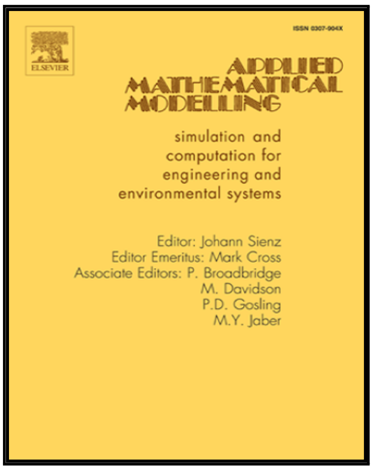
Paper:

Naumann, D., Evans, B., Walton, S. & Hassan, O. (2017). Discrete Boundary Smoothing using Control Node Parameterisation for Aerodynamic Shape Optimisation. *Applied Mathematical Modelling*
http://dx.doi.org/10.1016/j.apm.2017.03.042



ISSN 0307-904X

APPLIED MATHEMATICAL MODELLING

simulation and
computation for
engineering and
environmental systems

Editor: Johann Sienz
Editor Emeritus: Mark Cross
Associate Editors: P. Broadbridge
M. Davidson
P.D. Gosling
M.Y. Jaber

Discrete Boundary Smoothing using Control Node Parameterisation for Aerodynamic Shape Optimisation

D.S. Naumann, B. Evans, S. Walton, O. Hassan

PII:	S0307-904X(17)30191-9
DOI:	10.1016/j.apm.2017.03.042
Reference:	APM 11683
To appear in:	*Applied Mathematical Modelling*
Received date:	28 October 2016
Revised date:	10 March 2017
Accepted date:	20 March 2017

Please cite this article as: D.S. Naumann, B. Evans, S. Walton, O. Hassan, Discrete Boundary Smoothing using Control Node Parameterisation for Aerodynamic Shape Optimisation, *Applied Mathematical Modelling* (2017), doi: 10.1016/j.apm.2017.03.042

Highlights

- a novel method of shape deformation using discrete boundary smoothing
- geometry and domain coupled via a control nodes approach
- volume mesh is deformed via a Delaunay Graph Mapping technique
- Modified Cuckoo Search - an evolutionary optimisation algorithm - is employed
- Successfully applied to a range of geometries and objective functions

Discrete Boundary Smoothing using Control Node Parameterisation for Aerodynamic Shape Optimisation[☆]

D. S. Naumann[a,*], B. Evans[a], S. Walton[b], O. Hassan[a]

[a] *Zienkiewicz Centre for Civil & Computaional Engineering, School of Engineering, Swansea University, Swansea SA1 8EN, Wales, UK*

[b] *Department of Computer Science, College of Science, Swansea University, Swansea SA2 8PP, Wales, UK*

Abstract

This paper presents an automated aerodynamic optimisation algorithm using a novel method of parameterising the search domain and geometry by employing user–defined control nodes. The displacement of the control nodes is coupled to the shape boundary movement via a 'discrete boundary smoothing'. This is initiated by a linear deformation followed by a discrete smoothing step to act on the boundary during the mesh movement based on the change in its second derivative. Implementing the discrete boundary smoothing allows both linear and non-linear shape deformation along the same boundary dependent on the preference of the user. The domain mesh movement is coupled to the shape boundary movement via a Delaunay graph mapping. An optimisation algorithm called Modified Cuckoo Search (MCS) is used acting within the prescribed design space defined by the allowed range of control node displacement. In order to obtain the aerodynamic design fitness a finite volume compressible Navier-Stokes solver is utilized. The resulting coupled algorithm is applied to a range of case studies in two dimensional space including the optimisation of a RAE2822 aerofoil and the optimisation of an intake duct under subsonic, transonic and supersonic flow conditions. The discrete mesh–based optimisation approach outlined is shown to be effective in terms of its generalised applicability, intuitiveness and design space definition.

Keywords: mesh movement, Cuckoo search, computational fluid dynamics, aerodynamic shape optimization, shape parameterisation

1. Introduction

In recent years, Computational Fluid Dynamics (CFD) has become a very mature field now being the primary tool for aerodynamic design [1]. As such computational aerodynamic shape optimisation has emerged aiming to replace the resource intensive manual optimisation process based on human expertise and intuition. At the same time Computer Aided Design (CAD) has grown rapidly to become an integral part of the aerodynamic design cycle [2] which can be observed in figure 1.

Recent innovations try to couple CAD systems, CFD tools and the mesh generation process. Examples include Isogeometric Analysis [3, 4, 5] and NURBS Enhanced Finite Elements Methods [6, 7]. Despite all these

[☆] The intellectual property rights in the research data are asserted by the authors and their research partners.

[*] Corresponding Author: Tel.: +447805243278

Email addresses: 717761@swansea.ac.uk (D. S. Naumann), b.j.evans@swansea.ac.uk (B. Evans), s.p.walton@swansea.ac.uk (S. Walton), o.hassan@swansea.ac.uk (O. Hassan)

[1] Abbreviations: FDGD - Fast Dynamic Grid Deformation, DG - Delaunay Grid, CS - Cuckoo Search, MCS - Modified Cuckoo Search

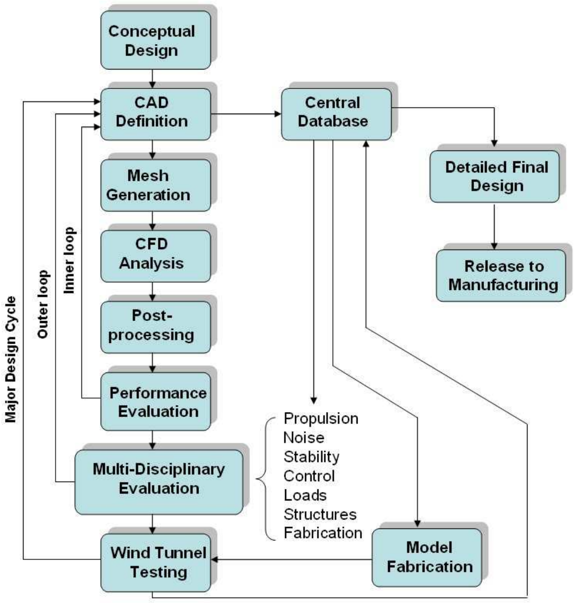


Figure 1: A typical multi–disciplinary aerospace design cycle as prevalent in industry [1]. Particularly the inner design cycle is still to date a resource intensive manual optimisation process. The aim is to automate the entire inner design cycle.

advancements significant challenges remain particularly in the parameterisation approach [8], in the efficient transfer between CAD and CFD systems and the improvement of the computationally expensive mesh regeneration process[1, 9].

Selecting the appropriate parameterisation strategy is an essential part of the aerodynamic shape optimisation process. Hicks and Henne [10] were first to explore parameterisation techniques particularly for aerodynamic shape optimisation. A large number of different parameterisation techniques have been propoosed since then.

According to Samareh [8], implementations of parameterisation techniques in the literature can be separated into eight categories: analytical, basis vector [11], free-form deformation (FFD) [12, 13], domain element [14], discrete, partial differential equation (PDE), polynomial/spline and CAD-based representations [15, 16]. Other methods include the control grid approach [17] and some parameterisation approaches are specific to a geometry type such as the PARSEC method [18] for aerofoils. All of the methods mentioned above can be broken down into two different classes [19]: mesh–based and mesh–free methods. Mesh–based methods include basis vector, domain element, discrete, analytical and free-form deformation (FFD). Those are independent of any grid generation tool. Polynomial/spline, PDE and CAD-based approaches on the other hand are usually linked to a mesh generation tool hence requiring an automated mesh generation approach and it is not obvious on how to break that link.

Kulfan [20] and Samareh [8] have identified the following criteria for parameterisation approaches: local and global control, robustness in terms of shape preservation, compactness, consistent to allow a wide range of applications, intuitive in defining the design space and understanding the impact of parameters, directly linked to mesh, allow large deformations and maintain smoothness. The last point has been redefined to smoothness control, since the authors believe that it is rather desired to have control over the smoothness, not to inevitably obtain a smooth geometry.

A popular analytical approach introduced recently is Class-Shape-Transformation (CST) [21]. The method proved very effective in terms of the number of design variables required even for three–dimensional problems [22]. In a survey by Mousah it has been shown to be less effective when compared to mesh–based and spline–based approaches [23]. FFD is a method adopted from the software animation industry. It does not act directly on the geometry, instead a hull is covering the geometry and topology changes on the hull are mapped onto the geometry. The topology of the deformation is highly dependent on the hull shape applied making it unsuitable for general applications [24, 25]. Also it is generally limited to small deformations and local control is difficult to achieve [26]. The domain element method is similar to FFD approaches [19]. PDEs allow for a compact set of parameters, however it is rather computationally expensive [27]. Basis vectors employ a set of orthogonal basis vectors as a mean to construct new shapes. With this methods only small changes can be obtained before errors in the reconstruction occur [28].

More recently parameterisation schemes are dominated by CAD based methods [15, 16, 29, 30]. This allows the designer to exploit the existing shape parameterisation in which the control points of the NURBS or spline based representation are employed as the design parameters. However, the purpose of control points is to accurately represent the shape of the geometry. This can prove impractical when used at the same time as the design parameters for the optimisation problem, since the number, location and distribution of control points may not be best suited for the purpose of optimisation limiting the explorable design space (i.e. range of potential shapes) [30]. It has also been reported that a CAD based approach can produce undesirable wiggles along the optimised shape [22]. Additionally, it is not possible to have a different order along a NURBS curve which poses problems when wanting to control the smoothness and allow for linear and non-linear features developing along the same curve.

Our approach

Aerodynamic designers have a clear preference towards tools that are intuitive, efficient and allow a wide–ranging applicability. The optimisation and design approach demands an effective geometry parameterisation to allow sufficient exploration of a design space. At the same time, the number of parameters defining a design space should be minimal to reduce the dimensionality of the problem. This paper is a continuation of the work in [31] in which the concept of control nodes in the mesh as a scheme for both defining the geometry movements and as the design parameters for the optimisation process was presented. It can be classified as a mesh-based discrete parameterisation technique. Control nodes are user–defined exploiting the expertise and intuition of the designer. In contrast to CAD based approaches the control nodes may be arbitrarily positioned without affecting the geometry. Also, the control node approach is generally applicable to any mesh–based analysis problem. Control nodes are coupled to the rest of the discrete shape boundary via a discrete boundary smoothing. This is the novel contribution presented in this paper. The proposed method allows control over the smoothness with the ability to generate linear as well as non-linear shape deformation along the same boundary in a simple, intuitive fashion. Such a feature is not obvious to achieve with currently dominating approaches in the field of shape parameterisation (e.g. FFD, CAD-based). It preserves or generates edges along the geometry whilst maintaining a smooth geometry for the remaining part.

In this work a mesh deformation scheme needs to be implemented to move the domain mesh. Boer et al. separate mesh deformation methods into point-by-point schemes and schemes that exploit the mesh connectivity

[32]. Point-by-point schemes are preferred since they are in principal applicable to any mesh type which is favourable for a generally applicable optimisation framework. Amongst them the most popular approaches are Fast Dynamic Grid Deformation (FDGD) [33] and Radial Basis Function (RBF) interpolation [34]. The latter approach has to solve a system of equations of the size of the number of boundary nodes, hence increasing the computational expense [35]. Consequently the authors implemented the FDGD approach to move the domain mesh.

Implementing the FDGD approach results in a self-contained algorithm formulated to propagate the effect of the control node displacement throughout the discrete shape boundary and computational mesh. There is no need to re–mesh in between the different stages of the optimisation. Since all knowledge of the geometry is 'stored' in the discrete boundary, there is no requirement to convert the geometry definition stored in the mesh into any other format during the optimisation process. This reduces the problem of translation of CAD-based geometry definitions to CFD meshes.

2. Methodology

2.1. Geometry Shape Parameterisation

It has been pointed out in the introduction that the transfer of geometries from CAD systems into computational meshes for simulation is critical. This problem can be circumvented by a discrete shape parameterisation method acting directly along the computational mesh. In contrast to CAD based parameterisation approaches control nodes are selected for optimisation purposes by the designer at what he deems critical locations on the discrete shape boundary. Note, that the driving factor during the optimisation is the change in the control node location rather than the control nodes themselves. The number and position of these control nodes is crucial in the evolution of the geometry. Figure 2 visualises the terms 'control nodes', 'boundary nodes' and 'domain nodes' as defined in this paper. An important feature of the parameterisation is the dimensionality of the explorable design space which can be adjusted through the number of control nodes and the designers settings. It is possible to adjust the degrees of freedom for each control node separately as well defining their range of motion. Control nodes may also be linked together to have identical displacements. This can be beneficial in various instances, e.g. maintaining the thickness of a geometry as shown in section 3.

The total degrees of freedom d within the system - a measure of the dimensionality of the design space - is defined as

$$d = \sum_{k=1}^{N_{CN}} f_{cn} \tag{1}$$

where f_{cn} is the number of degrees of freedom per control node.

More details on the control node approach is provided in [31].

2.2. Mesh Movement

In a previous paper of the authors, a new mesh movement methodology has been proposed and successfully tested to propagate the deformation of the control nodes throughout the boundary nodes and finally throughout the domain nodes [31]. A methodology involving three steps has been defined starting with the displacement of the control nodes based on the fitness that has been evaluated by the optimization algorithm. Consequently, the control node displacement triggers a deformation of the discrete shape boundary. In the end the domain nodes

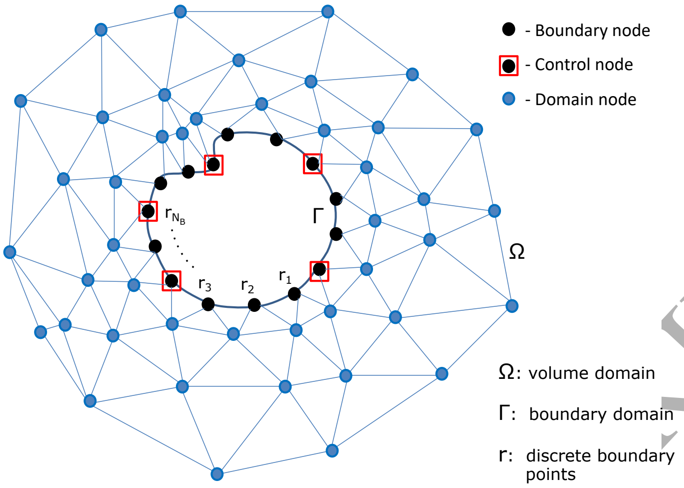
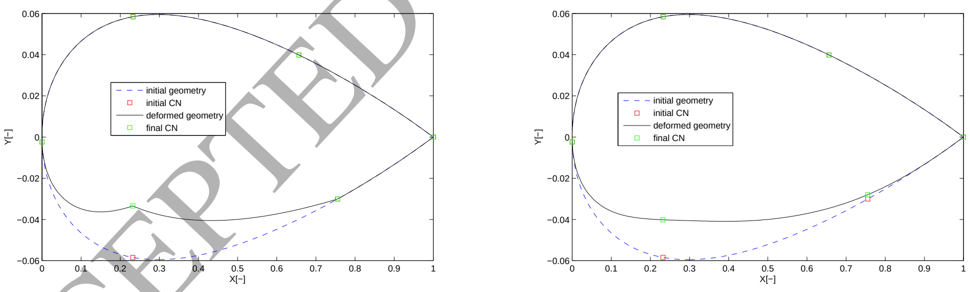


Figure 2: Definitions of control nodes, boundary nodes and domain nodes

are moved utilizing the Fast Dynamic Grid Deformation (FDGD) [33] for mesh deformation. This methodology needs an addition as it previously led to aerodynamically unfavourable geometric features along the boundary as illustrated in Figure 3 and in the results section.

As a result, a new scheme has been developed to propagate the deformation of control nodes throughout the boundary nodes. The new scheme is termed 'discrete boundary smoothing' in this paper.



(a)

(b)

Figure 3: Illustration of issues with linear deformation in part (a). Part (b) includes discrete boundary smoothing and results in an aerodynamically more desirable shape.

2.2.1. Discrete Boundary Smoothing

Inspiration

At first, a range of criteria were identified for the development of the discrete boundary smoothing. As mentioned by Samareh et al. [8], local control and shape preservation are important for applications such as optimum mounting of an engine below a wing [36]. Furthermore, for aerodynamic problems, it is in most cases advantageous to have a smooth shape that avoids sharp geometric features along the boundary to maintain attached flow, avoid turbulent flow and flow separation, all of which lead to increased drag and a loss of lift. There are applications however, that require sharp features along the boundary e.g. at the trailing edge of a wing. Those features need to be preserved or generated during shape optimization. The criteria can be summarized as follows:

- Allow linear and non-linear shape deformation along the same boundary (smoothness control)

  In most cases, a smooth boundary is desired using non-linear shape deformation. However, there are moments where sharp geometric features are necessary for example at the trailing edge of a wing. This is only achieved using a linear shape transformation technique.

- Allow local control

  The optimization of only a part of the geometry e.g. the leading edge of a wing should be possible.

- Allow shape preservation

  During optimization, there may be features that the designer wishes to preserve whilst optimising other parts. This is closely linked to allowing local shape deformation.

- Allow coupling with control nodes displacement and the discrete boundary representation

  Ultimately, the boundary shape deformation including the discrete boundary smoothing needs to be controlled by the control nodes and their displacement.

The existing linear technique [31] addresses the final three criteria. Using a linear deformation alone, however, inevitably induces sharp geometric features during boundary mesh deformation. Figure 3 and figure 19 (a) provide two such examples.

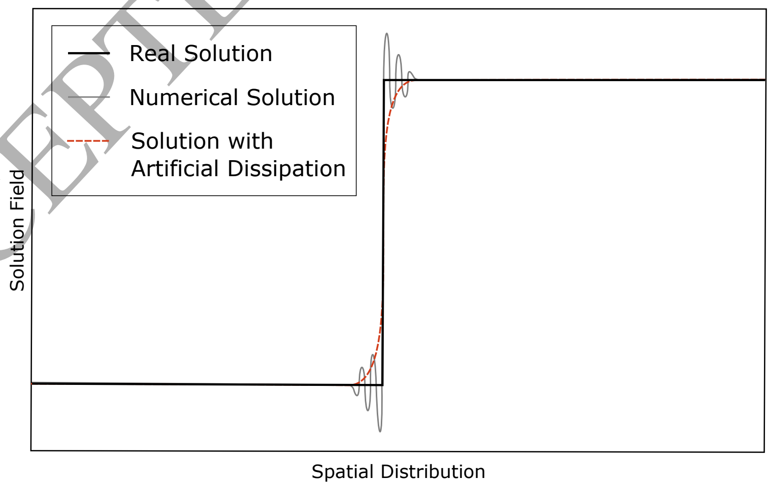


Figure 4: Effect of Artificial Dissipation

In conclusion, a flexible higher order method is necessary to allow smooth boundary shapes whilst also maintaining linear characteristics in local regions of the boundary. A mathematical solution already existing in the field of computational fluid dynamics to tackle a similar problem is called artificial dissipation, a numerical method used to damp oscillations inherent in the solution of the unknowns [37].

By adding the artificial dissipation term, regions of high gradients are getting 'smeared' across the solution field to achieve numerical stability. This is represented in figure 4 in which a sudden step in the solution as for instance common across shock waves is being smoothed by applying artificial dissipation. Artificial Dissipation is comprised of two components, a 3rd order dissipation term to ensure robustness of a numerical scheme and a 1st order term to preserve monotonicity and smooth oscillations near strong gradients/discontinuities. Particularly the effect of smoothing is a similar characteristic desired for the boundary shape deformation. On one hand, sharp geometric features as a result of the boundary deformation shall be avoided to maintain aerodynamically smooth surfaces. On the other hand, a scaling factor is required to control the level of smoothing, even to the point of no smoothing to allow for sharp features when desired.

Numerical Description

The discrete boundary smoothing takes upon the idea of a damping term similar to artificial dissipation. Let $r \subset \Gamma$ be the discrete boundary nodes ($r = (x, y)$) of a geometry with boundary Γ, then the discrete smoothing term for a number of smoothing iterations $i = 2, 3, ..., N_s$ is set to

$$r_j^{i+1} = r_j^i + \beta\lambda_j \left(\|\nabla^2 r_j^i\| - \|\nabla^2 r_j^{i=0}\| \right) \tag{2}$$

with $j = 1, 2, ..., N_B$. N_B is the number of boundary nodes and N_s is the number of smoothing iterations. $\| \quad \|$ is defining the norm operator in this paper. β is both a scaling factor and stability factor and λ is the smoothing factor. Their purpose will be demonstrated later in this chapter. The smoothing iterations propagate the deformation of a boundary node and its respective change in $\nabla^2 r$ throughout the boundary. This principle is illustrated in figure 5. Note, that $i = 1$ is excluded from the iteration scheme since the first step is necessary to perform an initial linear deformation. Such linear deformation may be achieved by simple interpolation in between each control node. In order to calculate $\nabla^2 r$ a central difference scheme for non-uniform mesh density distribution has been employed in two dimensions.

Throughout the paper the index $k = 1, 2, ..., N_{CN}$ defines the control nodes r_k and r_k^∞ are the control nodes in their final position as given by the optimisation algorithm.

Within the formulation the three parameters, β, N_s and λ are of great importance. All of them fulfil a different purpose as will be outlined within the next paragraphs. For consistent results it is important that the effect of the smoothing algorithm is independent of the mesh spacing (i.e. resolution and boundary node distribution). Neither the number of boundary nodes (mesh resolution) nor the boundary node distribution should influence the smoothing behaviour. Both problems have been addressed through appropriate selection of β and N_S as will be shown.

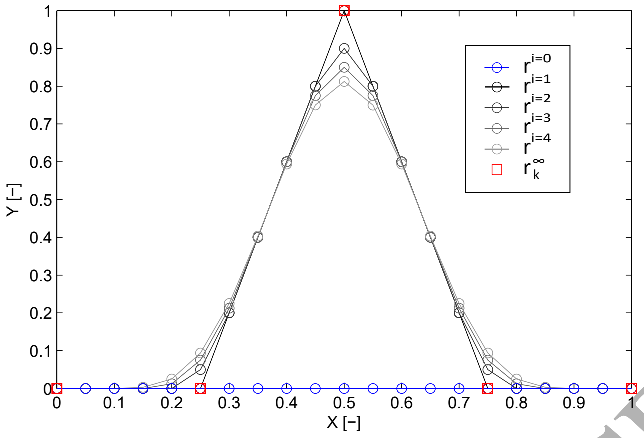


Figure 5: The first four steps of the Discrete Boundary Smoothing for an example with a flat plane as a starting point and a deformation of the central control node. β is given by equation 7 and λ was set to 0.1.

Principally any value could be prescribed for N_s and some level of smoothing would be obtained. However, the number of iterations N_s is a parameter that can balance the effects of different mesh resolutions. Empirical testing to find the number of iterations N_s providing mesh resolution independence has yielded

$$N_s = \left(\frac{N_B}{N_{CN}}\right)^2 \tag{3}$$

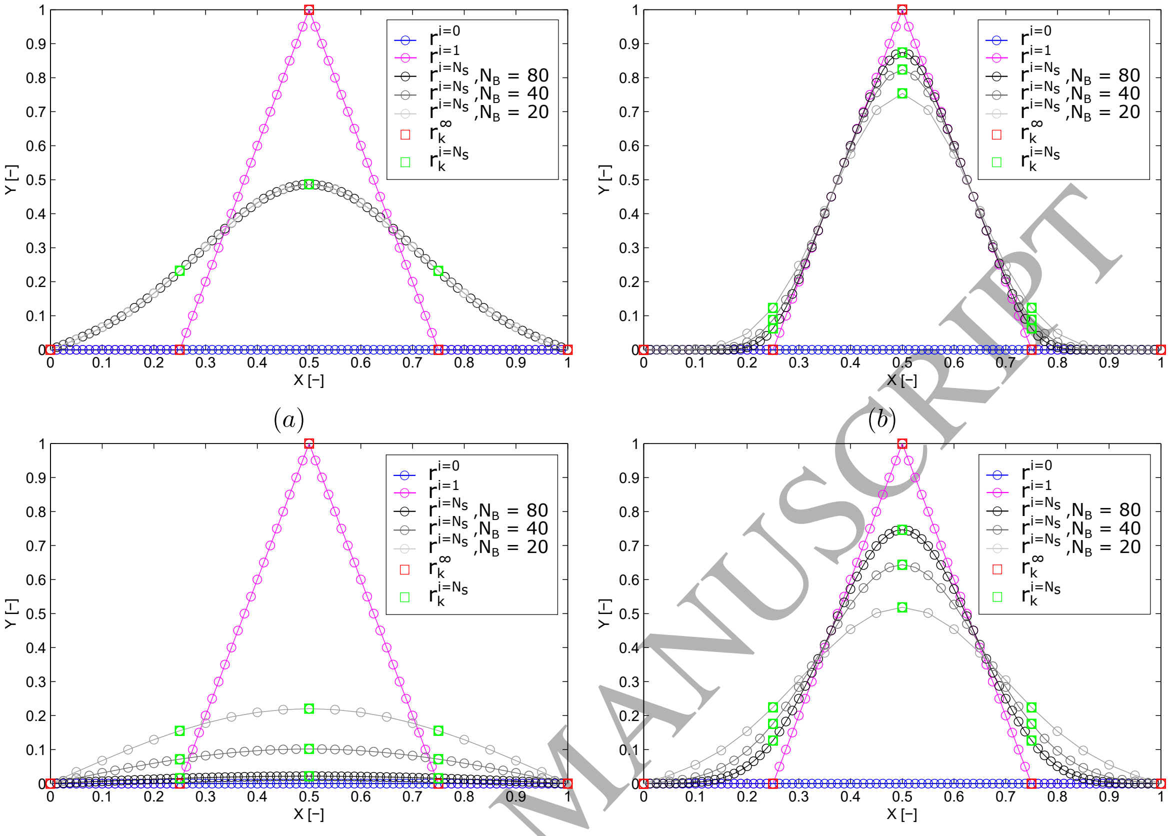
where N_B is the number of boundary nodes and N_{CN} the number of control nodes. Sample results of this study are shown in figure 6. In this figure, the behaviour of the boundary for four different numbers of smoothing iterations N_s is compared when exposed to a deformation of the central control node. Each sub-figure contains the initial boundary, in this case a flat plate, the result of the linear deformation during step $i = 1$ and the effect of the smoothing. The effect of the smoothing is shown on three different meshes that differ in the number of boundary nodes applied ($N_B = [20, 40, 80]$). Only figure 6 (a) yields the exact same deformation disregarding the number of boundary nodes N_B.

β is introduced initially as a scaling factor. In equation 2 the second derivative is directly added to the coordinate values r even though these parameters are of different magnitudes. A conversion is thus required in form of a scaling. The scaling factor β is derived from a central difference approach used to determine $\nabla^2 r$. Every $\nabla^2 r_j$ is calculated on a reference element as illustrated in figure 7. On this reference element $n_{j-1} = n_{j+1} = 0$, whereas s is the tangential coordinate along the boundary and n is the normal coordinate. Formulating the central difference scheme $\nabla^2 n_j$ may be derived as

$$\nabla^2 n_j \approx \left.\frac{d^2 n}{ds^2}\right|_j = \frac{n_{j-1} - 2n_j + n_{j+1}}{\Delta s^2} \tag{4}$$

where Δs is the distance between two consecutive boundary nodes in tangential direction. With $n_{j-1} = n_{j+1} = 0$ Equation 4 becomes

$$n_j = \frac{\Delta s^2}{2}\Big(-\nabla^2 n_j\Big) \tag{5}$$



(a) (b) (c) (d)

Figure 6: Mesh Resolution for different N_s: (a) $N_s = \left(\frac{N_B}{N_{CN}}\right)^2$ (b) $N_s = \left(\frac{N_B}{N_{CN}}\right)$ (c)$N_s = \left(\frac{N_B}{N_{CN}}\right)^3$ (d) $N_s = N_B$. Only (a) results in the same deformation even for different N_B.

This equation illustrates the relation between $\nabla^2 n$ and the coordinate n, whereas a coefficient has appeared as a result of the derivation of the form

$$\beta = \frac{\Delta s^2}{2} \tag{6}$$

Now it is obvious that β is a scaling factor that allows to directly relate $\nabla^2 r$ to r. However, Equation 4 assumes a uniform mesh density distribution in which β is the same for every boundary node. In order to account for a non-uniform mesh density distribution (a varying Δs) β is modified to

$$\beta = \frac{\Delta s_{min}^2}{2} \tag{7}$$

where Δs_{min} is the smallest tangential distance between two consecutive boundary nodes. By adding this modification β ensures stability of the scheme. In this capacity β is comparable to the CFL condition. Both identify the size of the smallest element as critical for stability.

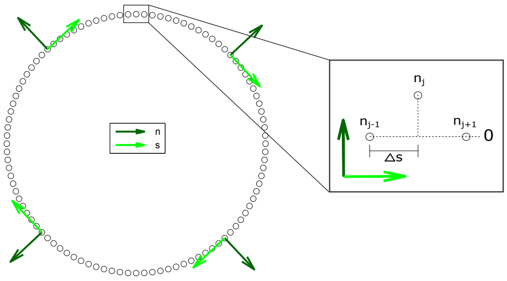


Figure 7: An example boundary given normal and tangential directions

During the conversion from equation 6 to 7 a coefficient $c = \Delta s^2/\Delta s^2_{min}$ has been neglected that should have appeared as a result of the modification. In its current form equation 2 is dependent on the size of the smallest element, thus the smoothing becomes dependent on the boundary node distribution. A uniform mesh would deliver different results when compared to a non-uniform mesh. In order to maintain boundary node distribution independence the coefficient needs to be considered and compensated in some form. Looking back at equation 2 it is apparent that there is a correlation between β and N_s. If β is reduced and hence the amount of smoothing would be reduced this may be compensated by an increase in smoothing iterations N_s. Recall figure 6 in which an increase in N_s results in more smoothing. This correlation may be formulated as

$$\beta \sim \frac{1}{N_s} \tag{8}$$

so that the coefficient c may be added to the number of smoothing iterations $N_s = cN_s$. This way, boundary node distribution independent smoothing is achieved as presented in figure 8.

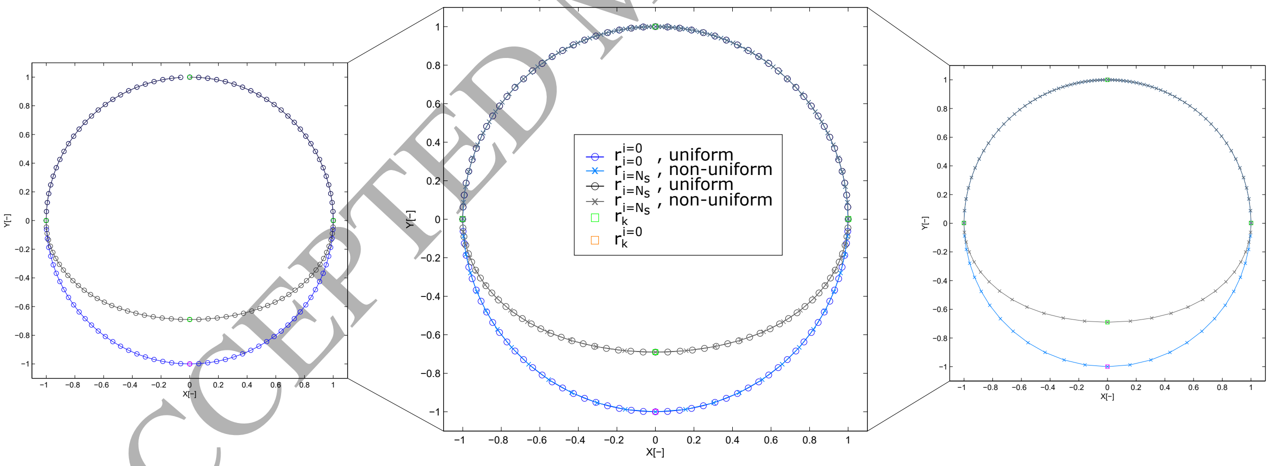


Figure 8: A comparison of the discrete boundary smoothing on the example of a circle for a uniform (left) and non-uniform (right) boundary mesh. Both deformations overlaid (centre figure) present the exact same deformation.

λ is the 'smoothing factor' selected by the designer dependent on the level of smoothing desired. It is a simple constant added to the system and as such does not effect mesh independence. For a value of $\lambda = 0$ the boundary deformation is purely linear. Any other value imposes a smoothing resulting in non-linear deformation. This parameter can be used to bridge the gap between linear and non-linear deformation. For a problem to behave linear and non-linear along the same boundary λ needs to be variably definable such that linear features can be preserved locally or different levels of smoothing are possible. Practically the user first defines the values of λ_k, which is the λ at each control node. Then all values λ_j of all boundary nodes are linearly interpolated. In two dimensions this is calculated by applying

$$\lambda_j = \lambda_k \left(1 - \left(\frac{s_j - s_k}{s_{k+1} - s_k}\right)\right) + \lambda_{k+1}\left(\frac{s_j - s_k}{s_{k+1} - s_k}\right) \tag{9}$$

whereas k and $k+1$ are the indices of the control nodes enclosing the boundary node j. In summary N_s is to prevent mesh resolution dependency of the scheme, β is introduced for stability and scaling so that $\nabla^2 r$ can be added directly to the coordinates r and λ is added artificially to provide the user with a control over the level of smoothing. β and N_s are mesh dependent parameters with β depending on the smallest tangential edge length between consecutive boundary nodes Δs_{min} and N_s depending on the number of boundary nodes N_B and number of control nodes N_{CN}. λ on the other hand is a user–defined constant.

Finally, the correct selection of λ values whilst maintaining a stable scheme is important. A value $\lambda = 1$ poses a stability limit as otherwise the stability factor β would be exceeded. This is again much similar to the CFL condition that cannot exceed 1. The ideal value for λ is subject to the required or desired level of smoothness and hence case dependent. Figure 9 illustrates the effects of different λ values. Additionally figure 10 illustrates a case in which two different λ values were applied. $\lambda = 0$ ensured that the smoothing remained local not affecting the left half of the circle. This figure also shows the benefits that linear deformations may have. At the step $l = 1, i = 1$ a trailing edge was generated resembling the shape of an aerofoil. This edge would be maintained if $\lambda = 0$ at $r_k = (1, 0)$. It is a first small step towards a more general deformation approach allowing a more flexible design space. Throughout the paper, the authors used a value of $\lambda = 0.1$ as it proved sufficient for small to medium deformations.

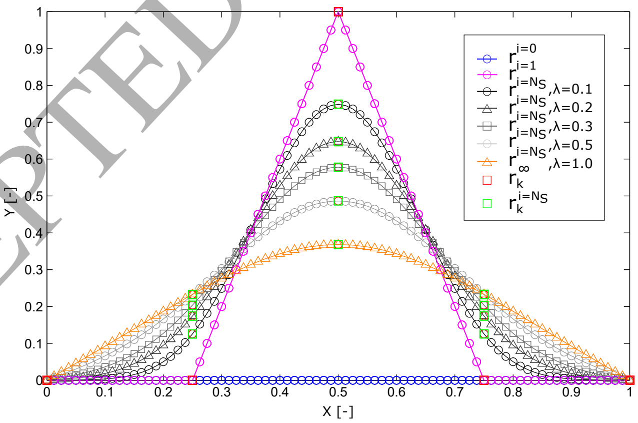


Figure 9: The effect of smoothing for different λ values. The stability limit is $\lambda = 1.0$

2.2.2. Convergence

A consequence of implementing the discrete boundary smoothing after the linear deformation is a new position of the control nodes away from the originally prescribed r_k^∞ as in figure 6. This is a conflicting characteristic to the idea of an intuitive algorithm that allows the designer direct control over his design space and the level of deformation. As such an additional iterative step has been integrated to converge the control node position towards the originally set location r_k^∞ until the convergence criteria $\epsilon_l \leq \epsilon_0$ is met. ϵ_0 is selected by the user and $l = 1, 2, ..., N_c$ is the index indicating the convergence iterations. In this article ϵ_0 has been set to a value of -3. A convergence ratio of -3 has been found sufficient, however, if the computational expense becomes prohibitive it may be reduced to -2 or less. ϵ_l is

$$\epsilon_l = \log\left(\frac{\sum_{k=1}^{N_{CN}} \|r_k^l - r_k^{l-1}\|}{\sum_{k=1}^{N_{CN}} \|r_k^{l=1} - r_k^{l=0}\|}\right) \tag{10}$$

The additional convergence iterations are placed around the boundary mesh movement step and ensure that the control nodes maintain their position r_k^∞. With every convergence iteration the discrete boundary smoothing is performed again using the already smoothed geometry of the previous iteration as the starting point. Hence, the smoothing iterations become sub-iterations that are repeated within each new convergence iteration l. The total number of steps amount to $N_s * N_c$. To guarantee the control nodes to be placed exactly at r_k^∞ smoothing is suppressed for the last iteration during convergence. Figure 10 shows the outcome with all intermediate steps until the boundary shape movement converges.

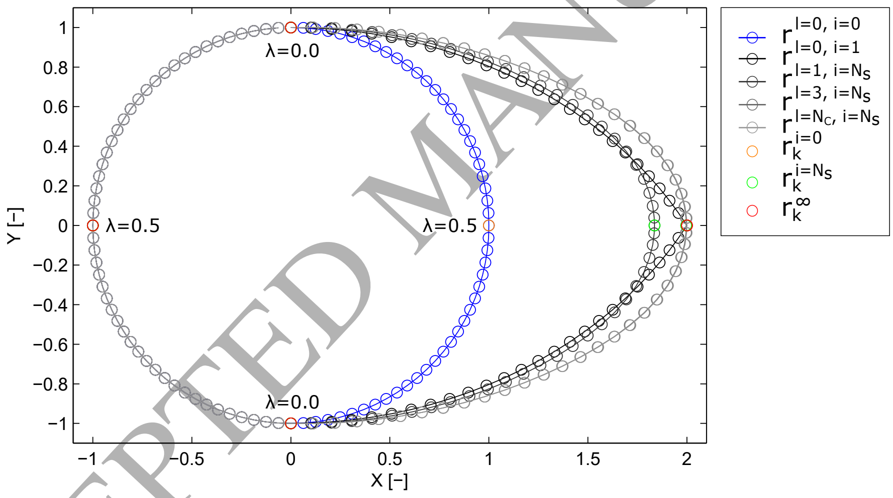


Figure 10: A final example illustrating all features of the discrete boundary smoothing. Shape preservation is guaranteed through $\lambda = 0$. The boundary is smoothed for $\lambda \neq 0$. The final control node location r_k^∞ is maintained through convergence.

2.3. Optimisation Approach

'Control nodes' used to parameterise the problem are advantageous since they act as both the geometry parameters as well as the design parameters for the optimisation process. The design space is defined by the user through the components position vectors. Traditionally gradient based optimisers [15, 38, 39] have been used in context of aerodynamic shape optimisation. The optimisation approach derived in this paper shall allow the exploration of large design spaces with multiple degrees of freedom. Gradient based optimisation approaches are local in nature and as such do not suit the search of large design spaces. This lead to the conclusion to use an evolutionary algorithm, a subclass of heuristic global optimisation algorithms. The optimisation algorithm applied in this paper is called Modified Cuckoo Search (MCS) [40]. It was developed with the goal of reducing the number of fitness evaluations required and has shown its effectiveness in various applications [41, 42]. The MCS algorithm is a population-based algorithm and applies mutations and crossovers in every generation to create a new set of agents, called Nests. It is based on the CS algorithm [43] which was used on a number of occasions [44, 45, 46, 47, 48]. A comprehensive description of the optimisation algorithm as implemented in this paper may be found in [31, 40].

2.3.1. Initial Sampling

Before the initiation of the MCS algorithm an initial population of size N_P is required as a starting point based on the initial geometry provided by the user. Latin Hypercube Sampling [49] was selected for its efficient sampling of the entire design space. The design space is determined by the allowed range of control node displacements. The dimensionality of the design space is defined by the degrees of freedom d. LHS divides every dimension into N_P equal intervals to create N_P^d cells. Afterwards a cell is selected at random. Any other cell being part of the same interval of every dimension is now excluded. This process is repeated N_P times until each interval contains exactly one selected cell. Commonly used values for N_P are related to the degrees of freedom in the system by $N_P = 10d$ [41, 50].

2.4. Computational Fluid Dynamics

All case studies conducted in this work were two–dimensional in space, although all the techniques considered (LHS [49], MCS [40], FDGD [33]) have natural three dimensional extensions. Since the algorithm utilizes the CFD solver as a black box, the CFD approach will only be described in brief.

The Swansea University FLITE CFD system was used to generate two dimensional unstructured triangular meshes [51]. The system applies the advancing layers technique for the boundary layer mesh with cell heights defined by the user. An isotropic triangular mesh is then generated using the Delaunay technique with point insertion governed by a pre-defined mesh cell size function across the computational domain.

The FLITE CFD system fluid solver is an edge–based, vertex–centred finite volume discretisation for solution of the compressible Reynolds Averaged Navier–Stokes equations [52]. The turbulence model applied for the viscous example was the one equation Spallart-Allmaras [53]. Standard subsonic and supersonic far-field boundary conditions were selected. For solid surfaces a no slip condition was imposed for viscous cases and for inviscid cases, they were treated as slip walls. In case of the engine intake examples the mass flow was fixed on the compressor intake. Its implementation approach for this is outlined in reference [54]. The results of the CFD solver forms the input for the optimisation algorithm to determine the shape ‘fitness’ value.

3. Case Studies

In the following chapter two different 2D aerodynamic optimisation problems are considered to illustrate the characteristics of the developed algorithm. The first problem contains two case studies on a widely recognised reference case from NASA on supercritical aerofoils [55], the optimisation of the transonic RAE2822 aerofoil [16, 50, 56, 57]. All other case studies are applied to subsonic, transonic and supersonic optimisation of an engine intake duct of a generic aerospace vehicle. However they differ in the optimisation set up to test its effectiveness and flexibility regarding shape optimisation. The first aerospace vehicle case study is compared to previous results [31] to demonstrate the improvement introduced by the discrete boundary smoothing. The cases have been parallelised by taking advantage of the population-based optimisation algorithm such that all agents per generation are solved simultaneously.

3.1. Problem Set Up: RAE2822 Aerofoil

The RAE2822 transonic aerofoil is a commonly used reference case for validation of CFD solvers as well as aerodynamic optimisation problems. It is a very challenging case as even small changes in the geometry can have large effects on the pressure distribution in the transonic regime.

Starting point was a NACA0012 aerofoil with the objective to match the pressure distribution of the RAE2822 aerofoil. The mesh utilized is illustrated in figure 12 and exhibits 32068 mesh nodes and 63672 mesh elements. Solutions to the problem were sought at a Mach number of $Ma = 0.729$ with an angle of attack of $\alpha = 2.31$ deg using viscous CFD simulations with a Reynolds number of $Re = 6,500,000$. To capture viscosity effects, a triangulated boundary layer mesh with 27 layers was integrated into the mesh. The Y+ value 1.0 for the first layer and mesh layers growing linearly to meet the background volume mesh size by layer 27 were chosen. No wall functions were used. The CFD results were compared to previous experimental data by AGARD (Case 6) [55] as well as simulation data by NASA [58] and excellent agreement was found as visible in figure 11.

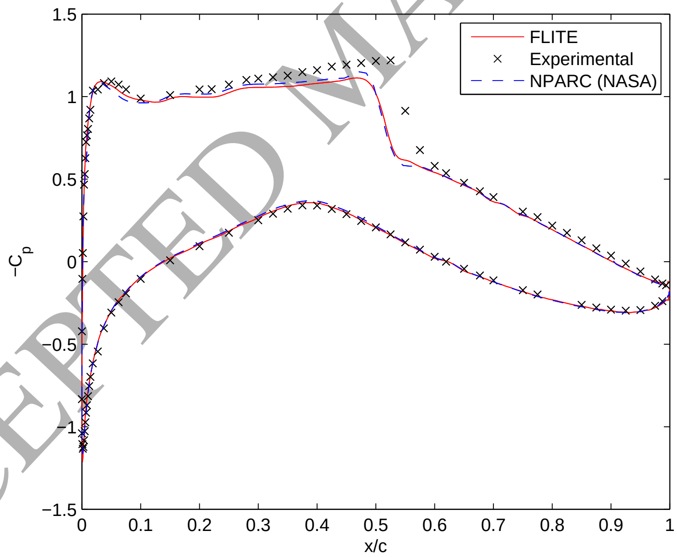
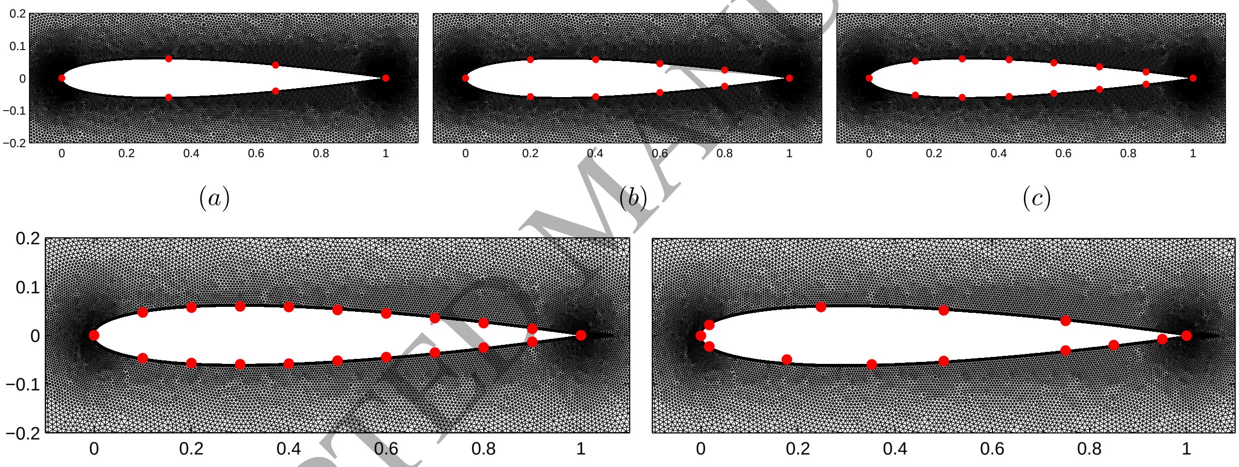


Figure 11: Validation of CFD results on RAE2822 aerofoil

For Case 1 in Section 3.2 a range of different control node numbers and locations were tested. For Case 2 13 control nodes r_k were employed. The control nodes at the leading and trailing edge of the aerofoil were fixed in position. All other control nodes exhibited an explorable design space of $x_k \in [-0.025, 0.025]$. Figure 12 (e) visualises the location of all control nodes. A lambda value of $\lambda = 0$ was defined for the trailing edge control node in order to maintain the edge shape. All other control nodes had a lambda value of $\lambda = 0.1$. Considering two control nodes to be fixed, the total number of degrees of freedom range from 4 to 18 using equation 1 resulting in up to 180 Nests according to the convention discussed in section 2.3.1. However, in order to reduce the computational cost the number of Nests was reduced to 40 for all cases considered. Table 1 summarizes all parameters and may be found in the conclusion section.

3.2. Case 1: Match Geometry

The first case is purely geometrical to demonstrate the new discrete boundary smoothing irrespective of any aerodynamic considerations. As such, the CFD solver was decoupled from the system. In order to study the effects of the number of control nodes on the system the solution was sought after at a range of $N_{CN} = [6, 10, 13, 14, 20]$. This results in $DoF = [4, 8, 11, 12, 18]$ degrees of freedom in the system considering the control nodes at the leading and trailing edge were fixed. The control nodes were distributed evenly across the boundary Γ of the aerofoil. Only in the case of 13 control nodes, the locations were selected based on the designers expertise. All control node locations are shown in figure 12.



(a) (b) (c)

(d) (e)

Figure 12: Mesh of NACA0012 aerofoil (starting geometry) with (a) 6 N_{CN}, 4 DoF (b) 10 N_{CN}, 8 DoF (c) 14 N_{CN}, 12 DoF (d) 20 N_{CN}, 18 DoF (e) 13 N_{CN}, 11 DoF

3.2.1. Fitness (objective function)

The fitness is the difference in the enclosed area between the current aerofoil and the RAE2822 aerofoil. This has been mathematically defined as follows:

$$F = -A = -\iint_{\Omega_n} \mathrm{d}A \tag{11}$$

where $\Omega_n = \Omega_{RAE} \cup \Omega_{current} \setminus \Omega_{RAE} \cap \Omega_{current}$ is the domain enclosed between the reference RAE2822 aerofoil domain Ω_{RAE} and the current aerofoil domain $\Omega_{current}$. The negative enclosed area $-A$ is taken to transform the system into a maximization problem with the theoretical maximum of zero.

3.2.2. Results

An overview of the fitness development for all number degrees of freedom is provided in figure 13. Shown is the fitness value for all degrees of freedom whereas every line is all fitness values at a given generation. On one hand, it can be observed that with increasing degrees of freedom the fitness improves considering the maximum number of generations (500). On the other hand, if the number of generations is lower fewer degrees of freedom seem advisable in order to maximise fitness meaning that the convergence rate is more rapid. Exemplary is the line for 50 generations in which 8 degrees of freedom achieve the best result.

Another important consideration is the inclusion of the designers expertise since it can improve the effectiveness of the optimisation process as it is the case with 13 user placed control nodes shown. In terms of convergence rate, it competes with 4 and 8 degrees of freedom whilst the final maximum fitness is better than the fitness of the case with 18 degrees of freedom. Here, one of the strengths of this method becomes apparent as it allows to consider the designers expertise during the set up ultimately leading to fewer degrees of freedom. Also, the ability of selecting the control node locations independent of the geometrical parameterisation allows to put emphasis on specific areas of interest.

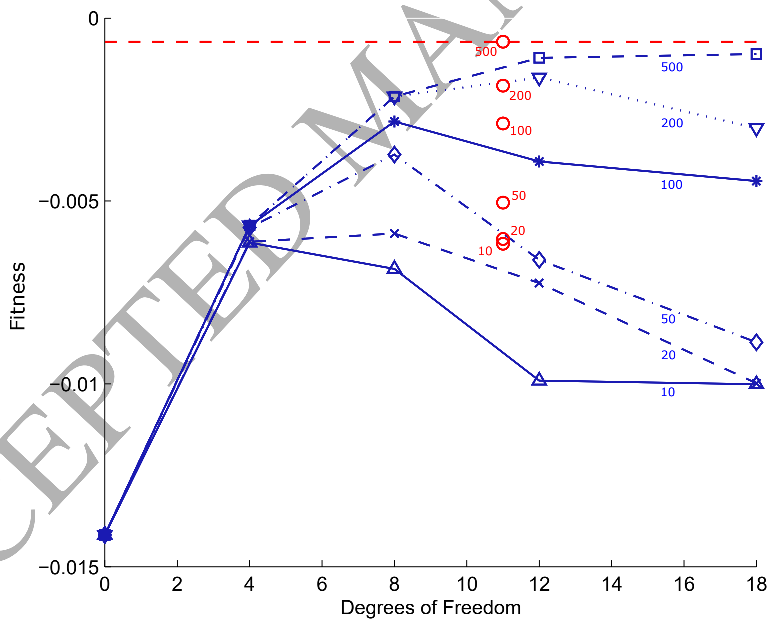


Figure 13: Validation of CFD results on RAE2822 aerofoil. Every line depicts the fitness for a given number of generations. The points in red are for the case with user placed control nodes.

After all, the final results are excellent when looking at the ultimate gains in figure 13. An improvement of up to 96 % (100 % being the maximum) has been achieved. The starting shape (NACA0012), target shape (RAE2822) and optimum result of the algorithm are depicted in figure 14. Particularly figures 14 (d) and (e) show an excellent match of the final shape. The parameterisation with individually chosen control nodes has been selected for the subsequent case aiming to match the pressure distribution.

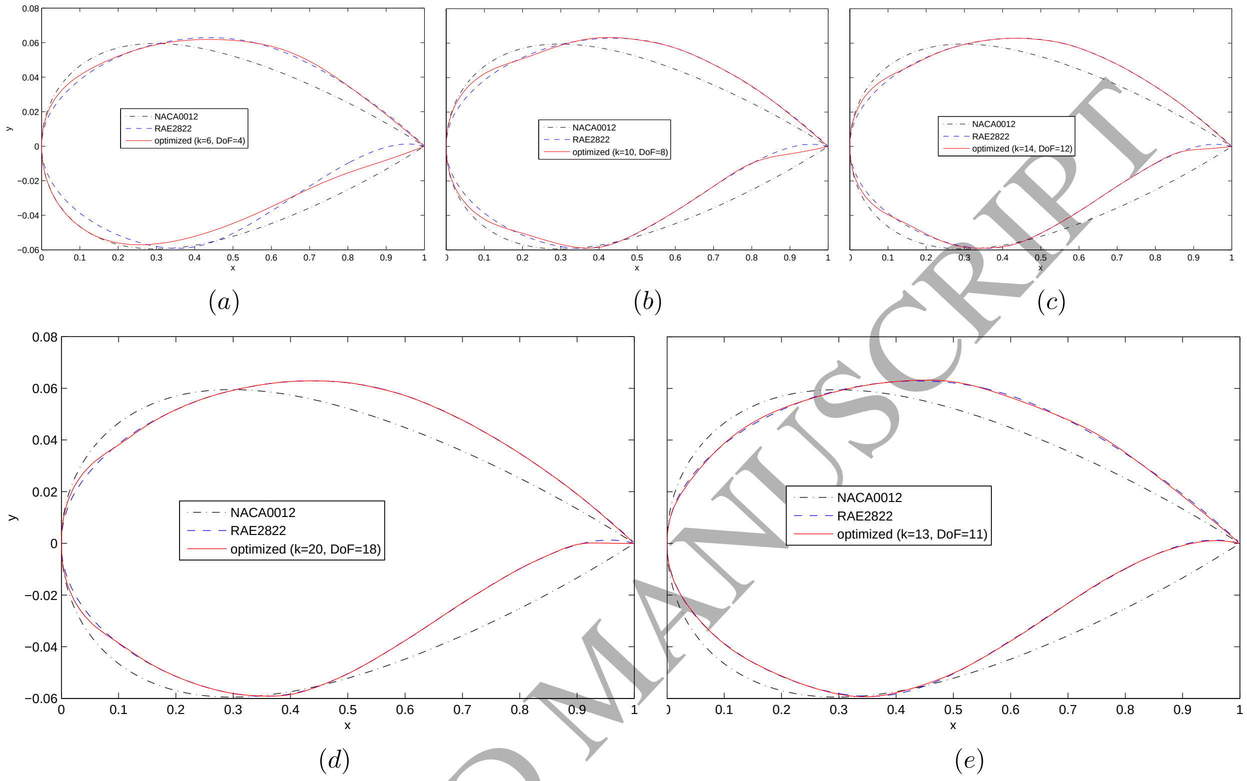


Figure 14: Optimized geometry in comparison with the starting geometry (NACA0012) and the target geometry (RAE2822) using (a) 6 N_{CN}, 4 DoF (b) 10 N_{CN}, 8 DoF (c) 14 N_{CN}, 12 DoF (d) 20 N_{CN}, 18 DoF (e) 13 N_{CN}, 11 DoF

3.3. Case 2: Match Pressure Distribution

After having demonstrated the new discrete boundary smoothing in itself it has been applied to an aerodynamic optimisation case. The aim is to match the pressure distribution of the RAE2822 aerofoil. The set up uses 13 control nodes of section 3.2 has been selected for this study.

3.3.1. Fitness (objective function)

The fitness is the difference of the pressure distribution between the RAE2822 aerofoil and the current aerofoil. In mathematical terms this is expressed as

$$F = -A_{C_p} = -\iint\limits_{\Omega_n} \mathrm{d}A_{C_p} \qquad (12)$$

where C_p is the pressure coefficient and $\Omega_n = \Omega_{C_{p,RAE}} \cup \Omega_{C_{pcurrent}} \setminus \Omega_{C_{p,RAE}} \cap \Omega_{C_{pcurrent}}$ is the domain enclosed between the reference RAE2822 aerofoil pressure distribution $\Omega_{C_{p,RAE}}$ and the current aerofoil pressure distribution $\Omega_{C_{p,current}}$. The negative integral is taken to transform the system into a maximization problem with the theoretical maximum of zero.

3.3.2. Results

The results are illustrated in two figures. Figure 16 contains the pressure distribution, its related flow field and the geometric change of the aerofoil at selected generations. The respective improvement in fitness over the generations is visualised in figure 15. It is well known that it is difficult to define a convergence/stopping criteria for an evolutionary optimisation algorithm other than identifying a tendency to stagnate. In this case study a stopping criteria was initially set to 500 generations. Due to the sudden jump in generation 486 the run has been extended to 700 generations at which point the algorithm seems to have reached near convergence.

If the theoretical optimum of $F = 0$ is considered to be 100 % the MCS algorithm has achieved a significant improvement of more than 90 % and very accurately matches the pressure distribution in figure 16 (b) as well as the shape of the RAE2822 aerofoil. The pressure fields in figure 16 (c) illustrate the challenge of this optimisation task to match the shock wave accordingly. Ultimately, it captured the shock formation of the RAE2822 aerofoil. However, a small shock wave at the leading edge occurred. This is attributed to the small geometric deviation possible in that location. Both, in Figure 14 (e) and in figure 16 (a) the shape is not entirely matched at that point and even such a slight geometric deviation can cause a shock wave given the sensitivity in the transonic regime. In order to alleviate this problem, another control node may be inserted or the smoothing parameter λ adjusted.

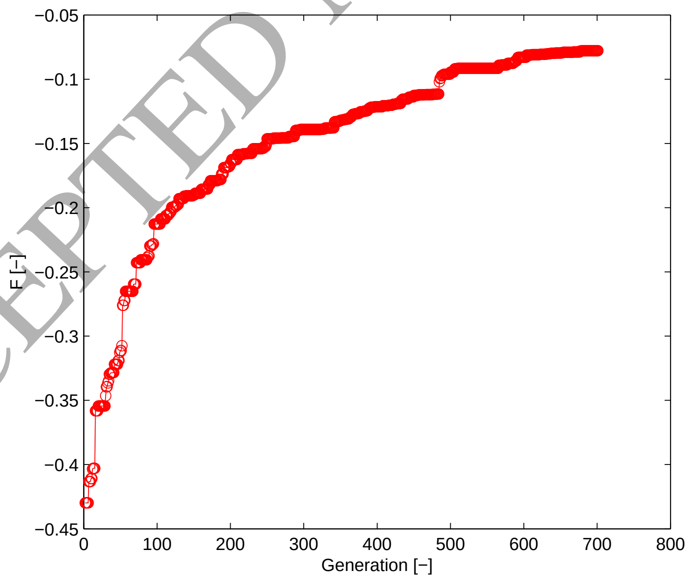
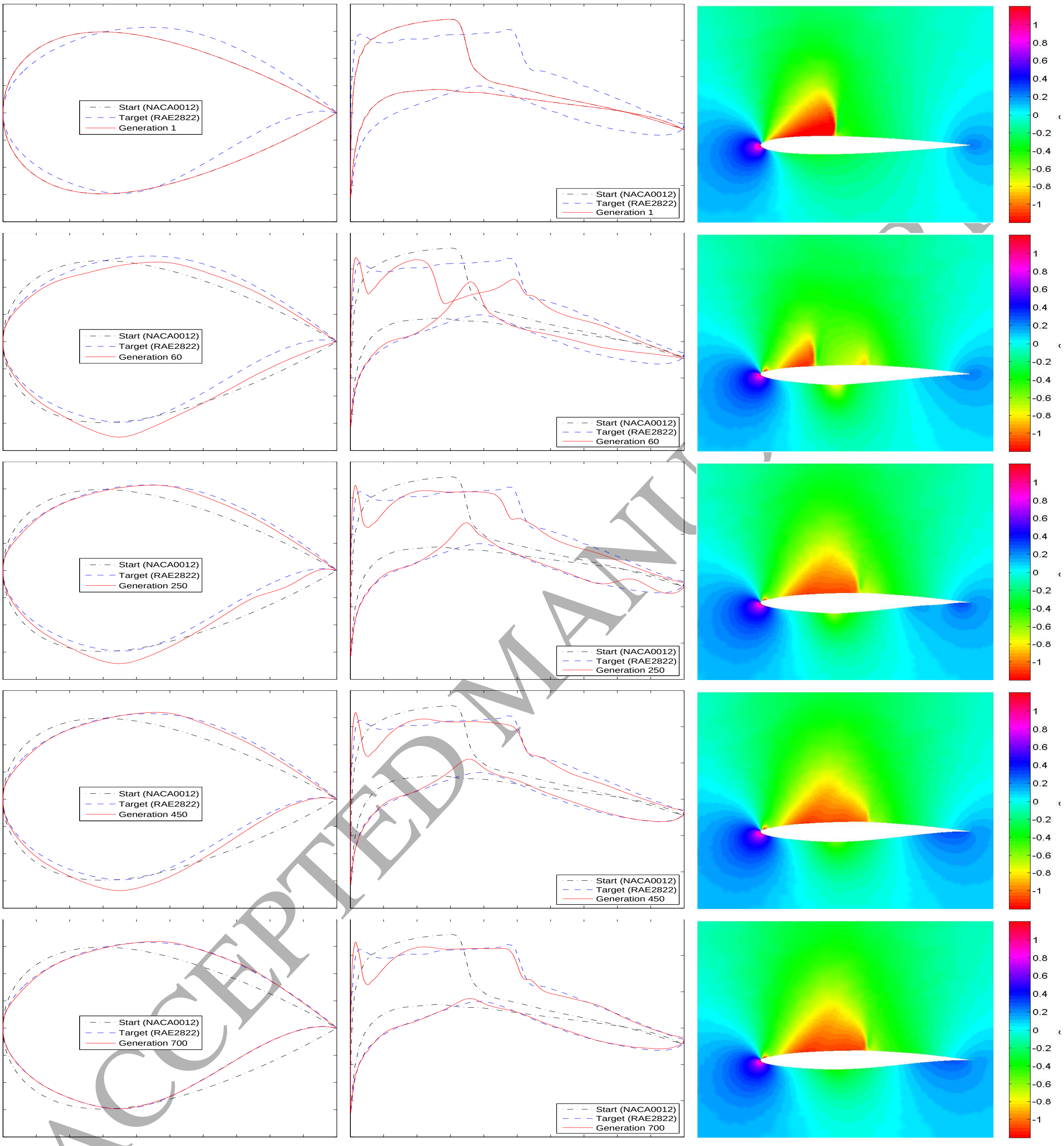


Figure 15: Fitness development over cuckoo generations



(*a*) (*b*) (*c*)

Figure 16: Snapshots of the (a) Geometry deformation (b) $-C_p$ distribution (c) C_p field at Generation 1, 60, 250, 450 and 700

3.3.3. Comparison

In order to judge the compactness and efficiency of the approach a brief comparison to more classical approaches is useful. One of the most compact approaches is the PARSEC method, a geometric approach specifically designed for optimizing aerofoils [18]. It applies a total of 11 physical parameters directly related to the geometry, e.g. aerofoil thickness. An increasingly common method of the class of analytical parameterisation is the CST method. It applies analytical shape functions to represent geometries and is known for its compactness [22]. It also requires 11 parameters to represent an aerofoil shape [21]. CAD-based approaches usually require around 32 control points to accurately represent an aerofoil and even then it is sometimes considered premature [59]. Martin et. al have used 14 control points to represent a RAE 2822 aerofoil and still obtained a significant difference in lift and drag values compared to an exact geometry representation [16]. The method presented here has applied 11 degrees of freedom obtaining very good results with the potential of obtaining near optimal results by adding another couple of control nodes (degrees of freedom). As such it can be described as a compact and efficient parameterisation approach with good control over the smoothness and local control over the design space.

3.4. Problem Set Up: Intake Duct Optimisation

In the aerodynamic design community the optimisation of a jet engine intake duct[60] tailored to a specific aircraft configuration is a common problem [61]. In order to ensure efficient performance and avoid stall or surge two conditions need to be usually satisfied. First, the flow pattern at the engine's compressor face must be satisfactory and secondly, the loss of energy in the flow must be kept minimal across the entire speed range and flight conditions that the vehicle will experience [61, 62]. In most instances, the design process is driven manually by the designers expertise and engineering intuition[51, 54]. The developed algorithm aims to automate this process.

The starting geometry was an engine intake duct of a generic aerospace vehicle. The mesh utilized is illustrated in figure 17 and exhibits 82868 mesh nodes and 163419 mesh elements. Solutions to the problem were sought at a range of Mach numbers $Ma = [0.5, 0.8, 1.1, 1.4]$ using viscous CFD simulations with a Reynolds number of $Re = 6,500,000$. To capture viscosity effects, a triangulated boundary layer mesh with 7 layers was integrated into the mesh. The Y+ value 1.0 for the first layer and mesh layers growing exponentially to meet the background volume mesh size by layer 7 were chosen. No wall functions were used. Instead, a Spalart-Allmaras turbulence model was applied at a temperature $T = 288.15K$ and zero angle of attack. Note, that the emphasis of this optimisation concept proof study was in the understanding of the optimiser characteristics over generating highly accurate CFD simulations. As such, results are not analysed and compared for their error even though the test set up should produce sufficient accuracy.

A total number of 10 control nodes C were selected for the problem with either a fixed position or an explorable design space of $x_C \in [-0.3, 0.3]$ and $y_C \in [-0.3, 0.3]$. For every control node on the lower surface a control node on the upper surface was selected and the displacements were coupled to ensure the thickness of the engine intake lip is maintained. The first pair of control nodes at position $x = 0$ are fixed and have a lambda value of $\lambda = 0$. This allows local control of only the upper lip of the engine intake whilst the remaining geometry is preserved in its shape. The pair of control nodes at position $x = 8.01$ have a lambda value $\lambda = 0$ as well. All other control nodes exhibit a lambda value of $\lambda = 0.1$. Considering one pair of control nodes to be fixed, the total number of degrees of freedom amounts to 8 using equation 1 resulting in 80 Nests according to the convention discussed in section 2.3.1. The design space has been normalized in relation to L of equation 13. Figure 17 (b) visualises the location of all control nodes including the specified explorable design space. Table 1 summarizes all parameters and may be found in the conclusion section.

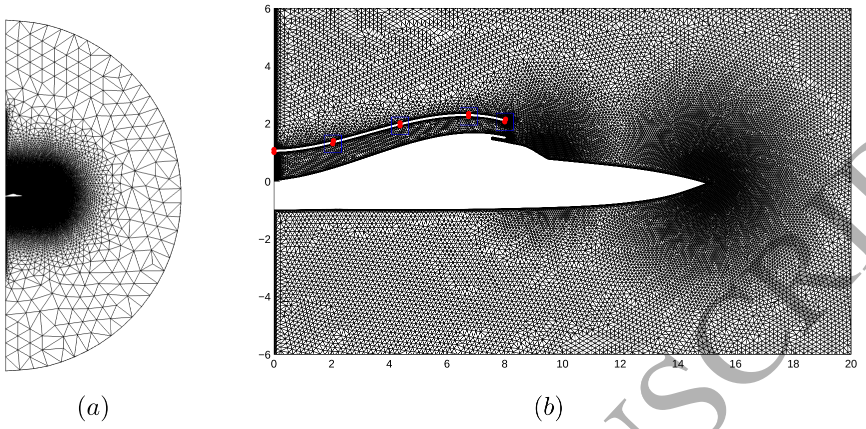


Figure 17: Mesh of supersonic vehicle with engine intake duct (a) total mesh (b) detailed mesh including the control node locations and their range of motion

3.5. *Case 3: Distortion*

The first case is a validation case applying the equivalent set up of purely linear geometry deformation example in [31]. This allows a direct comparison of the effect of the discrete boundary smoothing method proposed in this paper.

3.5.1. *Fitness (objective function)*

One of two main flow parameters considered most crucial in the design of a jet engine intake is distortion. It provides a measure of standard deviation of the total pressure across a plane of interest, in this case the jet engine compressor face. The deviation across the plane should be minimized for an optimum flow pattern into the engine. In the context of optimization, fitness is always maximized, thus, in order to minimize distortion, fitness is a product of the negative distortion σ.

$$F_1 = -\sigma = -\int_0^L \frac{\left|P_t - \bar{P}_t\right|}{\bar{P}_t L}\,\mathrm{d}l \tag{13}$$

where P_t is the total pressure and $\bar{P}_t$ is the mean total pressure. l is a coordinate moving along a line in 2D, both of which are defined along the engine inlet. The equation is normalized against L, which is the length of the engine inlet.

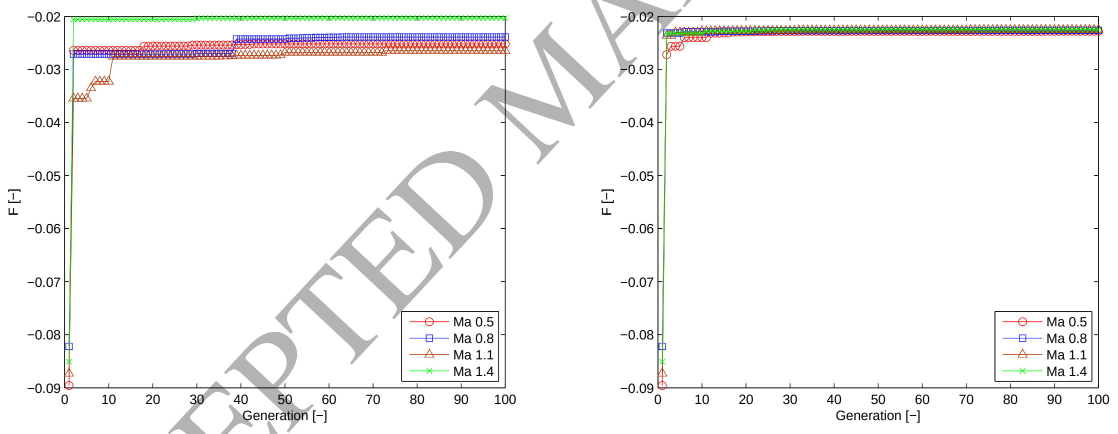
3.5.2. *Results*

An overview of the fitness development for all Mach numbers $Ma = [0.5, 0.8, 1.1, 1.4]$ compared with a previous case without smoothing are illustrated below in figure 18. The fitness development shows the expected improvement particularly in the first generations. When compared to the previous case without smoothing, a faster convergence rate is detectable whereas the overall improvement is similar with a better mean outcome of 1.54 % with smoothing. Details of the fitness values are shown in Table 1 and the table may be found in the conclusion section.

Figure 19 visualises the resulting flow pattern across all Mach numbers compared with the original flow field and the flow results without smoothing applied. The emphasis was put on the engine inlet to analyse the distortion. As a general observation for both cases a more homogeneous pressure pattern was obtained across the engine inlet leading to a reduction in pressure distortion (fitness). The level of improvement is similar as the fitness plot already suggested. Also, the displacement of the control nodes is comparable as the resulting shapes have similar features giving confidence in the algorithm providing consistent results.

Having used non–zero lambda values, non–linear shape deformation was achieved resulting in shapes without sharp geometric features along the deformed boundary. All final shapes utilized the converging/diverging nozzle effect by which a more consistent flow pattern is obtained throughout the expansion phase.

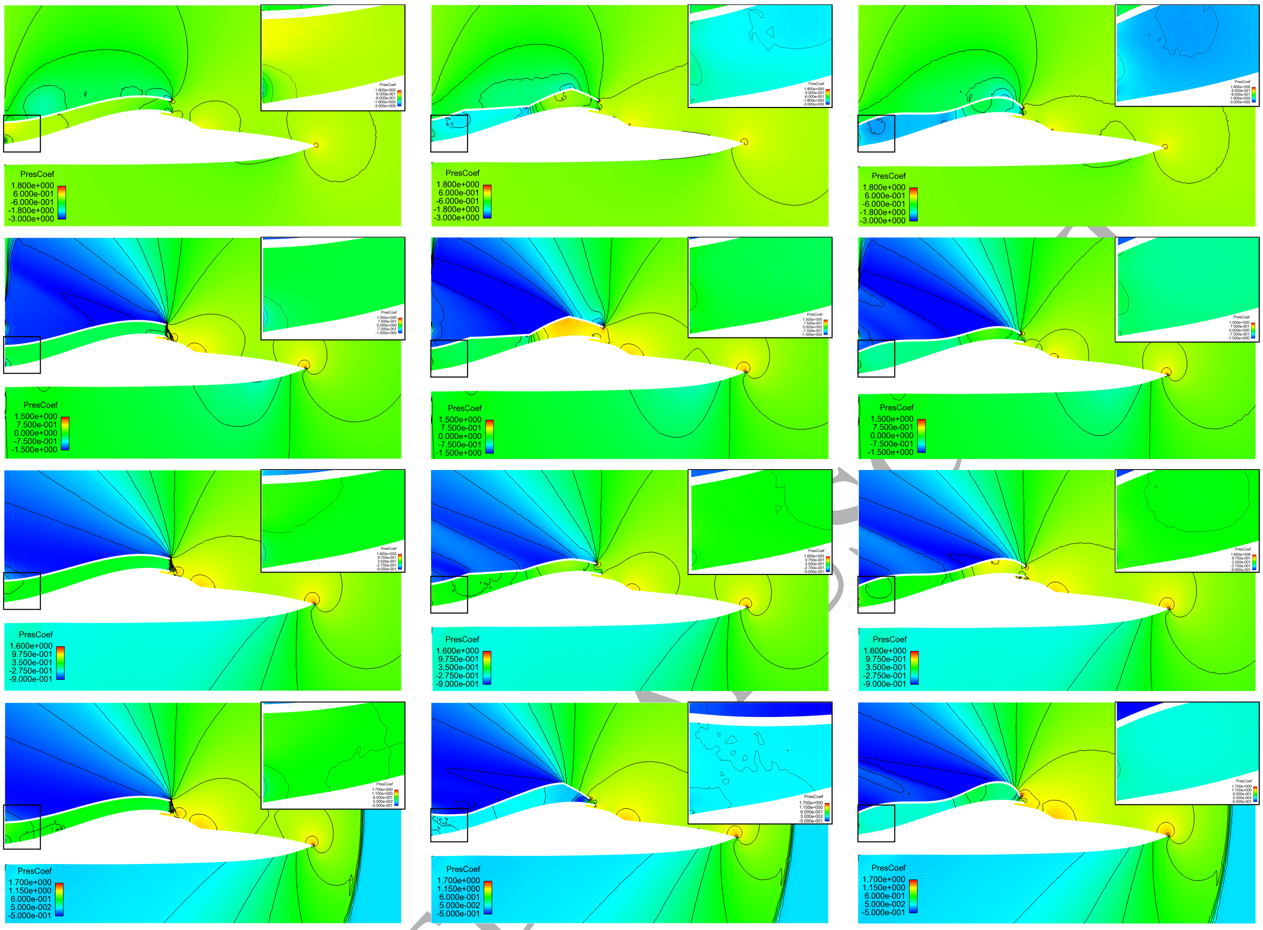
Nonetheless, the final geometries obtained seem impractical which may be due to the single–objective nature of the fitness function. A weighted objective optimisation approach including other important factors in the design such as pressure recovery would most likely result in more practical shapes. This will be investigated in the following case study.



(*a*)

(*b*)

Figure 18: Fitness development over cuckoo generations of case (a) linear deformation (b) discrete boundary smoothing



(a) (b) (c)

Figure 19: Pressure Coefficient Fields of the supersonic vehicle with engine intake duct: (a) initial geometry (b) linear deformation (c) discrete boundary smoothing for Ma 0.5, 0.8, 1.1, 1.4

3.6. Case 4: Distortion and Pressure Recovery

For the second case, the complexity of the case study was increased by integrating another objective into the fitness function. This ultimately led to a constrained optimisation and a weighted objective optimisation case. The study was conducted at Mach numbers $Ma = 0.5$ and $Ma = 1.4$. In contrast to the previous case study the number of generations was limited to 50 as previously convergence has been reached well below 100 generations.

3.6.1. Fitness (objective function)

Besides distortion, pressure recovery is another crucial factor in the design of a jet intake duct. Pressure recovery P_{rec} is a measure of the energy losses in the flow and defined as the ratio of ambient total pressure $\bar{P}_{t\infty}$

to the mean total pressure $\bar{P}_t$ across the engine compressor face $P_{rec} = \frac{\bar{P}_t}{P_{t\infty}}$. One option is to define pressure recovery as a constraint added to the existing fitness function:

$$F_1 = -\sigma = -\int_0^L \frac{|P_t - \bar{P}_t|}{\bar{P}_t L} \, \mathrm{d}l$$

subject to the constraint

$$S = P_{rec} \geq 0.90 \tag{14}$$

As a second option pressure recovery may be incorporated directly into equation 14 to perform a weighted objective optimisation. The maximum value for pressure recovery is $max(P_{rec}) = 1$ so by setting $(\frac{\bar{P}_t}{P_{t\infty}} - 1)$ both distortion and pressure recovery share the same theoretical minimum of 0 and the fitness function becomes

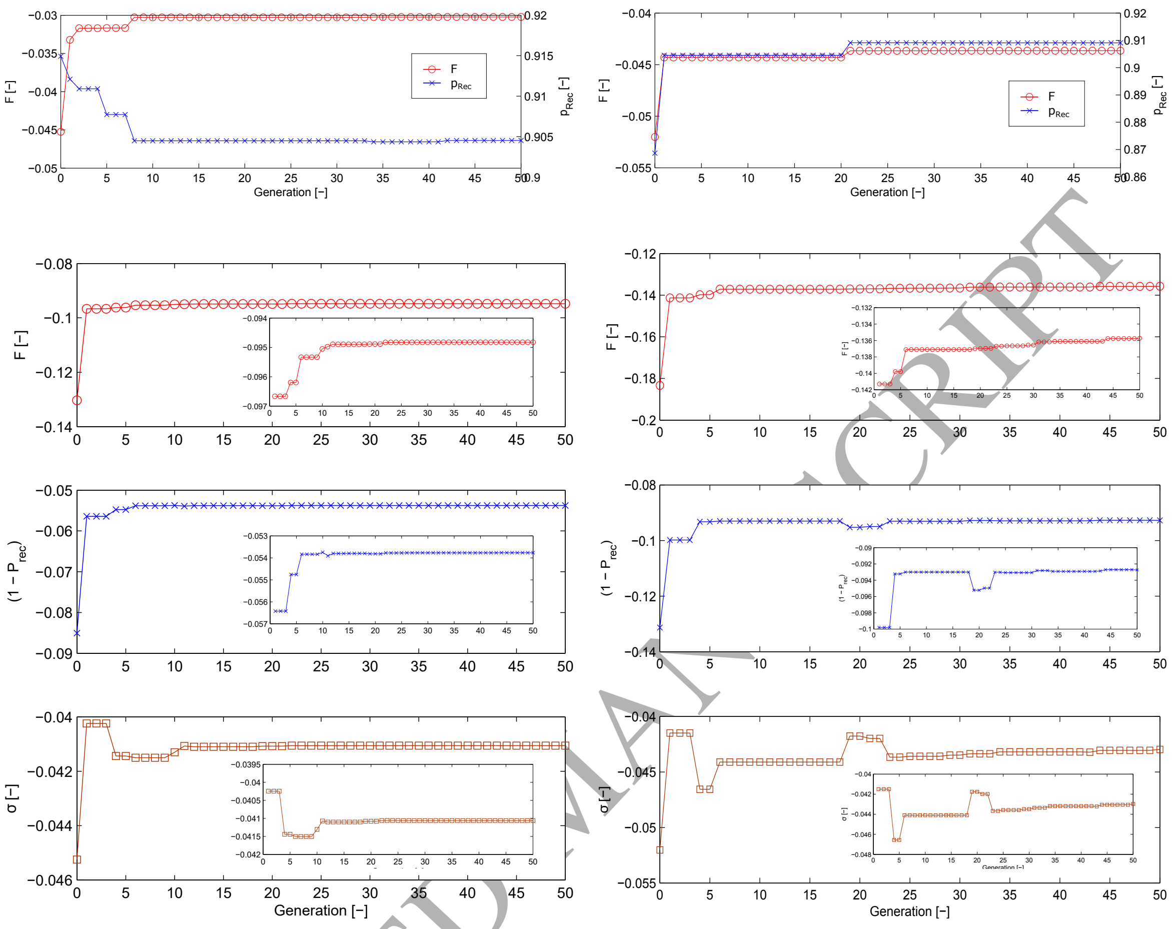
$$F_2 = -\int_0^L \frac{|P_t - \bar{P}_t|}{\bar{P}_t L} \, \mathrm{d}l + (\frac{\bar{P}_t}{P_{t\infty}} - 1) \tag{15}$$

3.6.2. Results

Figure 20 shows the fitness development of both options at Mach numbers $Ma = [0.5, 1.4]$ and Table 1 provides details of all values including start and end values of the fitness. It is clear to see that a great improvement has been achievement in all cases. Note, that the improvement in the first generation is due to the Latin Hypercube Sampling. As such for F_2 a second version of the graphs were generated excluding the first step to highlight the working of the MCS itself. Also, the fitness function was separated into its components distortion and pressure recovery to illustrate the conflict between these two contradicting objectives.

The first fitness option F_1 shows a great improvement of the distortion of about 33 % at a Mach number of 0.5 whilst the pressure recovery value drops from 91.49 % to 90.45 %. For a Mach number of 1.4 an improvement of about 16 % was achieved with an increase in the pressure recovery value from 86.87 % to 90.91 %. Whilst for both Mach numbers the constraint S is fulfilled, the upper figure 20 outlines the conflict occurring at a Mach number of 0.5 between both objectives. Distortion aims at maintaining homogeneous flow whilst pressure recovery only targets a maximum intake of energy and hence the fitness (distortion) increases whilst pressure recovery P_{rec} decreases. For a Mach number of 1.4, supersonic flow causes an opposite reaction. In order to obtain more homogeneous flow, the duct needs to open up which at the same time cancels the shock wave on the engine inlet. This in turns reduces pressure losses so in total an increase in distortion coincides with an increase in pressure recovery.

The second fitness option F_2 proves, that both objectives can be improved at the same time, however, it again remains a compromise between both objectives and distortion is sacrificed for gains in pressure recovery. Hence, the overall improvement in distortion by 9.3 % is smaller compared to the first fitness option. The improvement is mainly achieved in P_{rec} stepping from a value 91.4 % to 94.6 % for a Mach number of 0.5 and from 86.87 % to 90.72 % for a Mach number of 1.4, a great improvement in jet engine intake design [62].



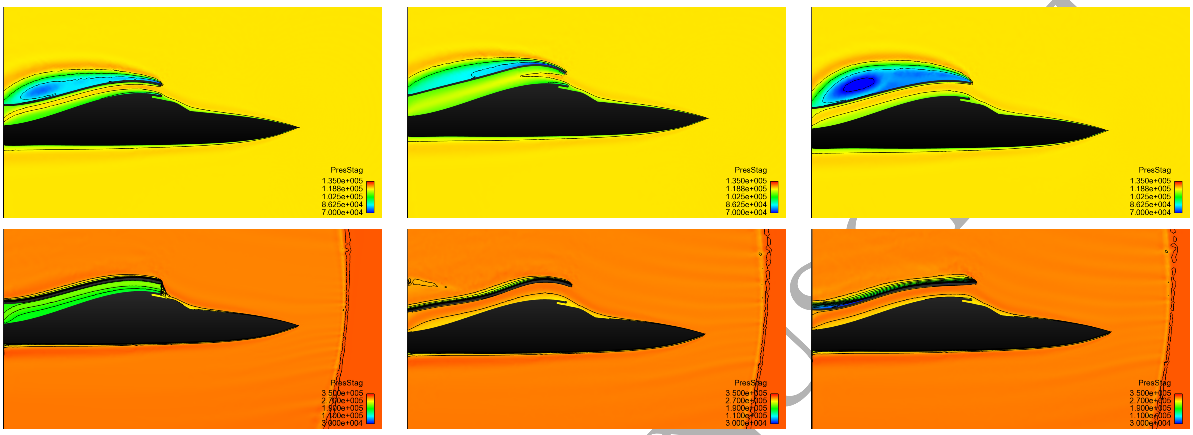
(*a*) (*b*)

Figure 20: Fitness development over cuckoo generations for (a) $Ma = 0.5$ (b) $Ma = 1.4$ using fitness (upper) F_1 with constraint S (lower) F_2 including the evolution of the separate components of the fitness function and a version of each graph excluding the first step of the initial sampling

All these findings are supported by figure 21. For a Mach number of 0.5, a great reduction in distortion in figure 21 (b) (upper) is apparent according to the reduced number of pressure contours towards the engine inlet face. In figure 21 (c) (upper) the total pressure in the engine intake is increased whilst the distortion has not changed noticeably. Note, that here total pressure is plotted as opposed to the pressure coefficient of previous examples. When comparing both options with the previous case study 3.2 it can be observed, that the upper lip was kept straight, in fact, the curvature has reduced compared to the original geometry. For fitness option F_1 the leading edge of the lip was slightly closed to again use the converging/diverging nozzle effect whereas for the fitness option F_2 the leading edge of the lip was opened up to reduce disturbance and hence pressure losses in the flow.

For a Mach number of 1.4, the results are very similar between F_1 and F_2 since the outcome is dominated by the effect of the vanished shock wave.

Given the same degrees of freedom and location of control nodes compared to 3.2 it has generated a more practical aerodynamic shape only due to a change in the objective function. This illustrates, that the developed algorithm provides a very flexible explorable design space for shape optimisation being capable of generating a wide range of shapes with only few degrees of freedom.



(a) (b) (c)

Figure 21: Total Pressure fields of the supersonic vehicle with engine intake duct: (a) initial geometry (b) using F_1 with constraint S (c) using F_2 (for (upper) $Ma = 0.5$ (lower) $Ma = 1.4$)

4. Conclusions

An automated aerodynamic optimisation algorithm has been presented using the novel concept of control nodes [31] for parameterising a computational mesh. The discrete boundary smoothing has been developed to enhance the parameterisation and boundary deformation approach allowing for control over the smoothness with linear and non-linear deformation along the same boundary as well as local control over the geometry. Since the parameterisation is directly applied onto the computational mesh, no re-meshing step is required and CAD to CFD transfer issues are reduced. The approach is coupled with the FDGD mesh movement technique to propagate the boundary deformation throughout the domain mesh. The algorithm is driven by Modified Cuckoo Search, an evolutionary optimisation method that links the CFD solver with the parameterisation. The resulting algorithm was successfully applied to match the pressure distribution of a RAE2822 aerofoil and the optimisation of an intake duct at subsonic, transonic and supersonic flow conditions for a range of different optimisation configurations including constrained and weighted objective optimisation. It has demonstrated to be effective in terms of its intuitive yet efficient parameterisation that exhibits great flexibility. It has also demonstrated to be effective for a range of different applications both in terms of geometries as well as number of objectives and constraints. Future work will look into extending it to three dimensions and adding a Reduced Order Model to accelerate the CFD solution procedure. It has also been recognized that control node locations itself could be optimized and this will be investigated in the future.

Case	DoF	Generations	Ma	Re	λ	Fitness		
						Start	After LHS	End
3.2	4	500	N/A	N/A	0.1 & 0	-0.0141	-0.0085	-0.0057
3.2	8	500	N/A	N/A	0.1 & 0	-0.0141	-0.0121	-0.0021
3.2	12	500	N/A	N/A	0.1 & 0	-0.0141	-0.0141	-0.0011
3.2	18	500	N/A	N/A	0.1 & 0	-0.0141	-0.0141	-0.0010
3.2	11	500	N/A	N/A	0.1 & 0	-0.0141	-0.0140	-0.0006
3.3	11	700	0.729	$6.5e5$	0.1 & 0	-0.429	-0.429	-0.078
[31]	8	100	0.5	$6.5e5$	0	-0.089	-0.027	-0.025
[31]	8	100	0.8	$6.5e5$	0	-0.082	-0.027	-0.024
[31]	8	100	1.1	$6.5e5$	0	-0.087	-0.035	-0.026
[31]	8	100	1.4	$6.5e5$	0	-0.085	-0.021	-0.020
3.5	8	100	0.5	$6.5e5$	0.1 & 0	-0.089	-0.027	-0.022
3.5	8	100	0.8	$6.5e5$	0.1 & 0	-0.082	-0.023	-0.022
3.5	8	100	1.1	$6.5e5$	0.1 & 0	-0.087	-0.024	-0.022
3.5	8	100	1.4	$6.5e5$	0.1 & 0	-0.085	-0.023	-0.022
3.6 F_1	8	50	0.5	$6.5e5$	0.1 & 0	-0.045	-0.033	-0.030
3.6 F_2	8	50	0.5	$6.5e5$	0.1 & 0	-0.130	-0.097	-0.094
3.6 F_1	8	50	1.4	$6.5e5$	0.1 & 0	-0.052	-0.045	-0.043
3.6 F_2	8	50	1.4	$6.5e5$	0.1 & 0	-0.183	-0.141	-0.135

Table 1: Summary table of all input parameters and results. Included are the fitness values after the Latin Hypercube Sampling of generation 1 to extract the effect of the MCS algorithm.

Acknowledgements

The authors would like to acknowledge the support provided by Fujitsu and the HPC Wales for provision of PhD studentship funding and computational support.

References

[1] A. Jameson and J. C. Vassberg, "Computational fluid dynamics: Its current and future impact," *AIAA Paper 2001-0538*, 2001.

[2] A. J. Keane and P. B. Nair, *Computational Approaches for Aerospace Design.* Wiley, June 2005.

[3] T. Hughes, J. Cottrell, and Y. Bazilevs, "Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement," *Computational Methods of Applied Mechanical Engineering*, vol. 194, no. 1, pp. 4135–4195, 2005.

[4] B.-U. Park, Y.-D. Seo, O. Sigmund, and S.-K. Youn, "Shape optimization of the stokes flow problem based on isogeometric analysis," *Structural Multidisciplinary Optimisation*, vol. 48, no. 1, pp. 965–977, 2013.

[5] P. Nortoft and J. Gravesen, "Isogeometric shape optimization in fluid mechanics," *Structural Multidisciplinary Optimisation*, vol. 48, no. 1, pp. 909–925, 2013.

[6] R. Sevilla, S. Fernández-Méndez, and A. Huerta, "NURBS-enhanced finite element method (NEFEM): A seamless bridge between CAD and FEM," *Archives of Computational Methods in Engineering*, vol. 18, no. 4, pp. 441–484, 2011.

[7] R. Sevilla, S. Fernández-Méndez, and A. Huerta, "3D-NURBS-enhanced finite element method (NEFEM)," *International Journal for Numerical Methods in Engineering*, vol. 88, no. 2, pp. 103–125, 2011.

[8] J. Samareh, "A review of shape parameterization techniques," *NASA*, vol. 209136, pp. 333–343, 1999.

[9] S. Shahpar, "Challenges to overcome for routine usage of automatic optimisation in the propulsion industry," *The Aeronautical Journal*, vol. 115, no. 1172, 2011.

[10] R. M. Hicks and P. A. Henne, "Wing design by numerical optimization," *Journal of Aircraft*, vol. 15, no. 7, pp. 407 – 412, 1978.

[11] R. Picket, M. Rubinstein, and R. Nelson, "Automated structural synthesis using a reduced number of design coordinates," *AIAA Journal*, vol. 11, no. 4, pp. 4994–498, 1973.

[12] A. Watt and M. Watt, *Advanced animation and rendering techniques.* Addison-Wesley, 1992.

[13] T. Sederberg and S. Parry, "Free-form deformation of solid geometric models," *Computer Graphics*, vol. 20, no. 4, p. 151160, 1986.

[14] C. Allen and T. Rendall, "Aerodynamic shape optimisation of hovering rotors using compressible cfd," *The Aeronautical Journal*, vol. 115, no. 1170, pp. 513–519, 2011.

[15] S. Xu, W. Jahn, and J.-D. Mller, "Cad-based shape optimisation with cfd using a discrete adjoint," *International Journal for Numerical Methods in Fluids*, 2013.

[16] M. J. Martin, E. Andres, M. Widhalm, P. Bitrian, and C. Lozano, "Non-uniform rational b-splines-based aerodynamic shape design optimization with the dlr tau code," *Journal of Aerospace Engineering*, vol. 226, pp. 1225–1242, 2011.

[17] W. Anderson, S. Karman, and C. Burdyshaw, “Geometry parameterisation method for multidisciplinary applications,” *AIAA Journal*, vol. 17, no. 6, pp. 1658–1578, 2009.

[18] H. Sobieczky, “Parametric Airfoils and Wings,” *Notes on Numerical Fluid Mechanics*, vol. 68, pp. 71–87, 1999.

[19] A. M. Morris, C. B. Allen, and T. C. S. Rendall, “Cfd-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation,” *International Journal for numerical Methods in Fluids*, vol. 58, pp. 827 – 860, 2008.

[20] B. Kulfan and J. Bussoletti, “Fundamental parametric geometry representation for aircraft component shapes,” *Proceedings of 11th AIAA Multidisciplinary Analysis and Optimisation Conference*, vol. 39, no. 6948, pp. 549–567, 2006.

[21] B. M. Kulfan and R. Hilton, “A universal parametric geometry representation method cst a universal parametric geometry representation method cst ,” *Aiaa-07-0062*, no. January, pp. 1–36, 2007.

[22] S. Walton, “Phd thesis: Gradient free optimisation in selected engineering applications,” *Swansea University*, 2013.

[23] S. Nadarajah, P. Castonguay, and A. Mousavi, “Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization,” *18th AIAA Computational Fluid Dynamics Conference*, vol. AIAA, 2007.

[24] S. S. Sarakinos, E. Amoiralis, and I. K. Nikolos, “Exploring freeform deformation capabilities in aerodynamic shape parameterization,” *EUROCON, Belgrade*, pp. 535–539, November 2005.

[25] S. Coquillart, “Extended free-form deformation: A sculpturing tool for 3d geometric modeling,” *SIGGRAPH*, vol. 24, no. 4, pp. 187–196, 1990.

[26] J. A. Samareh, “Aerodynamic shape optimization based on free-form deformation,” *AIAA Paper*, vol. 1, no. September, pp. 1–12, 2004.

[27] N. Sevant, M. Bloor, and M. Wilson, “A hierarchical approach to optimal aerodynamic design,” *American Institute of Aeronautics and Astronautics*, vol. 2, no. 98, pp. 1099–1107, 1998.

[28] G. R. A. Keane, “Concise orthogonal representation of supercritical airfoils,” *Journal of Aircraft*, vol. 38, no. 3, pp. 580–583, 2001.

[29] A. Jameson and J. C. Vassberg, “Influence of shape parameterization on aerodynamic shape optimization,” *Von Karman Institute*, pp. 1–51, 2014.

[30] M. J. Martin, E. Andres, C. Lozano, and E. Valero, “Volumetric b-splines shape parametrization for aerodynamic shape design,” *Aerospace Science and Technology*, vol. 37, pp. 26–36, 2014.

[31] D. S. Naumann, B. Evans, S. Walton, and O. Hassan, “A novel implementation of computational aerodynamic shape optimisation using modified cuckoo search,” *Applied Mathematical Modelling*, 2015.

[32] A. de Boer, M. van der Schoot, and H. Bijl, "Mesh deformation based on radial basis function interpolation," *Computers and Structures*, vol. 85, pp. 784–795, 2007.

[33] X. Liu, N. Qin, and H. Xia, "Fast dynamic grid deformation based on delaunay graph mapping," *Journal of Computational Physics*, vol. 211, pp. 405–423, 2006.

[34] S. Jakobsson and O. Amoignon, "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization," *Computers and Fluids*, vol. 36, pp. 1119–1136, 2007.

[35] J. Witteveen and H. Bijl, "Explicit mesh deformation using inverse distance weighting interpolation," *AIAA San Antonio, Texas*, pp. 1–10, June 2009.

[36] L. Jing, G. Zhenghong, H. Jiangtao, and Z. Ke, "Aerodynamic design optimization of nacelle/pylon position on an aircraft," *International Forum on Aeroelasticity and Structural Dynamics*, vol. 26, no. 4, pp. 850–857, 2013.

[37] A. Jameson, W. Schmidt, and E. Turkel, "Numerical solution of the euler equations by finite volume methods using runge-kutta time-stepping schemes," *AIAA 14th Fluid and Plasma Dynamic Conference*, pp. 1–27, 1981.

[38] A. Jameson, ed., *Efficient Aerodynamic Shape Optimization*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004.

[39] M. Harbeck and A. Jameson, eds., *Exploring the Limits of Shock-free Transonic Airfoil Design*, 43rd Aerospace Sciences Meeting and Exhibition, 2005.

[40] S. Walton et al., "Modified cuckoo search: A new gradient free optimisation algorithm," *Chaos, Solitons and Fractals*, 2011.

[41] S. Walton, O. Hassan, and K. Morgan, "Reduced order mesh optimisation using proper orthogonal decomposition and a modified cuckoo search," *International Journal for Numerical Methods in Engineering*, vol. 93, pp. 527–550, 2013.

[42] H. Salimi, D. Giveki, M. Soltanshahi, and J. Hatami, "Extended mixture of mlp experts by hybrid of conjugate gradient method and modified cuckoo search," *International Journal of Artificial Intelligence & Applications*, vol. 3, 2012.

[43] X.-S. Yang and S. Deb, eds., *Cuckoo search via Lévy flights*, Proceedings of World Congress on Nature & Biologically Inspired Computing, IEEE Publications, 2009.

[44] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, pp. 17–35, 2013.

[45] A. Natarajan and S. Subramanian, eds., *Bloom Filter Optimization using Cuckoo Search*, Proceedings of the 2012 International Conference on Computer Communication and Informatics, 2012.

[46] G. Selvi and T. Purusothaman, "Cryptanalysis of simple block ciphers using extensive heuristic attacks," *European Journal of Scientific Research*, vol. 78, no. 4, pp. 198–221, 2012.

[47] E. Speed, "Evolving a mario agent using cuckoo search and softmax heuristics," *2nd International IEEE Consumer Electronics Societys Games Innovations Conference*, 2010.

[48] R. A. Vazquez, ed., *Training spiking neural models using cuckoo search algorithm*, 2011 IEEE Congress on Evolutionary Computation, 2011.

[49] M. D. McKay, R. J. Beckmann, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[50] S. Walton, O. Hassan, and K. Morgan, "Selected engineering applications of gradient free optimisation using cuckoo search and proper orthogonal decomposition," *Archives of Computational Methods in Engineering*, vol. 20, pp. 123–154, 2013.

[51] B. Evans, T. Morton, L. Sheridan, O. Hassan, K. Morgan, J. W. Jones, M. Chapman, R. Ayers, and I. Niven, "Design optimisation using computational fluid dynamics applied to a landbased supersonic vehicle, the bloodhound ssc," *Structural Multidisciplinary Optimization*, vol. 47, pp. 301–316, 2013.

[52] K. A. Sørensen, "Phd thesis: A multigrid accelerated procedure for the solution of compressible fluid flows on unstructured hybrid meshes," *Swansea University*, 2001.

[53] P. R. Spalart and S. R. Allmaras, "A one–equation turbulence model for aerodynamic flows," *La Recherche Aerospatiale*, vol. 1, pp. 5–21, 1994.

[54] B. J. Evans, O. Hassan, J. W. Jones, K. Morgan, and L. Remaki, "Computational fluid dynamics applied to the aerodynamic design of a land–based supersonic vehicle," *Numerical Methods for Partial Differential Equations*, vol. 27, pp. 141–159, 2010.

[55] P. Cook, M. McDonald, and M. Firmin, "Aerofoil rae2822 - pressure distributions and boundary layer and wake measurements," *AGARD Report AR 138, NATA*, 1979.

[56] M. Guenot, I. Lepot, C. Sainvitu, J. Goblet, and R. F. Coelho, "Adaptive sampling strategies for non-intrusive POD-based surrogates," *Engineering Computations*, vol. 30, no. 4, pp. 521–547, 2013.

[57] X. Han and D. W. Zingg, "An adaptive geometry parametrization for aerodynamic shape optimization," *Optimization and Engineering*, vol. 15, no. 1, pp. 69–91, 2014.

[58] NASA, "Rae 2822 transonic airfoil: Study 1," *http://www.grc.nasa.gov/WWW/wind/valid/raetaf/raetaf01/raetaf01.html*, 2002.

[59] P. Castonguay and S. Nadarajah, "Effect of shape parameterization on aerodynamic shape optimization," *AIAA*, vol. 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.

[60] S. Allright, "Multiblock topology specification and grid generation for complete aircraft configurations," *Applications of Mesh Generation to Complex 3D Configurations (Conf. Proc. No 464)*, pp. 11.1–11.11, 1989.

[61] H. Kim and M.-S. Liou, “Shape design optimization of embedded engine inlets for n2bh ybrid wing-body configuration,” *Aerospace Science and Technology*, vol. 30, pp. 128–149, 2013.

[62] D. L. Rodriguez, “Multidisciplinary optimization of a supersonic inlet using a cartesian cfd method,” *AIAA Multidisciplinary Analysis and Optimization Conference*, vol. 4492, pp. 1–14, 2004.