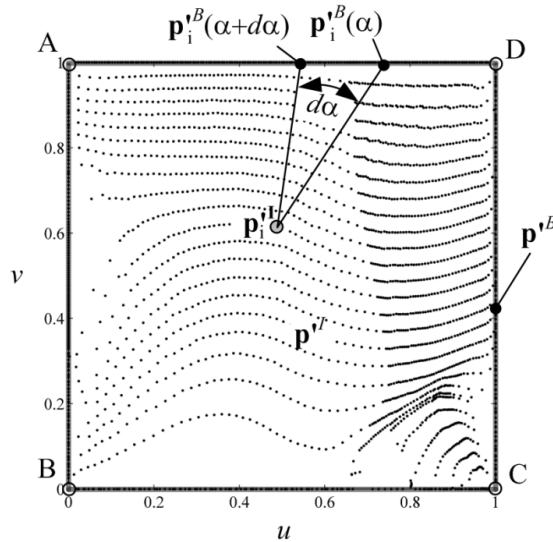


Figure 19. Point cloud projected to rectangular parametric domain.

Before the projection of the remaining (internal) points, designated  $\mathbf{p}^I$ , (Figure 19) additional values and functions have to be defined as followed. First a pre-projection of the point cloud has to be conducted, for the case of a ship hull a simple  $y$ -axis projection is sufficient as illustrated in Figure 20. Now the parameter of angle  $\alpha$  can be defined as the angle between the vector connecting the boundary point  $\mathbf{p}_j^B$  with the internal point  $\mathbf{p}_i^I$  and  $u_0$  axis in the pre-projection domain. Now the angle values  $\alpha p_{i,j}$  and distance values  $dp_{i,j}$  for each combination of boundary-internal point has to be calculated, pseudo-code as follows:

```

for i=1:nI
  for j=1:nB
     $\alpha p_{i,j} = \text{calculate\_angle}(\mathbf{p}_j^B, \mathbf{p}_i^I)$ 
     $dp_{i,j} = \text{calculate\_distance}(\mathbf{p}_j^B, \mathbf{p}_i^I)$ 
  end
end

```

where  $nI$  is the number of internal points and  $nB$  is the number of boundary points. The distances  $dp_{i,j}$  are calculated on the original triangulated surface by Dijkstra's method.

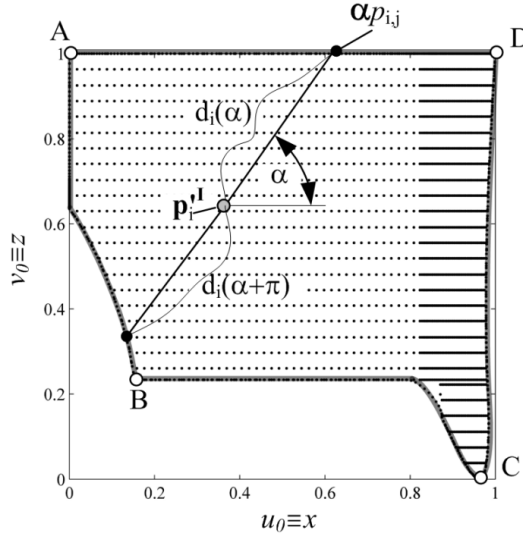


Figure 20. Definition of angle  $\alpha$  and angle-related distance.

The function set  $\mathbf{p}_i^B(\alpha)$  representing a function of projected boundary coordinates with respect to angle can now be created from  $\alpha p_{i,j}$  and the coordinates of points  $\mathbf{p}_i^B$  by interpolation. The function  $\mathbf{p}_i^B(\alpha)$  is different for each internal point since the boundary points are at different angular positions with respect to the internal point as illustrated in Figure 19. From  $\alpha p_{i,j}$  and  $dp_{i,j}$ , function  $d_i(\alpha)$  for each internal point can also be created by

interpolation. If two or more boundary points have the same angle value  $\alpha$ , points with larger distance  $d$  are ignored.

The projection of each internal point  $\mathbf{p}'_i$  is finally obtained by:

$$\mathbf{p}'_i = \frac{\int_{-\pi}^{\pi} f_P(d_i(\alpha), \alpha) \cdot \mathbf{p}'_i^B(\alpha) \cdot d\alpha}{\int_{-\pi}^{\pi} f_P(d_i(\alpha), \alpha) \cdot d\alpha} \quad (40)$$

where  $f_P$  is the projection operator, selected as:

$$f_P = 1 - \frac{d_i(\alpha)}{d_i(\alpha) + d_i(\alpha + \pi)} \quad (41)$$

Other projection operators can be used, but the above one has shown the best results. The value of  $f_P$  varies from 0 to 1 depending on the ratio of closest point an angle  $\alpha$  and angle  $\alpha + \pi$  (Figure 20). If the projection is calculated only by using two angles  $\alpha$  and  $\alpha + \pi$ , the projected point would be located on the line between  $\mathbf{p}'_i^B(\alpha)$  and  $\mathbf{p}'_i^B(\alpha + \pi)$ . The point would divide the line in ratio  $d_i(\alpha)/d_i(\alpha + \pi)$ . When integrated over all angles  $\alpha$ , a smooth projection can be obtained. The denominator in (41) represents sort of normalization, an equivalent is the number of points in the case when the projection is achieved with finite number of angles  $\alpha$  with spacing  $\Delta\alpha$ . After projection of unstructured points  $\mathbf{p}^U$  into the square parametric domain, the required matrix topology of point cloud is obtained by simple interpolation. The fitting procedure is continued according to (38).

## 3.2. Adaptive fitting

This section presents an iterative adaptive fitting procedure that was developed as a part of this doctoral thesis in paper [8]. The iterative method assumes that an initial solution exists obtained by fitting a parametric surface  $\mathbf{S}(u, v)$  to a functionally determined point-cloud  $\mathbf{F}(u, v)$ . The point-cloud  $\mathbf{F}(u, v)$  is obtained by using a spring-based projection method [67], [68]. The initial control points are later redistributed towards the desired areas by application of a scalar field called the relaxation field. The adaptive fitting iteration step is composed of evaluating the relaxation field based on the previous iteration, re-mapping and re-fitting.

### 3.2.1. Relaxation surface

First present the construction of the relaxation scalar field that will be used for subsequent re-parameterization. For the case of shape fitting an error metric can be used for the relaxation field. This error is calculated by using the results from the previous iteration. The error is defined as the distance between the point-cloud surface  $\mathbf{F}(u, v)$  and the selected parametric surface  $\mathbf{S}(u, v)$ :

$$E(u, v) = \|\mathbf{S}(u, v) - \mathbf{F}(u, v)\| \quad (42)$$

For the construction of the relaxation field used for, both magnitudes, the error and its gradient are used:

$$\varepsilon(\mathbf{u}, \mathbf{v}) = c_e \cdot E_{norm}^{abs}(\mathbf{u}, \mathbf{v}) + c_g \cdot \|\nabla E_{norm}^{sign}(\mathbf{u}, \mathbf{v})\| \quad (43)$$

where the exponent *abs* denotes that the absolute value was used, the exponent *sign* implies that the error surface is signed (can be both positive and negative depending on the normal vector on the surface) and *norm* is the subscript denoting normalization with respect to the corresponding maximum value. The gradient  $\nabla$  does not necessarily exist so it is calculated numerically by the central difference formula. The normalized error values are multiplied by a factor  $c_e$  while the normalized gradient magnitudes are multiplied by a factor  $c_g$ .

The error as defined in (42) does not represent the true distance between the parametric surface and the triangulated point-cloud mesh. To obtain the true distance (minimum distance between the point-cloud and the parametric surface), the squared error is minimized as a function of parametric coordinates for the parametric (spline) surface while keeping the point-cloud points constant:

$$E_T(\mathbf{u}_v, \mathbf{v}_v) = \sum_{j=0}^m \|\mathbf{S}(u_{v,j}, v_{v,j}) - \mathbf{F}(u_j, v_j)\|^2 \quad (44)$$

where  $(\mathbf{u}_v, \mathbf{v}_v)$  are vectors of parametric coordinates on the parametric surface which are to be found. This means that for each selected (and fixed) point  $\mathbf{F}(u_j, v_j)$  on the point-cloud surface, the nearest point on the parametric surface  $\mathbf{S}(u_{v,j}, v_{v,j})$  is to be found. This optimization problem was solved by a low memory BFGS [69], [70], a quasi-Newton method which is necessary since there is a large number ( $n_{var}=2 \cdot m$  and  $m$  is typically  $200 \times 200$ ) of optimization variables  $(\mathbf{u}_v, \mathbf{v}_v)$ . This method requires calculation of the derivative for each variable. The derivative of the squared error for a point  $\mathbf{S}(u_{v,j}, v_{v,j})$  with respect to the parametric coordinate  $u$  (and similarly to  $v$ ) is calculated by:

$$\frac{dE_T(u_{v,j}, v_{v,j})}{du} = \sum_{d=1}^3 \left[ 2 \cdot [\mathbf{S}_d(u_{v,j}, v_{v,j}) - \mathbf{F}_d(u_j, v_j)] \cdot \frac{d}{du} \mathbf{S}_d(u_{v,j}, v_{v,j}) \right] \quad (45)$$

where  $\mathbf{S}_d$  is  $d$ -th coordinate of parametric surface  $\mathbf{S}(u, v)$  and  $\mathbf{F}_d$  is the value of  $d$ -th coordinate of the point-cloud surface  $\mathbf{F}(u, v)$ . The term  $\frac{d}{du} \mathbf{S}_d$  is the derivative of the surface with respect to the parametric coordinates. For the case of the B-spline surface it is simply obtained by a lower order B-spline surface. This makes the minimization of the error (i.e. finding the true distance error) computationally feasible although still relatively time-consuming.

### 3.2.2. Adaptive re-mapping

Now the relaxation surface  $\varepsilon(u, v)$  can be applied. The relaxation is applied to the spring mesh [67], [68] corresponding to point-cloud  $\mathbf{F}(u, v)$ . Individual spring stiffness was calculated as reversely proportional to the spring length. The relaxation surface is then used for increasing the lengths of the springs i.e. reducing the edge stiffness by:

$$l = l_0 + l_0 \cdot \varepsilon(s_u, s_v) \quad (46)$$

where  $s_u$  and  $s_v$  are the parametric coordinates of the spring center in the previous step. After the stiffnesses are relaxed for all springs, the new point-cloud projection is calculated by solving the spring system equilibrium.

### 3.2.3. Flowchart of the procedure

The developed overall procedure is illustrated by the flowchart in Figure 21. The procedure uses a triangulated point-cloud geometry as input and this is used to create an initial point-cloud mapping which is used for creating the initial functionally determined point-cloud  $\mathbf{F}(u,v)$ . Based on this initial mapping, the initial parametric surface  $\mathbf{S}(u,v)$  fitting is conducted. Now the relaxation surface can be calculated based on a selected error metric between the point-cloud  $\mathbf{F}(u,v)$  and the parametric surface  $\mathbf{S}(u,v)$ . This error is used as input for the spring mesh relaxation in the adaptive re-mapping step which results in a new functionally determined point-cloud  $\mathbf{F}(u,v)$ . The final step is fitting the parametric surface to the new point-cloud surface  $\mathbf{F}(u,v)$ . The last three steps of the procedure (calculation of the relaxation surface, adaptive re-mapping and re-fitting) can be conducted only once or repeated multiple times until the chosen stopping criteria are met. After each iteration step, the parametric surface and the point-cloud mapping are modified which results in a different relaxation surface which will again lead to a modified parametric surface. The stopping criterion used here was simply a predefined fixed number of iterations. If the number of iterations is not fixed, the procedure converges to a fixed solution in some cases while in some cases the procedure gradually diverges. This convergence will be investigated further in future work.

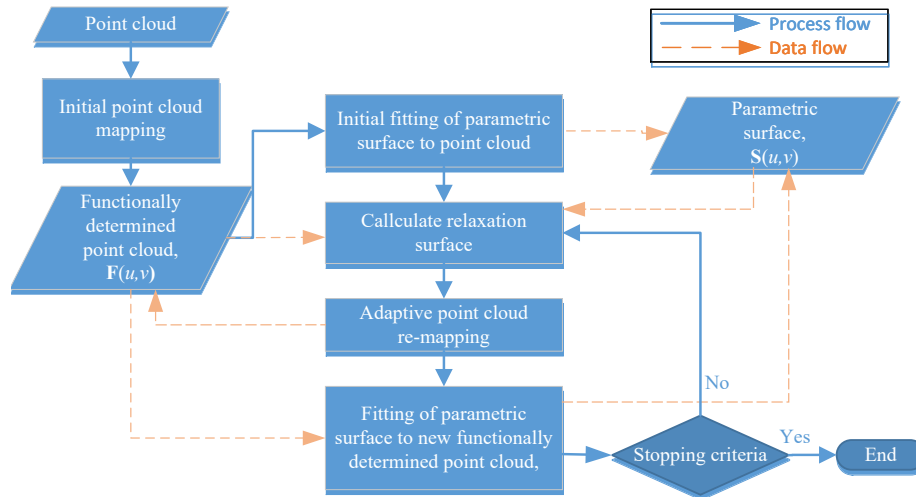


Figure 21. Flow chart of the adaptive fitting parametric surface.

### 3.3. Engineering application of parametric shape fitting

This section gives a summary of several considered engineering applications of parametric shape fitting methods.

### 3.3.1. Parameterization testing by enhanced shape fitting

Parametric shape fitting is considered important for testing of parameterizations on existing shapes (point-cloud) as this way a parameterization can be tested geometrically without including computationally expensive numerical simulation. This kind of testing was conducted (in addition to the shape optimization) in the paper [6] where several different shape parameterizations were tested on several different ship hull shapes.

### 3.3.2. Boat hull adaptive fitting using B-spline and T-spline

This section compares the performance of B-spline and T-spline surfaces for parameterization of a boat hull using the adaptive fitting procedure. The results of the adaptive fitting of the B-spline with a 15x20 CPs grid and a 15x9 CPs grid to boat hull surface are illustrated in Figure 22. The effect of adaptive fitting can be observed – the CPs are concentrated at the geometric features. In this case this means that the CPs are concentrated near the sharp edges while they move away from the flat areas such as the flat bottom. When using the 15x9 CPs grid, the overall geometry is represented fairly but still the sharp edges are not clearly visible. With an increase to the 15x20 CPs grid the sharp edges become visible. Still, the spike on the bow where multiple sharp edges come together is not clearly noticeable.

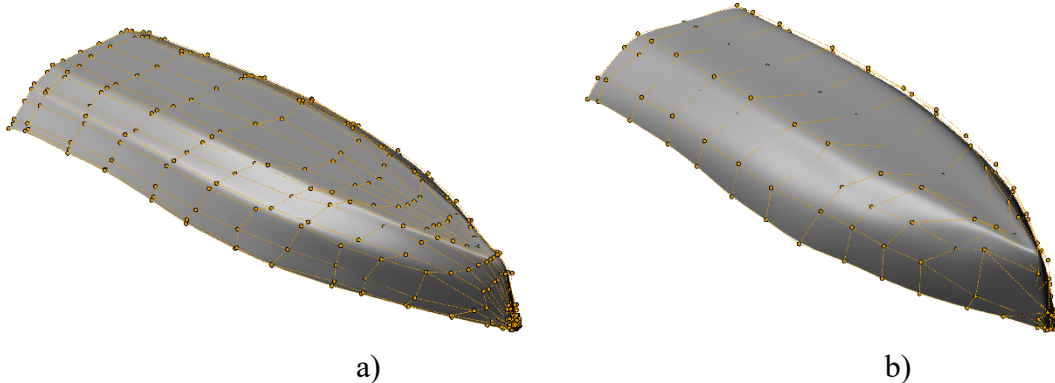


Figure 22. Results of fitting using the developed adaptive re-mapping for: a) B-spline with 15x20 CPs grid, b) B-spline with 15x9 CPs grid

Next, the evaluation of T-spline surface is conducted. At first, fitting using a T-spline surface with 15x20 control points (equivalent to fitting a B-spline surface) was used as the initial solution. Next, as a simple test of T-spline fitting, just two additional control points were added at the location of the largest error. This was conducted in two subsequent iteration steps. In the procedure flowchart (Figure 21) this addition of CPs would appear after the fitting of parametric surface  $\mathbf{P}(u,v)$  to new point-cloud  $\mathbf{F}(u,v)$ , and before the calculation of the relaxation surface step. This means that before and after each CP addition, the adaptive re-mapping step was conducted.

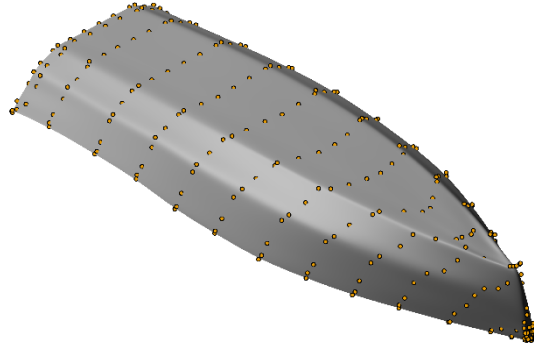


Figure 23. Fitting results

A comparison of the error when using  $15 \times 9$  and  $15 \times 20$  CPs B-spline surfaces with the  $(15 \times 20 + 2)$  CPs T-spline is illustrated in Figure 24. When comparing two B-spline cases, an increase of the number of control points by 165 reduces the maximum error from 24 mm to 19 mm. When the T-spline is used starting from the  $15 \times 20$  CPs B-spline with just two additional CPs, a reduction of the maximum error value from 19 mm to 12 mm is obtained. The error is reduced at the location of the maximum value while at the rest of the fitted surface the fitting error remains the same. Although the error is reduced by application of the T-spline surface in comparison to B-spline surface, the T-spline requires an additional algorithm for CP insertion. This limits the application of standard optimization methods.

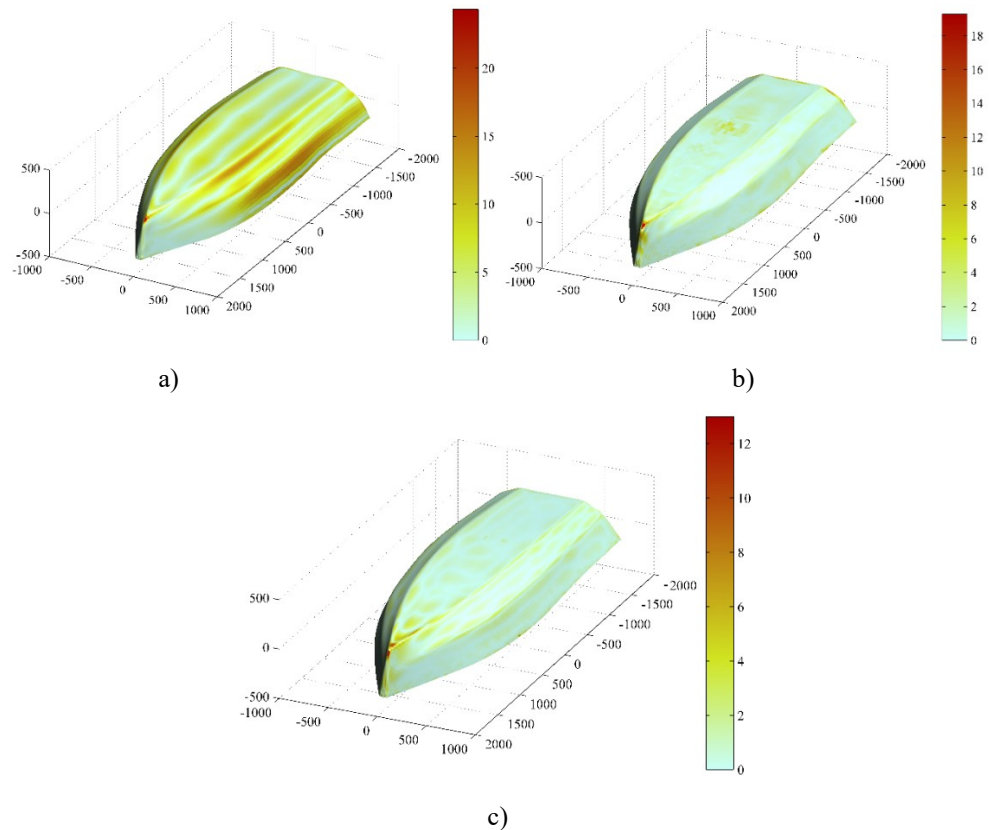


Figure 24. Comparison of error distributions for: a) B-spline with  $15 \times 9$  CPs grid, b) B-spline with  $15 \times 20$  CPs grid and b) T-spline with  $15 \times 20$  CPs grid and additional control points

### 3.3.3. Sensitivity based re-parameterization

The previous part of this section has shown how to apply the adaptive fitting procedure for geometric fitting. This section will show that the same procedure can be applied for control point redistribution in shape optimization. The idea is presented on a topology optimization example in which the density of material is optimized using a variant of gradient based optimization, so called SIMP approach [71]. The solution represented as material density distribution for the selected problem using the SIMP approach is shown in Figure 25a.

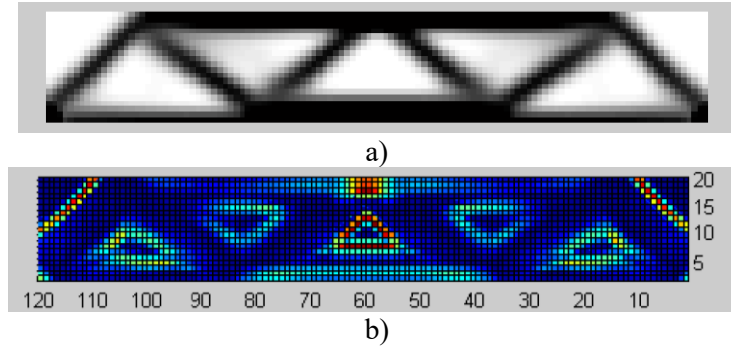


Figure 25. Topologically optimized structure a) distribution of material density b) magnitude of the sensitivity field gradient.

Instead of standard SIMP approach, here a B-spline surface was used to describe the density of the material and the optimization variables were the B-spline control points. It is appropriate to choose only the  $z$  coordinate of the control-points as optimization variables. In the same way to as in the SIMP method, a gradient based optimization was conducted. Optimization result (B-spline surface) for the case without the adaptive fitting procedure is shown in Figure 26a. An enhanced procedure is developed which conducts the adaptive control-point redistribution using the sensitivity field (Figure 25b shows the gradient magnitude of the sensitivity field). Conducting this adaptive fitting procedure several times during the optimization results in better control-point distribution (Figure 26b) in comparison to the first optimization case.

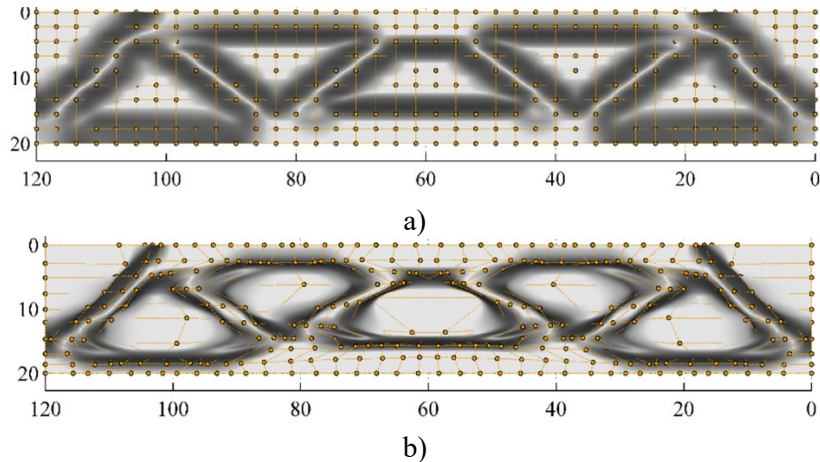


Figure 26. Density field described by B-spline surface: a) solution with constant  $x$ - $y$  control-point coordinates and b) solution with constant  $x$ - $y$  control-point coordinates and adaptive redistribution in  $x$ - $y$  direction.



### 3.3.4. Application to multi-patch parameterizations

For simple geometric shapes, such as half-sphere, there is no need for multi-patch surfaces as illustrated in Figure 27.

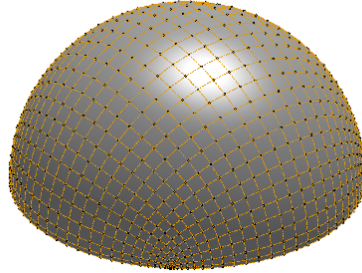


Figure 27. Single-patch B-spline surface describing a half-sphere.

However, some geometric shapes cannot easily be described by a single-patch surface. An example is a half-cube which is composed of five connected patches as illustrated in Figure 28a. The adaptive fitting can be used in these cases to reduce a multi-patch parameterization to a single patch parameterization as shown in Figure 28b.

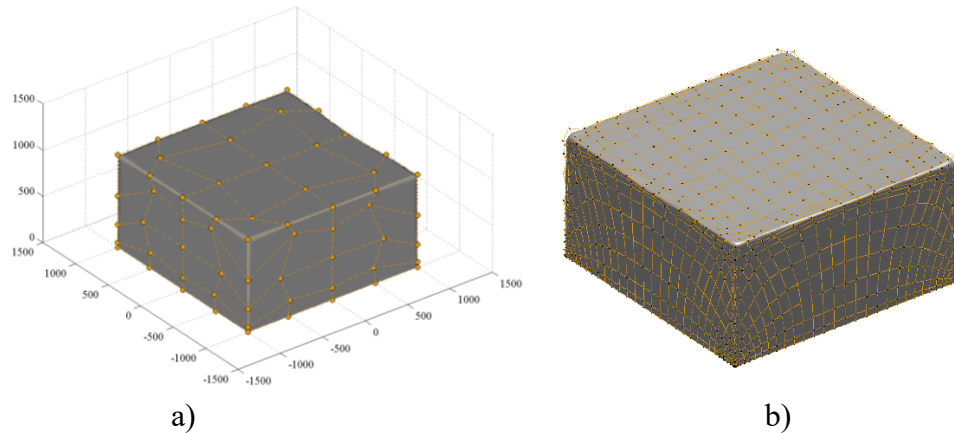


Figure 28. Half-cube described by: a) multi-patch parameterization with 5 B-spline surfaces and b) Single-patch B-spline surface obtained by adaptive fitting

Some more complex shapes are even harder to describe using a single-patch surface. The example is a two-bladed propeller which usually requires many mutually connected patches as shown in Figure 29a. The proposed adaptive fitting procedure can be used even in this complex case to reduce the multi-patch parameterization to a single patch B-spline surface as illustrated in Figure 29b.

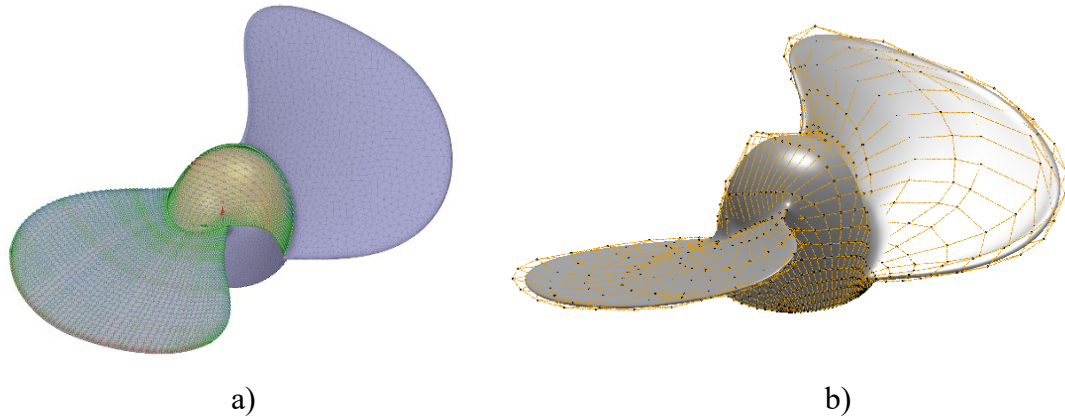


Figure 29. Two-bladed propeller described by: a) multi-patch parameterization (only two out of many B-spline patches are shown) and b) Single-patch B-spline surface obtained by the adaptive fitting.

This kind of fitting is useful on its own but it allows additional possibilities. Usually, differently patched shape parameterizations can be modified on their own, but they cannot be mixed together. But as it was shown, the adaptive fitting enables the reduction of complex multi-patch surface to a good single-patch approximation. This could allow the optimizer to use standard optimization methods for the cases where various patch configurations exist. For example, once the reduction to single-patch is conducted, the genetic crossover operator can easily be used without additional procedures. Otherwise, the various parameterizations (multi-patch, single-patch) would have different chromosomes preventing application of the genetic operators. Several crossover examples are shown in Figure 30.

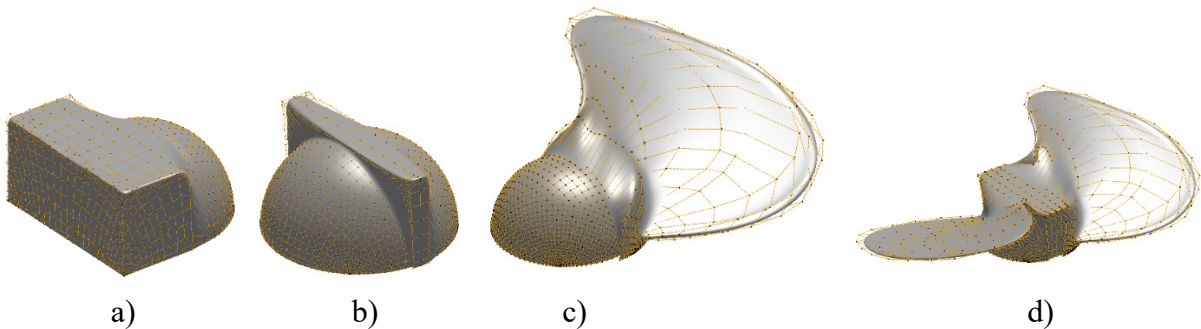


Figure 30. Crossover of single-patch surfaces: a) crossover of half-sphere and half-cube at chromosome mid-point and b) two-point crossover of half-sphere and half-cube c) crossover of half-sphere and boat propeller at chromosome mid-point and d) two-point crossover of boat propeller and half-cube

## 4. SUMMARY OF COMPLEMENTARY PAPERS

This section presents short summaries of the complementary papers and the respective scientific contribution.

### 4.1. Paper 1: Multi-regime shape optimization of fan vanes for energy conversion efficiency using CFD, 3d optical scanning and parameterization

The first paper is titled: Multi-regime shape optimization of fan vanes for energy conversion efficiency using CFD, 3d optical scanning and parameterization. The topic of the paper is shape optimization of a centrifugal roof fan. The first part of the paper consists of experimental tests in which the performance of the fan was measured. Next, a CFD model for simulation of the centrifugal fan performance was developed and validated with the experimental data. In this paper, a B-spline curve was used for the development of a generic shape optimization procedure. The original contribution of the paper is the analysis of B-spline application for centrifugal fan optimization. The paper developed new vane shapes which can potentially improve the efficiency of this kind of machines. The paper also includes the development of an original optimization workflow for optimization of the centrifugal fan subjected to multiple working conditions which could appear during the operation lifetime.

#### 4.1.1. PhD candidate contribution

Marinić-Kragić initiated the paper by conducting several different variations of the centrifugal fan shape optimization and obtaining promising results. Marinić-Kragić wrote the part of the paper regarding the shape parameterization, integration of different software tools and the results. Marinić-Kragić has also developed an original numerical workflow for optimization of centrifugal fan subject to variable operating conditions.

### 4.2. Paper 2: 3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise related excellence criteria and constraints

The second paper is titled: 3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise related excellence criteria and constraints. This paper includes a more detailed investigation of possible energy efficiency improvements for centrifugal fans. In comparison to previous researchers, the original contribution is the usage of a completely generic B-spline surface in combination with multi-regime optimization. The result of the paper are several new 3D shapes which could additionally improve the centrifugal fan energy efficiency. The paper also includes an elementary noise analysis and multi-objective optimization. By conducting a multi-objective optimization, it was shown that a compromise between the energy efficiency and noise emissions could be obtained. It was shown that a hybrid optimization method that combines the GA in the initial phase and the Nelder-Mead method in the final phase leads to fast convergence to the global optimum.

#### 4.2.1. PhD candidate contribution

Marinić-Kragić initiated the paper by extending the optimization cases from the previous paper by using a generic 3D B-spline surface. Marinić-Kragić has written the largest part of the

paper. In this paper, Marinić-Kragić has investigated the possibilities of improvements in energy efficiency conversion for centrifugal fans in more detail compared to the previous paper. Marinić-Kragić also developed the numerical implementation.

### **4.3. Paper 3: Numerical models for robust shape optimization of wind turbine**

The paper C is titled: Numerical models for robust shape optimization of wind turbine blades. The topic of this paper are numerical models for analysis of horizontal-axis wind turbines, which includes the analysis of pre-existing shape parameterization methods. The paper also develops original shape parameterization methods which offer additional freedom of shape. It was shown that the original parameterization can be adjusted to site specific wind conditions described by a statistical distribution. The paper finally shows how to integrate all required elements in a numerical shape optimization workflow and presents several preliminary results.

#### **4.3.1. PhD candidate contribution**

In this paper, Marinić-Kragić analyzed the existing methods for wind-turbine blade shape parameterization. Furthermore, Marinić-Kragić has showed that by using an original method of shape parameterization, greater freedom of shape and better adaptivity of the blade shape to specific location conditions can be achieved. Marinić-Kragić used a CFD model of a wind turbine and compared the results with the experimental data. Marinić-Kragić has showed the method for integrating all elements of the shape optimization procedure in a numerical workflow for wind-turbine shape optimization. Marinić-Kragić has written the sections of paper regarding shape parameterization, optimization procedures and the results.

### **4.4. Paper 4: Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes**

The paper four is titled: Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes. This paper is concerned with specifics of vertical-axis wind turbine (VAWT) optimization and shape parameterization. In comparison to the previous paper dealing with HAWTs, VAWT-s appear in far more mutually very different shapes. This paper develops an original method for shape optimization of VAWTs. The developed method can be used in an optimization procedure which starts from a random shape and converges towards different shapes depending on a selected operating condition (tip speed ratio and wind speed). This way, the computer can independently and without an initial solution numerically synthesize a VAWT shape designed for a specific location and operating conditions. The paper also compares the results obtained by using local and global shape optimization and it was shown that global shape optimization leads to designs with better energy conversion efficiency. Several optimization algorithms were tested, but GA algorithms have shown the best results.

#### **4.4.1. PhD candidate contribution**

Marinić-Kragić extended the methods from the previous paper (concerning HAWTs) to VAWTs for which Marinić-Kragić has developed an original shape parameterization method.

On several numerical examples, Marinić-Kragić has shown the advantage of this method in comparison to classical optimization which starts from an initial solution. Marinić-Kragić written and presented the paper at EWEA 2015 conference and at ESTECO UM16 conference.

#### **4.5. Paper 5: Reverzno inženjerstvo i dvo-stupanjska optimizacija broskog vijka i sapnice pomoću B-spline ploha**

The fifth paper is titled: “Reverzno inženjerstvo i dvo-stupanjska optimizacija broskog vijka i sapnice pomoću B-spline ploha”. This paper analysis the application of B-spline surfaces for optimization of a ducted boat propeller. The paper proposed that the optimization procedure can be conducted in two steps, first is the optimization with fixed propeller and following is the optimization of both the propeller and the duct simultaneously. The original contribution of the paper is the usage of B-spline surfaces for ducted boat propeller optimization.

##### **4.5.1. PhD candidate contribution**

This paper was mostly the work of graduate student Bagavac, where Marinić-Kragić has assisted in writing the paper and in technical details of coupling the B-spline shape parameterization with the CFD analysis.

#### **4.6. Paper 6: Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape**

The paper six is titled: Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape. In this paper, an original method for ship parameterization was developed. The parameterization method was tested using the enhanced fitting procedure (section 3.1) and it was shown that the method achieves a reduction in the number of variables and improves the shape generality. The parameterization method enables both shape and topology changes. The paper also shows how to integrate the procedure in a multi-disciplinary optimization which includes structural and flow analysis (construction price and hydrodynamic resistance). The result of the optimization is thus multitude of Pareto-optimal solutions which allows the decision maker to select the desired compromise between the construction price and hydrodynamic resistance.

##### **4.6.1. PhD candidate contribution**

Marinić-Kragić applied the mentioned original shape parameterization method for ship hull optimization. Marinić-Kragić has conducted a comparison of several shape parameterization methods with a variable number of control points using three mutually different hull shapes. Also, Marinić-Kragić has shown how to integrate the proposed parameterization as a part of a shape optimization workflow which includes flow analysis and structural analysis. Marinić-Kragić wrote the paper and developed the numerical implementation.

#### **4.7. Paper 7: Numerical analysis of energy efficiency performance and noise emissions of building roof fan**

The seventh paper is titled: Numerical analysis of energy efficiency performance and noise emissions of building roof fan. The paper compared various numerical models for energy efficiency performance and noise emissions predictions of building roof fan. It can be concluded that computationally efficient numerical models are required for accurate simulation of energy efficiency performance while accurate noise prediction requires high fidelity models. The results can be used to select which numerical models can be used in various stages of the optimization procedure.

##### **4.7.1. PhD candidate contribution**

Marinić-Kragić has written the paper and developed the numerical implementation. This includes the numerical implementation of the several different numerical CFD methods, and programming required for noise analysis based on the CFD results.

#### **4.8. Paper 8: Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting**

The seventh paper is titled: Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting. The paper presents the developed adaptive parameterization method. The paper introduces a re-parameterization method which can be used in combination with any scalar field. The method can be used in both geometric fitting and shape optimization. Several results show that the method is applicable for geometric fitting of mutually very different geometries. The method was compared to similar feature sensitive parameterization and it was shown that the developed method results in smoother surfaces that are suited for application in shape optimization.

##### **4.8.1. PhD candidate contribution**

Marinić-Kragić developed a novel re-parameterization method that smoothly re-distributes the B-spline control points towards the desired areas. Marinić-Kragić has written the paper and developed the numerical implementation. This includes the programming required for implementation of the method, testing the method on several mutually very different geometries and comparing the method to a similar method found in the literature.

## 5. CONCLUSION

Engineering shape optimization with integrated numerical simulations is a demanding multidisciplinary problem and various methods for conducting the optimization procedure exist. While various numerical simulations could be required, this doctoral thesis considers mostly CFD because of its wide usage in combination with shape optimization. Also, CFD involves problems such as time-consuming simulations and non-linear governing equations, making it a representative of common engineering numerical simulation. By conducting several different test cases [1]–[6], it was shown that both global and local optimization methods are required to efficiently solve an engineering optimization problem. After the investigation of various optimization methods, two approaches could be highlighted: gradient sensitivity-based method (as used in section 3.3.3), appealing because of its computational efficiency; and an integrated numerical workflow based on genetic-algorithms (GA), attractive since it offers a generic approach to wide array of optimization problems. While the sensitivity-based method offers a promising solution with fast optimization times, it has several disadvantages. First, it is not suitable for global optimization since it is by nature gradient-based. Since the objective function(s) and constraints are highly nonlinear and can contain noise, it is common that even a local optimum will not be reached from a selected initial solution. Engineering optimization can also contain discrete variables which cannot be solved by classical gradient methods. Furthermore, engineering optimization problems usually involve several objectives, making them suitable for GA based methods. A generic engineering shape optimization task can be solved by linking GA with shape parameterization methods, and (multiple) engineering simulation nodes in an integrated numerical workflow. Nevertheless, fast gradient based methods could eventually be implemented when the GA obtains a solution close to the global optimum. When using the GA, each generated shape needs to be subject to a time consuming numerical simulation. One of the major concerns in this kind of optimization is that a large number of simulations requires great computational effort.

One of the most important parts of the optimization problem is selecting a suitable parameterization method. If parameterization is selected such that the respective geometry can be described with a small number of parameters while keeping the shape generality, the number of necessary simulations can be reduced thus allowing practical realizations of the optimization procedure. Many different shape parameterization methods were considered as a part of this thesis and several novel methods were developed for specific applications. The first shape parameterization method considered in this thesis was the B-spline curve. In [1], B-spline curves were used for centrifugal fan vane 2D shape parameterization, while the 3D shape was constructed by extruding a 2D geometric entity. Generic parameterization method based on the B-spline curve enables both constant and variable vane thickness. Following the promising application of B-spline curves, a fully generic B-spline surface was used in [2]. Each B-spline control point has 3 degrees of freedom, and the B-spline requires at least 4x4 control point grid surface (3<sup>rd</sup> order B-spline with clamped ends was used). If all possible degrees of freedom were

used, this would result in 48 degrees of freedom which is not appropriate for global shape optimization. A parameterization method was developed such that there is only one degree of freedom per control point but an additional B-spline surface was used for trimming the leading edge. This way the number of degrees of freedom was reduced while keeping the shape generality. It was shown that both single and multi-objective optimization can be conducted using the developed procedure. However, accurate noise prediction requires high fidelity models which were investigated in [7]. The developed parameterization method is appropriate for constant thickness vanes. It was also interesting to note that the shape obtained by the 2D shape parameterization in [1] looks like the average of individual 2D cross-sections of the shape obtained by the 3D shape parameterization in [2].

An example of a complex shape which requires different shape parameterization is the wind turbine blade. Wind turbines can generally be divided in horizontal axis wind turbines (HAWT) and vertical axis wind turbines (VAWT). As it was shown in [3], [4], HAWT parameterization is comparatively simpler as the shape of the blade is airfoil-based with span-wise scaling and rotation. Several HAWT parameterization methods were considered in [3]. It was shown that the generic B-spline surface can be used in an optimization procedure to obtain superior results in comparison to an airfoil-based parameterization. Nevertheless, the applicability is limited by the CFD accuracy and currently available computational resources.

In comparison to HAWT parameterization, VAWT parameterization should cover a wider array of shapes such as the Savonius and the Darrieus VAWT designs. It was shown in [4] that it is possible to devise such parameterization which can cover both the Savonius and the Darrieus designs using only 15 shape parameters. The proposed parameterization method was successfully used and it was shown that autonomous design synthesis is possible. A similar shape parameterization method was used in [5] for a ducted boat propeller parameterization.

A very different object that is commonly subject to shape optimization is ship hull. This also means that a different shape parameterization method is required. For this purpose a novel method that is able to describe both shape and topology changes was developed in [6]. The method was tested on three different boat shapes and it was shown that ship hull shape can be described with much fewer shape variables compared to a standard full 3D B-spline surface. Also, it was shown that the method is applicable to multi-objective optimization and has an ability to start from an initially random shape.

Generally, it can be concluded that B-spline surfaces can be used for generic shape parameterization in various different engineering problems, but using all the available B-spline degrees of freedom would still result in a large amount of optimization variables. Thus, modified parameterizations should be developed depending on the specific problem. Afterwards, autonomous design synthesis can be achieved even with modest computational resources.

A more generic parameterization approach called the adaptive re-parameterization was developed in [8]. This approach is not application-specific and can potentially be used in many engineering shape optimization problems. The adaptive re-parameterization requires the selection of a scalar field which is used to re-distribute the B-spline control points towards the



desired area. If this field is selected appropriately, the B-spline control points will have a dense distribution in areas with complex geometry by removing the control points from simpler regions (such as flat or constant-curvature regions) in which they are not required. This permits the usage of fewer control points thus reducing the number of optimization variables while keeping the shape generality. A structural optimization case was shown in this thesis as an example where the sensitivity field with respect to the structural compliance was used as the scalar field. It was shown that the adaptive re-parameterization method could also be used for more complex cases which otherwise require a surface composed of multiple B-spline patches. These cases give rise to several problems regarding the optimization process. The main problem is the optimal number of patches and the way in which they should be connected. To solve this problem, an intermediate method that uses single patch B-spline surface fitted by the adaptive re-parameterization procedure can be used. Once a multi-patch surface is converted to an equivalent single-patch surface, standard optimization algorithms can be used as shown in section 3.3.4.

### **5.1. Ongoing and future work**

This doctoral thesis showed several possibilities for the reduction of the number of variables in engineering shape optimization problems. Still, there is no universal shape parameterization and this leaves room for further improvements. Parametric shape fitting methods were shown to be applicable to existing shapes as a method for evaluating individual shape parameterization methods but this might even be conducted during optimization, which allows for switching between different parameterization methods. It was also shown that adaptive fitting can be used for redistribution of the control-points towards the desired areas which allows for reduction of the number of shape variables and increased shape resolution in needed areas. Although several test were conducted, additional tests are required to verify if the method is more universally applicable. The possibility of optimization using different patch topologies was also shown but an actual optimization case has yet to be conducted.

## BIBLIOGRAPHY

- [1] Z. Milas, D. Vučina, and I. Marinić-Kragić, “Multi-regime shape optimization of fan vanes for energy conversion efficiency using CFD, 3D optical scanning and parameterization,” *Eng. Appl. Comput. Fluid Mech.*, vol. 8, no. 3, pp. 407–421, 2014.
- [2] I. Marinić-Kragić, D. Vučina, and Z. Milas, “3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise-related excellence criteria and constraints,” *Eng. Appl. Comput. Fluid Mech.*, vol. 10, no. 1, pp. 210–228, 2016.
- [3] D. Vučina, I. Marinić-Kragić, and Z. Milas, “Numerical models for robust shape optimization of wind turbine blades,” *Renew. Energy*, vol. 87, pp. 849–862, 2015.
- [4] I. Marinić-Kragić, D. Vučina, and Z. Milas, “Numerical workflow for 3D shape optimization and synthesis of vertical-axis wind turbines for specified operating regimes,” in *European Wind Energy Association Annual Conference and Exhibition 2015*, 2015.
- [5] I. Marinić-Kragić, P. Bagavac, and I. Peh nec, “Reverzno inženjerstvo i dvo-stupanjska optimizacija broskog vijka i sapnice pomoću B-spline ploha,” in *Sedmi susret hrvatskog društva za mehaniku*, 2016, pp. 121–126.
- [6] I. Marinić-Kragić, D. Vučina, and M. Ćurković, “Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape optimization workflow,” *Comput. Des.*, vol. 80, pp. 61–75, Nov. 2016.
- [7] I. Marinić-Kragić, Z. Milas, and D. Vučina, “Numerical analysis of energy efficiency performance and noise emissions of building roof fan,” in *9th International Exergy, Energy and Environment Symposium (IEEES-9)*, 2017.
- [8] I. Marinić-Kragić, M. Ćurković, and D. Vučina, “Adaptive re-parameterization based on arbitrary scalar fields or shape optimization and surface fitting,” *Submitt. Publ. (currently under Rev.*
- [9] O. P. Bijan Mohammadi, *Applied shape optimization for fluids*, 2ed. OUP, 2010.
- [10] J. Haslinger and R. a E. Mäkinen, *Introduction to Shape Optimization*, vol. 7. Society for Industrial and Applied Mathematics, 2003.
- [11] J. C. De los Reyes, *Numerical PDE-Constrained Optimization*. Cham: Springer International Publishing, 2015.
- [12] L. Blank, M. H. Farshbaf-shaker, H. Garcke, C. Rupprecht, and V. Styles, *Trends in PDE Constrained Optimization*, vol. 165. 2014.
- [13] D. Thévenin and G. Janiga, *Optimization and Computational Fluid Dynamics*, vol. XVI. 2008.
- [14] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Struct. Multidiscip. Optim.*, vol. 26, no. 6, pp. 369–395, 2004.
- [15] M. Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Comput. Methods Appl. Mech. Eng.*, vol. 192, no. 1–2, pp. 227–246, 2003.
- [16] I. Y. Kim and O. L. de Weck, “Variable chromosome length genetic algorithm for

- progressive refinement in topology optimization,” *Struct. Multidiscip. Optim.*, vol. 29, no. 6, pp. 445–456, Jun. 2005.
- [17] H. Stringer, “Behavior of variable-length genetic algorithms under random selection,” University of Florida, 2007.
- [18] D. a. Tortorelli and P. Michaleris, “Design sensitivity analysis: Overview and review,” *Inverse Probl. Sci. Eng.*, vol. 1, no. 1, pp. 71–105, 1994.
- [19] G. Allaire, “A review of adjoint methods for sensitivity analysis , uncertainty quantification and optimization in numerical codes,” *Ingénieurs de l’Automobile*, vol. 836, pp. 33–36, 2015.
- [20] M. B. Giles and N. A. Pierce, “An introduction to the adjoint approach to design,” *Flow, Turbul. Combust.*, vol. 65, no. 3–4, pp. 393–415, 2000.
- [21] O. Pironneau, “On optimum design in fluid mechanics,” *J. Fluid Mech.*, vol. 64, no. 1, p. 97, Jun. 1974.
- [22] S. Nadarajah, P. Castonguay, and A. Mousavi, “Survey of Shape Parameterization Techniques and its Effect on Three-Dimensional Aerodynamic Shape Optimization,” in *18th AIAA Computational Fluid Dynamics Conference*, 2007, no. June, pp. 1–23.
- [23] J. A. Samareh, “Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization,” *AIAA J.*, vol. 39, no. 5, pp. 877–884, May 2001.
- [24] D. Vucina, Z. Lozina, and I. Pehcec, “Computational procedure for optimum shape design based on chained Bezier surfaces parameterization,” *Eng. Appl. Artif. Intell.*, vol. 25, no. 3, pp. 648–667, Apr. 2012.
- [25] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 2002.
- [26] S. Nižetić, F. Grubišić- Čabo, I. Marinić-Kragić, and A. M. Papadopoulos, “Experimental and numerical investigation of a backside convective cooling mechanism on photovoltaic panels,” *Energy*, vol. 111, pp. 211–225, Sep. 2016.
- [27] ANSYS Inc, “ANSYS (computer program).” Ansys Inc., 2016.
- [28] E. Hardee, K.-H. Chang, J. Tu, K. K. Choi, I. Grindeanu, and X. Yu, “A CAD-based design parameterization for shape optimization of elastic solids,” *Adv. Eng. Softw.*, vol. 30, no. 3, pp. 185–199, Mar. 1999.
- [29] Esteco s.p.a, “modeFRONTIER (computer program).” Esteco s.p.a, 2016.
- [30] T. Ray, R. P. Gokarn, and O. P. Sha, “A global optimization model for ship design,” *Comput. Ind.*, vol. 26, no. 2, pp. 175–192, May 1995.
- [31] Y. Tahara, D. Peri, E. F. Campana, and F. Stern, “Computational fluid dynamics-based multiobjective optimization of a surface combatant using a global optimization method,” *J. Mar. Sci. Technol.*, vol. 13, no. 2, pp. 95–116, May 2008.
- [32] H. Kim and C. Yang, “A new surface modification approach for CFD-based hull form optimization,” *J. Hydrodyn.*, vol. 22, no. 5 SUPPL. 1, pp. 503–508, 2010.
- [33] Y. Tahara, D. Peri, E. F. Campana, and F. Stern, “Single- and multiobjective design optimization of a fast multihull ship: numerical and experimental results,” *J. Mar. Sci.*

*Technol.*, vol. 16, no. 4, pp. 412–433, 2011.

- [34] S. Jeong and H. Kim, “Development of an Efficient Hull Form Design Exploration Framework,” *Math. Probl. Eng.*, vol. 2013, pp. 1–12, 2013.
- [35] H. Cui and O. Turan, “Application of a new multi-agent Hybrid Co-evolution based Particle Swarm Optimisation methodology in ship design,” *Comput. Des.*, vol. 42, no. 11, pp. 1013–1027, Nov. 2010.
- [36] H. Cui, O. Turan, and P. Sayer, “Learning-based ship design optimization approach,” *Comput. Des.*, vol. 44, no. 3, pp. 186–195, Mar. 2012.
- [37] E. Sarioz, “An optimization approach for fairing of ship hull forms,” *Ocean Eng.*, vol. 33, no. 16, pp. 2105–2118, 2006.
- [38] F. Pérez and J. A. Clemente, “Constrained design of simple ship hulls with -spline surfaces,” *Comput. Des.*, vol. 43, no. 12, pp. 1829–1840, Dec. 2011.
- [39] H. J. Koelman and B. N. Veelo, “A technical note on the geometric representation of a ship hull form,” *Comput. Des.*, vol. 45, no. 11, pp. 1378–1381, Nov. 2013.
- [40] J. L. Loth and H. McCoy, “Optimization of Darrieus Turbines with an Upwind and Downwind Momentum Model,” *J. Energy*, vol. 7, no. 4, pp. 313–318, Jul. 1983.
- [41] M. Abdul Akbar and V. Mustafa, “A new approach for optimization of Vertical Axis Wind Turbines,” *J. Wind Eng. Ind. Aerodyn.*, vol. 153, pp. 34–45, Jun. 2016.
- [42] S. Joo, H. Choi, and J. Lee, “Aerodynamic characteristics of two-bladed H-Darrieus at various solidities and rotating speeds,” *Energy*, vol. 90, no. January 2016, pp. 439–451, Oct. 2015.
- [43] T. J. Carrigan, B. H. Dennis, Z. X. Han, and B. P. Wang, “Aerodynamic Shape Optimization of a Vertical-Axis Wind Turbine Using Differential Evolution,” *ISRN Renew. Energy*, vol. 2012, pp. 1–16, 2012.
- [44] M. Jafaryar, R. Kamrani, M. Gorji-Bandpy, M. Hatami, and D. D. Ganji, “Numerical optimization of the asymmetric blades mounted on a vertical axis cross-flow wind turbine,” *Int. Commun. Heat Mass Transf.*, vol. 70, pp. 93–104, Jan. 2016.
- [45] C. S. Ferreira and B. Geurts, “Aerofoil optimization for vertical-axis wind turbines,” *Wind Energy*, vol. 18, no. 8, pp. 1371–1385, Aug. 2015.
- [46] G. Bedon, S. De Betta, and E. Benini, “Performance-optimized airfoil for Darrieus wind turbines,” *Renew. Energy*, vol. 94, pp. 328–340, Aug. 2016.
- [47] Y.-T. Lee and H.-C. Lim, “Numerical study of the aerodynamic performance of a 500 W Darrieus-type vertical-axis wind turbine,” *Renew. Energy*, vol. 83, pp. 407–415, Nov. 2015.
- [48] G. Bedon, M. Raciti Castelli, and E. Benini, “Proposal for an innovative chord distribution in the Troposkien vertical axis wind turbine concept,” *Energy*, vol. 66, pp. 689–698, Mar. 2014.
- [49] X. Shen, H. Yang, J. Chen, X. Zhu, and Z. Du, “Aerodynamic shape optimization of non-straight small wind turbine blades,” *Energy Convers. Manag.*, vol. 119, pp. 266–278, Jul. 2016.

- [50] F. Pérez-Arribas and I. Trejo-Vargas, “Computer-aided design of horizontal axis turbine blades,” *Renew. Energy*, vol. 44, pp. 252–260, Aug. 2012.
- [51] C.-H. Cheng, Y.-X. Huang, S.-C. King, C.-I. Lee, and C.-H. Leu, “CFD (computational fluid dynamics)-based optimal design of a micro-reformer by integrating computational a fluid dynamics code using a simplified conjugate-gradient method,” *Energy*, vol. 70, pp. 355–365, Jun. 2014.
- [52] R. P. F. Gomes, J. C. C. Henriques, L. M. C. Gato, and A. F. O. Falcão, “Multi-point aerodynamic optimization of the rotor blade sections of an axial-flow impulse air turbine for wave energy conversion,” *Energy*, vol. 45, no. 1, pp. 570–580, Sep. 2012.
- [53] C. J. Simao Ferreira, “The near wake of the VAWT: 2d and 3d views of the VAWT aerodynamics,” Delft University of Technology, 2009.
- [54] A. Tadamasa and M. Zangeneh, “Numerical prediction of wind turbine noise,” *Renew. Energy*, vol. 36, no. 7, pp. 1902–1912, 2011.
- [55] R. Ramsay, J. Janiszewska, and G. Gregorek, “Wind Tunnel Testing of Three S809 Aileron Configurations for Use on Horizontal Axis Wind Turbines,” *Natl. Renew. Energy Lab. Golden, CO, Airfoil Perform. Rep. July*, 1996.
- [56] F. Bertagnolio, N. N. Sørensen, and J. Johansen, “Profile catalogue for airfoil sections based on 3D computatrion,” Roskilde, Denmark, 2006.
- [57] C. Kang, H. Liu, and X. Yang, “Review of fluid dynamics aspects of Savonius-rotor-based vertical-axis wind rotors,” *Renew. Sustain. Energy Rev.*, vol. 33, pp. 499–508, May 2014.
- [58] A. Bianchini, G. Ferrara, and L. Ferrari, “Design guidelines for H-Darrieus wind turbines: Optimization of the annual energy yield,” *Energy Convers. Manag.*, vol. 89, pp. 690–707, Jan. 2015.
- [59] V. Weiss, L. Andor, G. Renner, and T. Várady, “Advanced surface fitting techniques,” *Comput. Aided Geom. Des.*, vol. 19, no. 1, pp. 19–42, Jan. 2002.
- [60] H. Pottmann, S. Leopoldseder, and M. Hofer, “Approximation with active B-spline curves and surfaces,” in *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pp. 8–25.
- [61] H. Pottmann and S. Leopoldseder, “A concept for parametric surface fitting which avoids the parametrization problem,” *Comput. Aided Geom. Des.*, vol. 20, no. 6, pp. 343–362, Sep. 2003.
- [62] M. S. Floater, “Parametrization and smooth approximation of surface triangulations,” *Comput. Aided Geom. Des.*, vol. 14, no. 3, pp. 231–250, Apr. 1997.
- [63] M. S. Floater and M. Reimers, “Meshless parameterization and surface reconstruction,” *Comput. Aided Geom. Des.*, vol. 18, no. 2, pp. 77–92, Mar. 2001.
- [64] M. Ćurković, “Parameterization of 3D Objects for Numerical Analysis and Optimization of Shape and Topology,” University of Split, 2014.
- [65] M. Ćurković and D. Vučina, “3D shape acquisition and integral compact representation using optical scanning and enhanced shape parameterization,” *Adv. Eng. Informatics*, vol.

28, no. 2, pp. 111–126, 2014.

- [66] A. Olivieri, F. Pistani, A. Avanzini, F. Stern, and R. Penna, “Towing Tank Experiments of Resistance , Sinkage and Trim , Boundary Layer , Wake , and Free Surface Flow Around a Naval Combatant Insean 2340 Model,” Iowa, 2001.
- [67] G. Becker *et al.*, “An advanced NURBS fitting procedure for post-processing of grid-based shape optimizations,” *49th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo.*, no. January, pp. 1–19, Jan. 2011.
- [68] G. Greiner and K. Hormann, “Interpolating and Approximating Scattered 3D-data with Hierarchical Tensor Product B-Splines,” in *Surface Fitting and Multiresolution Methods*, 1997, pp. 163–172.
- [69] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer New York, 2006.
- [70] W. Zheng, P. Bo, Y. Liu, and W. Wang, “Fast B-spline curve fitting by L-BFGS,” *Comput. Aided Geom. Des.*, vol. 29, no. 7, pp. 448–462, Oct. 2012.
- [71] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, “Efficient topology optimization in MATLAB using 88 lines of code,” *Struct. Multidiscip. Optim.*, vol. 43, no. 1, pp. 1–16, Jan. 2011.

## ŽIVOTOPIS

Ivo Marinić-Kragić, rođen je u Splitu 02.11.1987 godine. Nakon pohađanja osnovne škole Marjan i srednje Industrijske škole u Splitu, smjer strojarstvo - obrađivanje odvajanjem čestica, započeo je studiranje na Fakultetu elektrotehnike strojarstva i brodogradnje u Splitu. Na stručnom studiju brodogradnje, završio je s odličnim uspjehom i dobio nagradu za najboljeg studenta stručnog studija brodogradnje. Nakon toga, Marinić-Kragić upisuje razlikovni studij strojarstva što je uvjet za nastavak diplomskog studija strojarstva. Na diplomskom studiju strojarstva, Marinić-Kragić je uključen u mnoge studentske projekte a najznačajniji je međunarodni projekt Formula Student koji je rezultirao s izradom prvog trkaćeg vozila tog tipa na Sveučilištu u Splitu. U sklopu projekta, Marinić-Kragić je sudjelovao na dva međunarodna natjecanja u Italiji (2013) i Češkoj (2014). Diplomski studij strojarstva Marinić-Kragić završava s odličnim uspjehom i dobiva nagradu za najboljeg studenta diplomskog studija strojarstva. Već tokom diplomskoj studija Marinić-Kragić započinje s znanstvenim radom, a iz istraživanja uključenog u diplomski rad su ukupno proizašla 3 znanstvena članka CC kategorije koja su vezana i za doktorski rad. Nakon završenog diplomskog studija, nastavio je znanstveni rad na poslijediplomskom doktorskom studiju strojarstva kojega u početku pohađa o vlastitom trošku. Uz doktorski studij strojarstva, Marinić-Kragić paralelno radi u poduzeću Stipsa d.o.o kao stručni suradnik za strojarstvo i u poduzeću Projekt Mensio d.o.o. kao razvojni inženjer na projektu razvoja bespilotne letjelice. Uz to, Marinić-Kragić obavlja nastavni rad iz kolegija analiza primjenom računala, programiranje i metodičko optimiranje kao vanjski suradnik za izvođenje nastave na Fakultetu elektrotehnike strojarstva i brodogradnje. U ožujku 2015, Marinić-Kragić se zapošljava na Fakultetu elektrotehnike strojarstva i brodogradnje kao asistent s punim radnim vremenom i prestaje s redovnim radom u navedenim poduzećima. Međutim, još uvijek Marinić-Kragić nastavlja rad na započet u poduzeću Stipsa d.o.o., a taj rad je rezultirao s dva računalna programa vezana za energetska certificiranja zgrada. Za vrijeme dokorskog studija Marinić-Kragić sudjeluje i u projektu MotoStudent, koji je rezultirao izradom prvog trkaćeg motorkotača ove vrste u Hrvatskoj. Projekt MotoStudent-2016 je završio natjecanjem u Španjolskoj u rujnu 2016. Marinić-Kragić planira doktorski rad obraniti po „Skandinavskom modelu“ što će biti prvi rad napravljen po ovom modelu na Fakultetu elektrotehnike strojarstva i brodogradnje u Splitu.

## **BIOGRAPHY**

Ivo Marinić-Kragić was born in Split, Croatia on 02.11.1987. After finishing the elementary school “Marjan” and mechanical engineering at high-school “Industrijska škola Split”, he started his undergraduate study at University of Split, Faculty of electrical engineering, mechanical engineering and naval architecture (FESB). At the naval architecture undergraduate study, he graduated with honors as the best student. Afterwards, Marinić-Kragić undertakes mechanical engineering differentiating program which is required for attending mechanical engineering graduate study. During the graduate study, Marinić-Kragić participated in many student projects of which the most noticeable is the international Formula Student project which resulted in a first race car of this type at the University of Split. As a part of the project, Marinić-Kragić has participated at two international competitions in Italy (2013) and Czech Republic (2014). Marinić-Kragić graduated the mechanical engineering graduate study with honors as the best student. Already during the graduate study, Marinić-Kragić has started with his scientific research which resulted in 3 scientific papers (in cc database journals) which were the basis for the approaching doctoral study. After the graduate study, he continued his research work on mechanical engineering postgraduate study also at University of Split. At first, Marinić-Kragić attends the postgraduate study at his own expense with parallel part-time jobs at Stipsa d.o.o. (associate mechanical engineering) and Projekt Mensio d.o.o (development of an unmanned aerial vehicle). He is also worked as an external associate at FESB where he teaches computer aided analysis, programming and optimization methods. In march 2015, Marinić-Kragić starts full-time job as research and teaching assistant at FESB but still continues working at Stipsa d.o.o on computer software related to building energy performance certificates. During the his doctoral studies, Marinić-Kragić participates in international student project MotoStudent (held in Spain, October 2016.) which resulted in first racing motorcycle of this type in Croatia. Now, Marinić-Kragić has completed all exams of the postgraduate doctoral studies and he is planning to complete his doctoral thesis in so called “Scandinavian model” which will be the first doctoral thesis written according to this model at FESB.