



Bevan, R., Poole, D., Allen, C., & Rendall, T. (2016). Optimization of Vane-Type Vortex Generators for Tiltrotor Wings using Computational Fluid Dynamics. In 54th AIAA Aerospace Sciences Meeting. [AIAA 2016-1304] American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2016-1304

Peer reviewed version

Link to published version (if available):
[10.2514/6.2016-1304](https://doi.org/10.2514/6.2016-1304)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via AIAA at <http://arc.aiaa.org/doi/abs/10.2514/6.2016-1304>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

Optimization of Vane-Type Vortex Generators for Tiltrotor Wings using Computational Fluid Dynamics

R.L.T. Bevan*, D.J. Poole†, C.B. Allen‡, T.C.S. Rendall§

Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, U.K.

Tiltrotor wings are thick and highly loaded airfoils, and are thus particularly susceptible to separation and early onset buffet and vortex generators are commonly used to alleviate these issues. Hence, the design of counter-rotating vortex generators on tiltrotor wings to control separation is considered. A representative tiltrotor wing airfoil (a modified NACA 64(4)-421) has been tested at a Reynolds number of 7.5 million ($M = 0.24$) and $C_L = 1.3$ to demonstrate that a large separation is problematic in this flow case. Five parameters have been introduced to quantify the vortex generator design space (length, aspect ratio, angle, chord location and spacings ratio) and Latin Hypercube sampling of this design space has been undertaken using high fidelity CFD to construct a radial basis function based surrogate model. This model has been interrogated and refined using an efficient adaptive sampling method which incorporates both space-filling and local refinement properties in order to achieve an optimum solution. To determine this optimum, a global search algorithm based on differential evolution was used for optimization of the model. The results obtained from the optimization eliminated the separation experienced by the clean geometry (from 11.1% to 0%), whilst reducing the drag obtained from the best Latin Hypercube sample by 7%.

I. Introduction

Tiltrotor aircraft combine the vertical lift capability of a helicopter with the speed and endurance of a conventional fixed-wing aircraft. In forward flight, the aircraft lift is provided by the wings. However, due primarily to structural design considerations, these are very stiff, low-aspect ratio wings, which inhibit efficient aerodynamic performance. Due to a short span and a typical cruise Mach between 0.4 and 0.5, these wings experience a high loading; a typical cruise lift coefficient for a tiltrotor being above 1.0. Structural requirements also result in a thick aerofoil. The Bell XV-15 utilised a NACA 64A223 in this regard.¹ These design factors therefore make tiltrotor aircraft particularly susceptible to stall and early onset buffet, so boundary layer control is critical and upper surface flow control devices are an important consideration.

Flow control devices are adopted regularly in aerodynamic design, to control boundary layer behaviour, and are often retro-fitted where problems are identified. Occasionally, complex active systems have been implemented to control boundary layer growth, including boundary layer suction, blowing, and bleeding, but by far the most common methods are passive schemes, with vortex generators (VGs) the most conventional approach. These involve simple small flat plates normal to the surface, usually rectangular in shape, inclined to the freestream flow. VGs were utilised on the Bell XV-15 tiltrotor to eliminate premature stall and retain attached flow up to an angle of attack of 15 degrees.² The V-22 Osprey tiltrotor also employs VGs; at mid to high angles of attack, experimental data shows that wing-mounted VGs suppress the early onset of separation.³ The majority of VGs protrude into the external flow, i.e. above the boundary layer, but there have also been immersed, or sub-layer, VGs adopted.⁴ The role of a VG is to induce vortical flow via the flow separation from its sharp upper edge. The vortex which subsequently propagates downstream entrains

*Researcher. Email: r.bevan@bristol.ac.uk

†Researcher. Email: d.j.poole@bristol.ac.uk

‡Professor of Computational Aerodynamics. Email: c.b.allen@bristol.ac.uk

§Lecturer. Email: thomas.rendall@bristol.ac.uk

higher energy, higher momentum flow from outside the boundary layer into the lower energy boundary layer. This re-energises the boundary layer, suppressing separation.

It is necessary to understand the physics that the VGs are introducing into the flow to be able to design them to mitigate against unwanted flow phenomena (e.g. separation), and this is particularly important on tiltrotor wings due to the susceptibility to separation or buffet. To understand and quantify the effects of VGs, computational fluids dynamics (CFD) approaches are often employed.^{5,6} However, the modelling of VGs by conventional CFD approaches poses difficult issues in capturing the vortex and its convection downstream. This often requires fine numerical meshes on geometries where the size of the flow control device is orders of magnitude smaller than the global object. Furthermore, a sufficiently high fidelity numerical scheme must be selected, often requiring Reynold’s averaged Navier-Stokes (RANS) solutions with turbulence models.

In order to determine an optimized VG geometry, the high fidelity simulation approach can either be coupled directly to an optimization routine or alternatively, it can be incorporated within a surrogate-based analysis. The latter approach uses a limited number of high fidelity runs to train a surrogate model. This model then provides interpolation of the known sample points to obtain a continuous design space approximation away from the data sites. Using this surrogate-based approach allows the relatively cheap design of the VG compared to the directly coupled approach. Design space interrogation methods can then locate an optimum VG design without the need for large numbers of simulations. This surrogate modelling approach for design of VGs has received only limited coverage in previous publications,^{7–9} including by the use of lower-fidelity simulation approaches.¹⁰

The surrogate-based approach relies on limiting the number high fidelity sample points within the design space, and thus efficient design space interrogation methods are desired. Two sampling strategies for constructing surrogate models were examined in Mackman *et al*¹¹ for construction of aerodynamic models. The two methods employed were adaptive, i.e. information from successive iterations is used to reconstruct the surrogate model between iterations. It was determined that both methods were suitable for aerodynamic application, and both provided better than traditional one-stage or sequential space filling algorithms. An alternative adaptive sampling method for surrogate-based modelling of multidimensional design space, combines both space-filling properties and local refinement.¹²

A surrogate modelling approach was applied to tilt-rotor aircraft wings in Bevan *et al*.¹³ This earlier work introduced a framework of statistical sampling and surrogate modelling, together with a preliminary examination to highlight the potential of the approach. This initial examination utilised a simplified design space, which was optimised for a single design point. The work presented in this paper refines the prior framework and broadens the design space through both increasing variable range and also improving and expanding on the parameters considered. Finally, adaptive sampling is introduced in order to augment interrogation of the surrogate model and further optimise the resulting VG configuration.

II. Computational Modelling of Vortex Generators

The physics of a flowfield when a vortex generator (VG) is placed into the flow differs considerably from a clean geometry. The VG produces a vortex which encourages mixing of turbulent flow downstream of the device, often used to avoid unwanted separation. The exact physics of flow control using VGs is complicated and often requires fine computational meshes with high fidelity flow simulations. The detail in this section gives examples of the type of simulations used for VG modelling.

II.A. Mesh Generation

The precise physical effects and the topology of VGs makes for difficult modelling, both in set-up and in solution methods. Multi-block structured meshes are difficult to produce for an aerofoil with a VG due to the large differences in scale that exist between the parent aerofoil and the flow control device, and the sudden changes in curvature when progressing over the surface of the aerofoil. Furthermore, to effectively capture and convect downstream the vortex produced by the device, fine resolution grids are often required around the VG and in its wake, resulting in the exponential growth of mesh sizes when compared to a clean wing counterpart. Despite the difficulties in producing structured meshes, this meshing approach is generally considered to produce the highest quality meshes that are most suitable to model flow around VGs and so many examples can be found.^{14–19}

To alleviate some of the issues with structured mesh generation, unstructured meshes can instead be

used. The advantage of unstructured meshes is that the algorithms for producing these meshes typically use space-filling or marching approaches and so only require surface and farfield geometry, making their production considerably simpler than multiblock structured meshes. However, to capture the physics of the VG in the flow, a structured prismatic layer must exist on the surface. Examples include.^{20–22}

II.B. Flow Solver

The suitable selection of a CFD solver is also imperative to ensure the complete convection of the vortex downstream from its source without too much dissipation being added. The lowest fidelity simulation of VGs that has been done is by Kerho and Kramer,²³ who used XFOIL,^{24,25} with a modification to the stress-transport equation at the VG location to mimic the effect of a VG. This was demonstrated to reduce the drag values near the separated region of an aerofoil.

In general, full flowfield analysis of a three-dimensional geometry must be performed to fully capture the physics of the vortex generator in the flow, and this must be done by high fidelity RANS, or even LES and DNS solutions. If using a RANS simulation approach then it is of paramount importance to select an appropriate turbulence model to ensure detailed capture of the vortex. Two common turbulence models are Spalart-Almaras (SA)²⁶ and $k-\omega$ SST,²⁷ which have both been used to model flow around a VG. Both tend to under-predict streamwise vorticity of the vortex, though the SST model generally matches the experiment better than SA.^{15,16} This can be alleviated somewhat by including the strain-vorticity modification²⁸ to the SA model. Yaras and Grosvenor²⁹ also compared both models with the $k-\epsilon$ and standard $k-\omega$, and showed that both SA and SST outperformed the $k-\epsilon$ and $k-\omega$ models, but that SA perhaps demonstrated the best balance between prediction, robustness and computational efficiency. The work of Tai⁵ utilised a modified Baldwin-Lomax algebraic model, as well as Menter’s SST model as a supplement to ensure the physical flow features were captured, on a V-22 Osprey simulation with VGs.

A lower-fidelity approach using embedded models within a CFD framework is also a popular approach. Models such as the vortex-source^{30,31} and lifting-force^{10,32–34} have been shown to produce results that mimic the effect of a vortex generator placed in the flow without the need to mesh and solve the full geometry.

To fully capture all the flow-physics associated with the VG, high-fidelity, real-time, simulations via large eddy simulation (LES) or direct numerical simulation (DNS) methods are required, though these are prohibitively expensive, particularly for design. As such, historically it has been difficult to perform such simulations for large Reynolds numbers, however, with the continual increase in computational resources available these are becoming more common.^{20,35,36} The Lattice-Boltzmann method³⁷ has also been used to produce time-dependent flow simulations around VGs.⁶

III. Simulation Approach

As outlined above, the modelling of the vortex generator and resulting flow is not a trivial task; the suitable capture of the physics requires high fidelity solvers and high density numerical meshes. The primary purpose of the research presented in this paper is the simulation of numerous VG configurations, in order to assess the optimal design considerations. A solution method is firstly developed and presented in this section. This involves obtaining high-fidelity viscous flows using OpenFOAM^a with a bespoke mesh generation tool to produce high quality structured viscous meshes around a representative aerofoil section with a vortex generator.

III.A. Mesh Generation

All the meshes generated utilised the C-grid approach. A representative tiltrotor aerofoil, which is a modified NACA 64(4)-421 section was used, with a 0.4% chord blunt trailing edge. A two-dimensional four-block structured multiblock mesh was generated^{38,39} initially, and three views are presented in figure 1; this is for a mesh with 769 points on the aerofoil surface, 97 streamwise points on either side of the wake line, 65 points across the blunt trailing edge, and 129 points in the normal direction.

To include the vortex generator, the two-dimensional mesh is extruded to the correct part-span position, and a high quality mesh deformation scheme used, to rotate the mesh to the required vortex generator

^a<http://www.OpenFOAM.org/>

incidence, and pull in mesh cells from upstream, downstream, above and either side of the generator to match the required local spacings. This is achieved via several local radial basis function deformations.^{40,41}

Figure 2 shows the resulting wing surface mesh near the leading edge and the surface with starboard plane, together with a closer view of the VG; this is for a sample point with 9.85% chord wide surface, with rectangular vortex generator $1.88\% \times 1.31\%$ chord length and height, with leading edge at 27.83% chord from leading edge, 0.5% chord from port plane, at 23.75 degrees incidence to the freestream. The number of spanwise planes extruded is dependent upon the spanwise domain width required. In this instance, 127 spanwise planes were extruded.

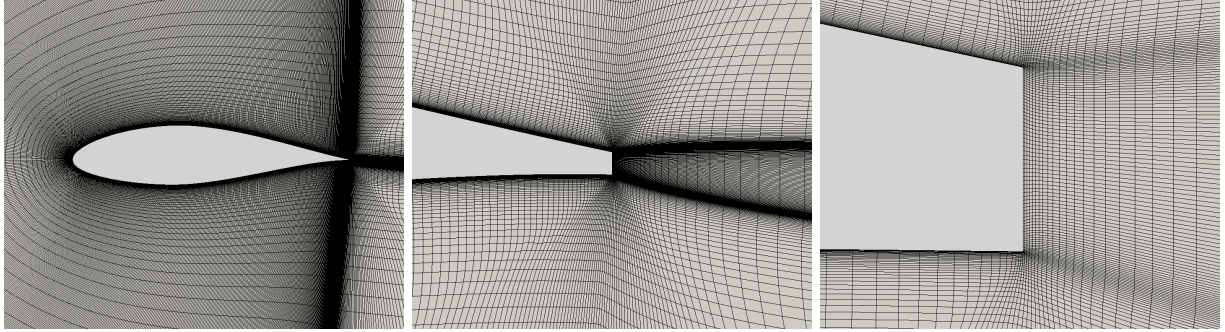


Figure 1: Four-block structured mesh

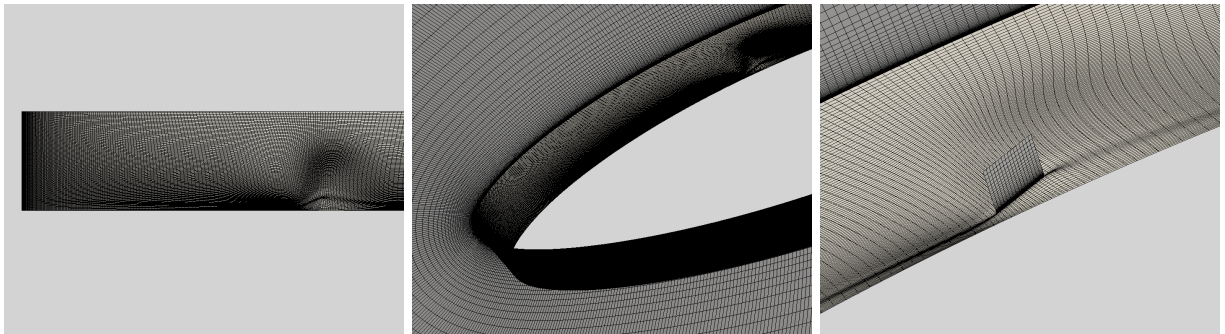


Figure 2: Surface mesh including vortex generator

The final stage in the mesh generation process reduces the cell count away from sections of interest. This is undertaken through point removal and combining of hexahedra cells to form polyhedra. The mesh reduction process typically involves a 50-60% cell count reduction.

III.B. CFD Solver

Once a suitable mesh has been produced, this is used to model the viscous flow around the wing with VGs using OpenFOAM with a RANS model. OpenFOAM is an open source CFD software package, capable of solving problems such as: complex fluid flows involving chemical reactions, turbulence and heat transfer, solid dynamics and electromagnetics.

The flow simulations utilise a unstructured solver based upon the SIMPLE algorithm (Semi-Implicit Method for Pressure-Linked Equations), an iterative procedure to solve the Navier-Stokes equations for steady-state problems. This iterative procedure, as originally implemented in OpenFOAM, relies on basic residual checking for steady state convergence criteria. After modification, steady state convergence is assessed using the standard deviation of the force coefficients.

OpenFOAM has numerous turbulence models that can be utilised, although these models are not necessarily considered standard implementations (e.g. SA-fv3). After modifications, this work utilises the standard Spalart-Allmaras (SA) model as defined by NASA Langley's Turbulence Modelling Resource^b. The standard SA model implemented in this work uses the minimum limiter of $0.3 \times \Omega$ for \hat{S} .

^b<http://turbmodels.larc.nasa.gov>

III.C. Example Solutions

The simulations were undertaken at a Reynolds number of 7.5 million, and a Mach number of 0.24. If present, the VG was located at 11.25%. The distance between two counter-rotating pairs was 5.6% and the intra-pair spacing was 0.5%. The VG height and aspect ratio (defined as length/height) were 1.16% and 1.78 respectively, while the VG angle relative to freestream was 20.25 degrees. All percentages are relative to the airfoil chord. No-slip conditions were applied to both the aerofoil and the VG. The farfield was located at 100 chord lengths from the aerofoil. At the farfield, freestream boundary conditions were imposed, together with a constant pressure at the downstream farfield. While the clean geometry (modelled using a two dimensional mesh) utilises the two dimensional specific span-wise boundary condition within OpenFOAM, the three dimension aerofoil and VG utilised slip conditions. Earlier work¹³ determined that this span-wise boundary condition was suitable.

At a C_L of 1.3 (9 degrees angle of attack for aerofoil with VG, 10 degrees for clean), the clean aerofoil has a significant separated region, as shown in figure 3a.

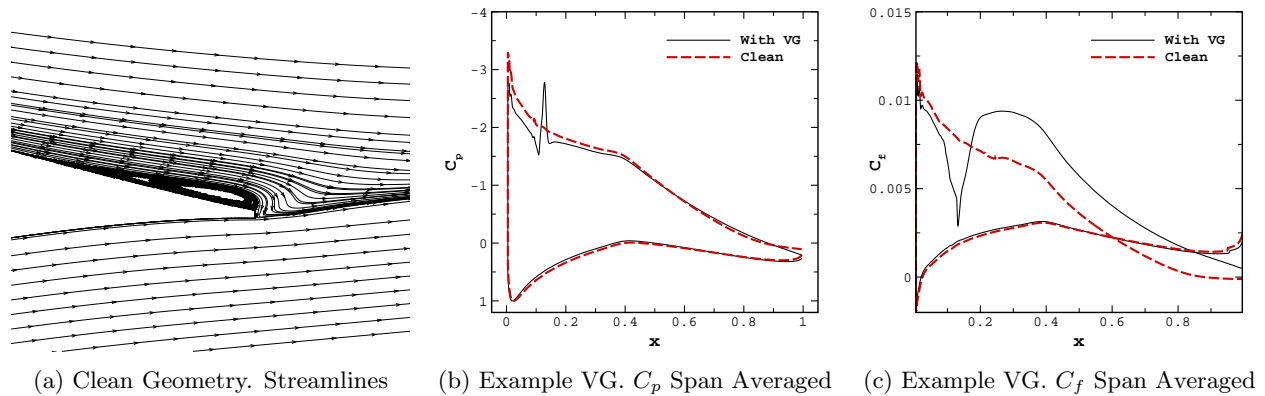


Figure 3: Example solutions around modified NACA 64(4)-421

In contrast, the aerofoil with VG does not have a separation region. This separation is assessed by examination of the x -direction friction coefficient (with x being the stream-wise direction), which is given in figure 3c. From the figure, it is clear that the separation has been eliminated through introduction of the VG, while the clean geometry has separation after 88% chord. The improved performance relative to the clean geometry is also evident in the plot of the pressure coefficient (figure 3b), with the span-wise C_p variation centred around the immediate downstream of the VG location and then diminishing.

IV. Vortex Generator Design Space

A high-fidelity simulation approach that utilises structured numerical meshes and viscous flow solutions by OpenFOAM has been outlined and tested above for the simulation of flows around VGs on infinite wings. The work in this paper is focussed on the process of performing effective design of the vortex generator (in terms of its planform and location on the wing) and this process is outlined in this section. A full design space of a number of design parameters is considered and sampled by the random Latin hypercube statistical sampling method. These sample points are then simulated using the tools developed to allow investigation of the effect of the design parameters on the flow solution around the VGs.

IV.A. Design Parameters

Five design parameters are investigated in this work. These are VG length (l), VG aspect ratio (AR), VG setting angle relative to freestream (θ), VG chordwise location (c) and the spacing ratio (s). The VG aspect ratio is defined as $AR = l/h$, and the spacing ratio is defined as the ratio between the intra-pair spacing (x_1) and the inter-pair spacing (x_2), where the intra-pair spacing is fixed ($x_1 = 0.5\%$). The use of VG aspect ratio, rather than VG height directly was based upon the work of Bevan *et al.*¹³ This earlier initial investigation incorporated both VG height and length as individual parameters. Results obtained during the study demonstrated that an $AR < 1$ produced undesirable flow with significant drag penalties. The five

parameters are given in figure 4.

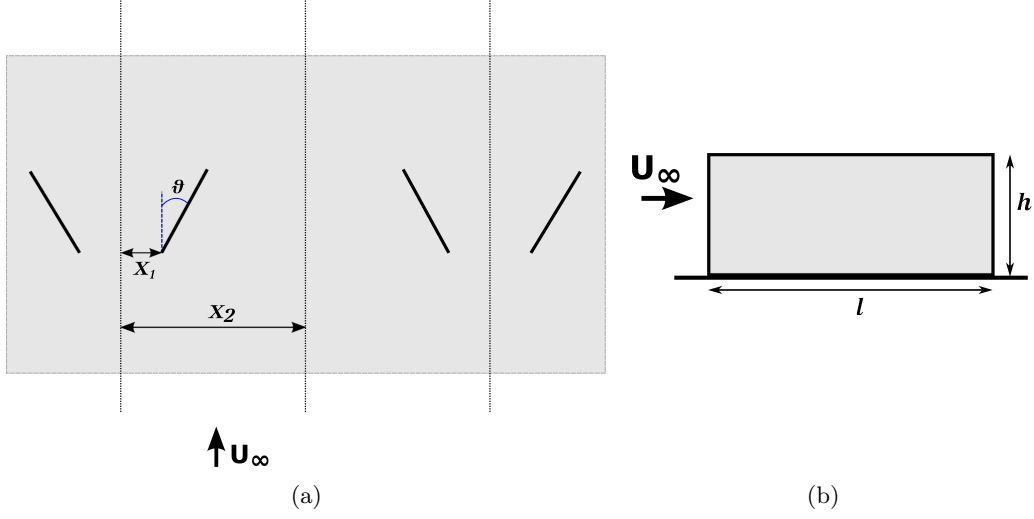


Figure 4: VG Configuration

Rectangular VGs are modelled in this work. The VGs are counter-rotating in nature. The five design parameters were chosen to give a sufficient variation in the flow nature and final force coefficients to design for a chosen objective function. The maximum and minimum values of the design parameters are shown in table 1, and chosen to represent sensible coverage without including extremes of the design space. All parameters are scaled such that the design space is $[0, 1]^5$.

Table 1: Design space range (% based on airfoil chord)

Parameter	Minimum	Maximum
h	0.2%	2%
AR	1	4
θ	5°	30°
c	5%	50%
s	5	20

For tiltrotor aircraft wings, which are thick and highly loaded, early onset buffet can be particularly problematic and is typically linked to separation. For this reason, buffet is one of the primary design considerations in this work, and particularly separation at a chosen design condition. The loading on the wings in cruise can be as high as $C_L = 1.0$, however, it is unlikely that buffet in cruise is a problem. The chosen design loading is therefore higher, and set to be $C_L = 1.3$, which is realistic for a manoeuvre loading.

To test for possible buffet, a simulation around the clean aerofoil has been performed for this loading at a representative flight condition ($Re = 7.5 \times 10^6$, $M_\infty = 0.24$, $C_L = 1.3$) and the flowfield is shown in figure 3a. As can be seen, separation at this flight condition is considerable and therefore it is probable that buffet may also be an issue. The vortex generators are used to alleviate the buffet, which is assumed to be caused by a large separation region. Therefore the objective function (equation 1), against which to design the vortex generators, is to minimize separation subject to minimizing the drag penalty. This objective function will balance drag induced by introduction of the VG and that produced via separation.

$$J = C_D + kS \quad (1)$$

J is the objective function to be minimized, S is the span-average separation chordwise location (as a percentage), and k is a constant used to balance the effect of the two. To ensure equal magnitude of both parameters, k is taken to be 0.0005 such that 1% separation equates to adding 5 drag counts to the problem.

IV.B. Latin Hypercube Sampling

To efficiently sample the design space, and reduce the number of function evaluations required from that of brute force sampling, a Latin Hypercube⁴² (LHC) statistical sampling approach is used. A Latin Hypercube sampling approach is a statistics-based approach designed to produce an optimum configuration of sample points through the design space; the design space produced has each division in all dimensions sampled by exactly one sample point. Corners of the sample space are added to the sampling as this is important for interpolation, which is the basis of a surrogate modelling approach developed below. Whilst a random LHC was utilised in,¹³ an optimised LHC offers an improvement in providing a more evenly sampled design space for construction of the surrogate model. Numerous methodologies exist for construction of an optimised LHC, however the detailing of these are beyond the scope of this paper. The approach chosen utilised the Morris and Mitchell⁴³ ϕ_p criteria

$$\phi_p(X) = \left(\sum_{j=1}^m g_j d_j^{-p} \right)^{1/p} \quad (2)$$

where d represents the distance between two points in hypercube X and g represents the number of occurrences of this distance within the hypercube. d can be calculated either with the absolute norm or the Euclidean norm. p is a positive integer. Minimizing ϕ_p improves the space-filling properties of the hypercube. As recommended by Morris and Mitchell, this work minimizes ϕ_p for $p = [1, 2, 5, 10, 20, 50, 100]$ then the best of the resulting hypercubes is used. Since the possible combinations for a 5 dimension hypercube with only 100 sample points is of the order of 10^{790} , an optimization algorithm is required. This work uses the Enhanced Stochastic Evolution algorithm proposed by Jin *et al.*⁴⁴ An example of the resulting Latin Hypercube sample is shown in figure 5 for 100 sample points in two-dimensions plus the corner points.

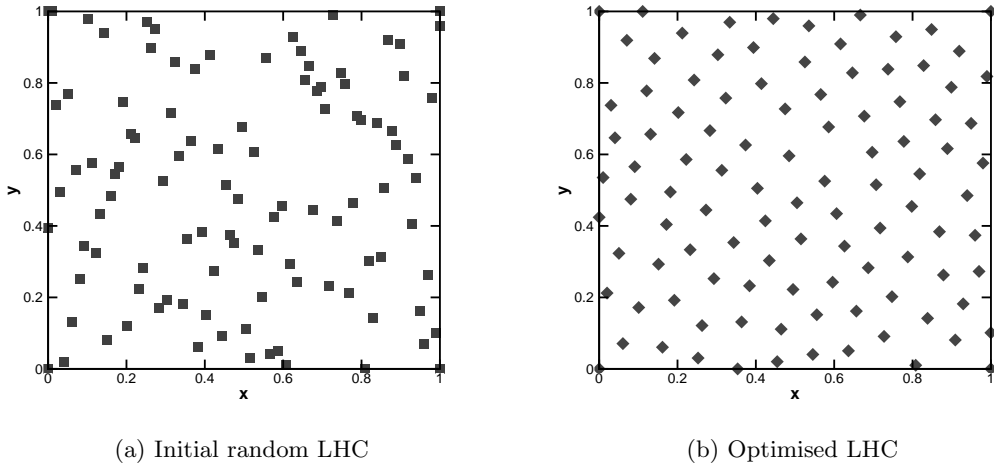


Figure 5: Latin Hypercube sampled design space (plus corner points)

The Latin Hypercube used in this work consisted of 181 sample and corner points.

IV.C. Results

At all Latin Hypercube samples, a complete polar was run and interpolation used to obtain the angle, and therefore drag, where the lift was the design lift coefficient ($C_L = 1.3$). A brief statistical summary of the results is shown in tables 2 and 3.

From table 2, the standard deviation relative to the mean is significant for both the separation (S) and drag (C_D), although the objective function J has a marginally smaller standard deviation amongst the points sampled. It is evident from table 3 that considering drag in isolation is not sufficient to reduce (and eliminate) separation. Of the 181 samples, 24.5% have $S < 0.1\%$, but only two (4.4%) of this reduced sample have a $C_D < 0.0210$. These two samples are spread throughout the design space, such that the distance

Table 2: Design space results

Parameter	Min	Max	Mean	St. Dev
C_D	0.0174	0.0784	0.0259	0.0098
S (%)	0.0	17.39	5.78	5.22
J	0.0204	0.0784	0.0288	0.0090

Table 3: Design space results

	Parameter	Value	h	AR	θ	c	s
Min. J	C_D	0.0203					
	Separation	0.08%	0.82%	3.84	16.89°	55.6%	8.8
	J	0.0204					
Max. J	C_D	0.0784					
	Separation	0.00%	1.98%	2.98	27.00°	15.73.0%	10.17
	J	0.0784					
Min. C_D	C_D	0.0178					
	Separation	10.39%	1.72%	2.58	22.75°	37.45%	18.05
	J	0.0230					
Max. C_D	C_D	0.0784					
	Separation	0.00%	1.98%	2.98	27.00°	15.73%	10.17
	J	0.0784					
Max. S	C_D	0.0263					
	Separation	17.39%	1.89%	1.87	13.13°	59.45%	13.16
	J	0.0349					

between these points with the normalised $[0, 1]^5$ design space is 0.85 which equates to 38% of the design space range.

V. VG Design Space Modelling

A full design space investigation of a number of VG design parameters has been considered above using high-fidelity viscous analysis. A Latin Hypercube sampled design space of the VG geometry and location on the aerofoil has been performed. The next stage of this work is the design optimization of the VG based on the design parameters that have been identified as being of significance. This is achieved by representing the remainder of the design space (i.e. away from the already simulated design points) by an interpolation-based surrogate model that is built around radial basis functions. This interpolation recovers the original data and therefore preserves the high-fidelity nature of the solutions while simultaneously distributing that information throughout the design space.

V.A. Surrogate Modelling

A surrogate model is an interpolation or approximation made from a small sample of data sites that give the value of the design space at those know data sites. The model can be formulated in a number of different ways, though the most common methods are by a polynomial, radial basis functions (RBFs) or Kriging. A brief outline of the three methods is presented below, but reviews of surrogate modelling and surrogate-based optimization have been presented,^{45–47} which the reader is guided to for more in-depth discussions on the formulations and common uses for each approach.

The polynomial method⁴⁸ involves fitting an order m polynomial through the given data, often by a least squares fit. The resulting model is not guaranteed to recover any of the existing data, meaning it tends to be more suited to fitting noisy data, where an underlying trend is the desired outcome instead of an exact

model. An RBF model is a linear combination of a number of functions that depend on the radial distance from known data sites.^{49,50} Unlike polynomial interpolation, RBF interpolation does provide exact recovery at the known data sites, though interpolation of noisy data using RBFs can also be done using a regulation parameter.⁵¹ Finally, the Kriging approach, pioneered by Krige⁵² and later used for surrogate modelling by Sacks *et al.*⁵³ and Jones *et al.*,⁵⁴ is a special form of an RBF interpolation that uses a generalised Gaussian equation as the basis function. The function allows variation in the response from variables to variable, and is more statistically-based than a pure Gaussian function.

The surrogate model used in this work is developed around a multivariate interpolation using RBFs. This interpolation method has the advantage of providing exact recovery of data at the known sites, thus preserving the CFD data at the Latin Hypercube sampled points.

The general theory of RBFs is presented by Buhmann⁵⁵ and Wendland,⁵⁶ and the basis of the method used here is described in detail by Rendall and Allen⁴⁰ and surrogate specific detail in Mackman *et al.*¹¹ Let $f(\mathbf{x})$ be the original function to be modelled, and f_i be the scalar values at n discrete points $\mathbf{x}_i, i = 1, \dots, n$, where \mathbf{x}_i is the vector of inputs at the i th sample point in d -dimensional space $x^j, j = 1, \dots, d$. The set of data points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is confined to a domain Ω in d -dimensional space. A RBF model is then a linear combination of basis functions, whose argument is the Euclidean distance between the point \mathbf{x} at which the interpolation is made and the n points in the known data set. In other words, the interpolation at an untried site is a sum of contributions from all the known function values, the influence of which is controlled by a basis function that depends on the distance they are from the new site. If ϕ is the chosen basis function and $\|\cdot\|$ is used to denote the Euclidean norm, then an interpolation model s has the form

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) \quad (3)$$

where $\alpha_i, i = 1, \dots, n$ are model coefficients, and $p(\mathbf{x})$ is an optional polynomial. The coefficients are found by requiring exact recovery of the original data, $s_{\mathcal{X}} = \mathbf{f}$, for all points in the training data set \mathcal{X} . Hence the model is an interpolant, and all original solution information is preserved. When the polynomial term is included, the system is completed by the additional requirement

$$\sum_{i=1}^n \alpha_i p(\mathbf{x}) = 0 \quad (4)$$

which is sometimes referred to as the side condition, for a polynomial that takes the form

$$p = \gamma_0 + \sum_{j=1}^d \gamma_j x^j \quad (5)$$

Setting up a global RBF interpolation then requires a solution to a linear system:

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{M} \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_d \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \quad (6)$$

where

$$P = \begin{pmatrix} 1 & \cdots & 1 \\ x_1^1 & \cdots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^d & \cdots & x_n^d \end{pmatrix} \quad M = \begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,n} \\ \vdots & \ddots & \vdots \\ \phi_{n,1} & \cdots & \phi_{n,n} \end{pmatrix} \quad \phi_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$$

Once the coefficients are known, the objective function at an interpolated point is then found by equation 3.

Many choices are available for the basis function itself, but a radial conditionally positive definite function is chosen to ensure the system has a unique solution. Basis functions are said to be compactly supported when they decay to zero at a given distance from the centre, known as the support radius R . Compactly supported functions have desirable numerical characteristics and modelling behaviour, hence the work presented here concentrates on Wendland's functions⁵⁶ $\phi_{d,k}$, which are compact functions of minimal degree for a stated number of continuous derivatives C^{2k} in d dimensions. Wendland's C^2 ($\phi_{d,1}$) in up to three dimensions is given by

$$\phi_{3,1} = (1 - r)_+^4 (4r + 1), \quad (7)$$

where $r = \|\cdot\|$ denotes the norm and $(\cdot)_+$ refers to the cut-off function

$$(1 - r)_+ = \begin{cases} 1 - r, & 1 - r \geq 0, \\ 0, & 1 - r < 0. \end{cases} \quad (8)$$

The support radius used in the function also acts to control the region of influence of each of the centres. A larger support radius allows each sample point to influence the interpolation at a given point from a greater distance away, and in general leads to a smoother interpolation. For a compact function, it defines a hypersphere around each sample point outside of which the value of ϕ is zero, and inside of which the point has some influence. The support radius scales the basis function argument as follows:

$$r_{scaled} = \frac{r}{R}, \quad (9)$$

hence the distance between two points is then the scaled Euclidean distance

$$\|\mathbf{x} - \mathbf{x}_i\|_{scaled} = \frac{1}{R} \sqrt{\sum_{j=1}^d (x_j - x_{j,i})^2}. \quad (10)$$

V.B. VG Surrogate Model

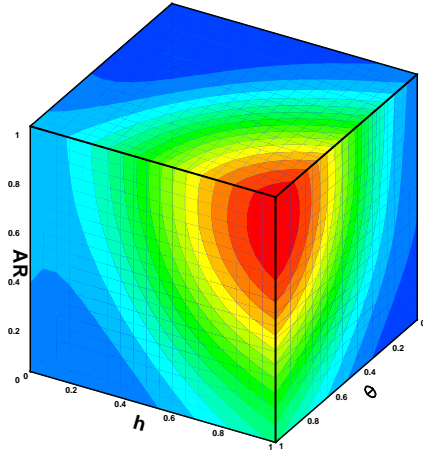
The surrogate model is constructed using Wendland's C2 function with a support radius sufficient to cover the entire design space. Selected projections for the objective function J are presented in figure 6 for the design space $[0, 1]^5$.

VI. VG Design Optimization

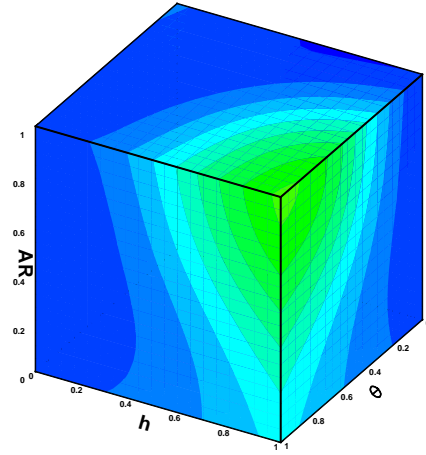
The use of a surrogate model allows very quick evaluation of an interpolation of an otherwise expensive, high-fidelity design space. The high fidelity nature of the data used to produce the interpolation is, however, preserved by building a surrogate model using RBFs. The next stage of this work is the optimization of the VG design based on the five parameters outlined as being important to VG design.

One option for the VG design optimization is to use the simulation approach developed in this work within an optimization framework, which is known as full-fidelity aerodynamic shape optimization. This approach, however, is expensive as very large numbers of simulation evaluations are required during the process. A less expensive approach, and one that exploits the use of the surrogate model, is surrogate-based optimization and is the approach used in this work. This optimization approach substitutes a full simulation evaluation to obtain an objective function value for a given set of inputs with a surrogate model evaluation to obtain an interpolated approximation to the true value of the objective function for the given inputs. The surrogate evaluation is simply given by equation 3, and represents a very cheap way of obtaining responses to the output for the given inputs.

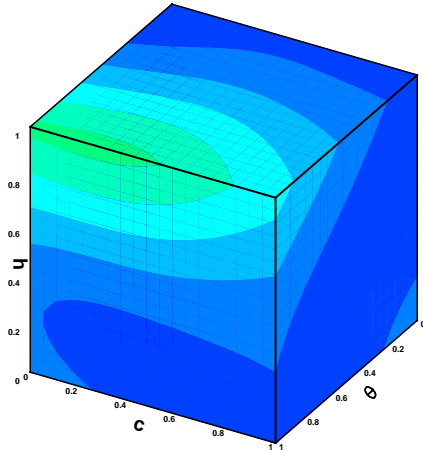
To interrogate the surrogate design space, an optimization algorithm is required. One choice of optimization algorithm is a gradient-based method, however, these are prone to termination in a local minimum that is not necessarily the global minimum solution. Gradient-based approaches are well suited to optimization problems with expensive objective function evaluations, such as full-fidelity aerodynamic shape optimization, for example.^{57–59} When multimodality (multiple local minima) is an issue, a global optimization algorithm is often more effective at locating the globally optimal solution. These algorithms tend to be agent-based, such as particle swarm⁶⁰ or gravitational search,⁶¹ or can also be evolutionary-based, such as differential



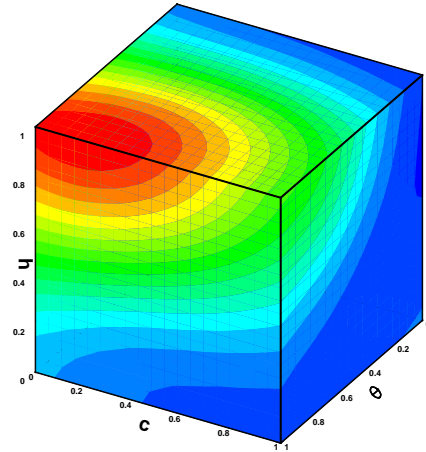
(a) Projection at $c = 0, s = 0$



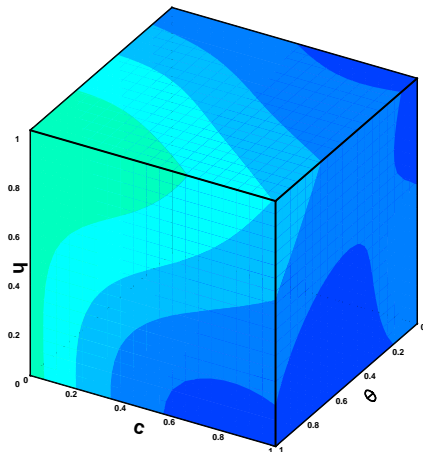
(b) Projection at $c = 1, s = 0$



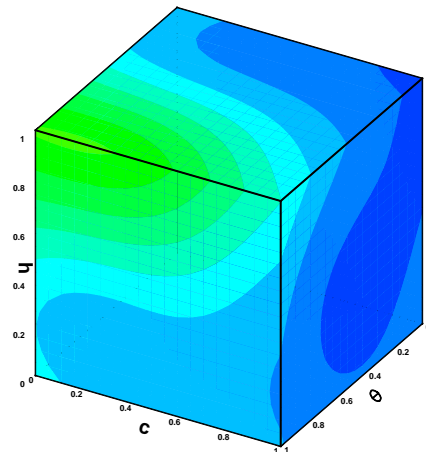
(c) Projection at $AR = 0, s = 0$



(d) Projection at $AR = 1, s = 0$



(e) Projection at $AR = 0, s = 1$



(f) Projection at $AR = 1, s = 1$

Figure 6: Surrogate model projections (blue represents lower objective, J)

evolution.⁶² These global methods usually require many more objective function evaluations compared to gradient-based methods and are therefore suited to problems that have cheap objective function evaluations.

An objective function evaluation in this work is represented by a surrogate evaluation, which is very cheap to evaluate. Furthermore, due to the unknown nature of the modality in the design space, a global optimization algorithm has been developed for optimizing the VG surrogate design space. The global algorithm that has been developed is based on the differential evolution algorithm presented by Storn and Price,⁶² and is outlined in section VI.A.

In addition to searching the surrogate design space for an optimum solution directly, a complementary approach interrogates the surrogate model in order to both refine regions of interest and ensure the accuracy of the surrogate design space. Regions of interest in this instance being local minima within the design space. An efficient method of improving surrogate-based accuracy and refinement is adaptive sampling. Adaptive sampling relies on reconstructing/updating the surrogate model across a number of iterative cycles by updating the surrogate model with newly determined adaptive sample points. A kriging based adaptive strategy was compared to a similar RBF adaptive strategy in Mackman *et al.*¹¹ These utilise the primarily space filling MaxMSE (largest value of error predicted by the mean squared error function, otherwise referred to as entropy)⁶³ and a statistical optimization function (expected improvement function). Expected improvement balances local and global search criteria.⁵⁴ An alternative RBF adaptive approach that incorporates both local gradient-based refinement and space-filling properties has also been developed.¹² This approach is presented in section VI.B.

VI.A. Differential Evolution

Differential evolution (DE) is a swarm intelligence algorithm built around the concept of evolutionary mechanics. Like the other swarm intelligence algorithms DE takes a number of agents (parameter vectors) which are often termed chromosomes, and follows three steps to advance the optimization algorithm:

1. *Mutation*: generate a mutated candidate solution by small, random changes
2. *Crossover*: ‘breeding’ of mutated and parent solutions to produce a child
3. *Selection*: accept the new child based on its fitness

The parameter vectors are vectors of length d , where each entry represents a position in each of the d design parameters:

$$\mathbf{x}_n = \{x_n^1, x_n^2, \dots, x_n^d\}^T \quad (11)$$

The mutation stage involves the production of a new, candidate solution to introduce variability and exploration into the algorithm. The candidate/mutated solution for the n -th parent/target solution is called the donor vector, \mathbf{v}_n , and is produced by combining three existing parameter vectors:

$$\mathbf{v}_n = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (12)$$

where r_1 , r_2 and r_3 ($r_1 \neq r_2 \neq r_3 \neq n$) are random integers chosen from the range $[1, N]$, for a population of N parameter vectors (or agents), for each parent vector, and F is a scalar that is typically in the interval $[0, 1]$ and is chosen to either maximize or minimize exploration away from the existing agents.

The crossover stage is used to enhance diversity in the population by combining aspects of the given parent and donor vectors. For the d -th entry in the parent (x_n^d) and donor vectors (v_n^d), the d -th entry in a trial/child vector (u_n^d) is produced where each entry is from the donor vector based on a probability CR , otherwise it is from the parent vector. This ensures there is guaranteed variability and that \mathbf{u}_n gets at least one entry from \mathbf{v}_n , a random entry from \mathbf{v}_n is always introduced into \mathbf{u}_n .

The final stage of the algorithm for each of the parameter vectors is to assess whether the new child is of sufficient fitness to replace the given parameter vector. If the fitness of the child vector is better than its parent then the child replaces the parent.

Multiple further additions have been presented for DE, and in the review by Neri and Tirronen,⁶⁴ many of these were tested for cost and performance comparison. A small addition that is simple to make to DE to considerably improve its performance is population size reduction (PSR),⁶⁵ which splits the overall number of iterations of the optimizer into a number of stages where the number of parameter vectors are reduced by

a half at each stage. Only the parameter vectors with the best fitness are kept after each stage which has the effect of managing the loading between exploration with many vectors at the beginning of the optimization, and exploitation towards the end of the optimization.

The DE algorithm was run with constants commonly found in the literature; 50 particles, crossover probability 0.3, mutation rate 0.7, 300,000 function evaluations and 4 stages for the PSR.

VI.B. Adaptive Sampling

An adaptive sampling method requires an initial set of samples to first build an interpolation, and then updated sampling points are added based on a compromise between space-filling updates and local refinement in non-linear regions. This non-linearity is determined by the Laplacian. A smooth separation function quantifies the sample spacing in order to assess space-filling updates. A criterion, C , was proposed in¹² which merges these two criteria:

$$C = (|\nabla^2 J| + \epsilon)(1 - h)^2 \quad 0 \leq h \leq 1 \quad (13)$$

where ϵ is a small number, sufficient to avoid $C = 0$ when $|\nabla^2 J| = 0$. The variable h is the space-filling function, such that $1 - h$ grows away from previously sampled locations. However, this criterion C was proposed in order to improve information around both maxima and minima within the surrogate model (hence the magnitude of the Laplacian). If only a minimum is desired (positive Laplacian) or maximum (negative Laplacian) then the C function is adjusted appropriately. Larger values of C indicate new potential sample locations. Either a grid evaluation can be conducted to determine maximum C , and hence the next sample location, or a more robust and efficient strategy utilising the global optimization algorithm outlined in the previous subsection is undertaken.

The space-filling (or separation) function, h , is defined as a smooth function, for blending with the local refinement component of C . In this function, $h = 1$ at a sample point, and diminishes as distance increases away from the sample point. This is produced using an RBF kernel with a compactly supported basis function and small support radius. The support radius used in this work was determined as a function of two distances, the minimum distance (d_1) and supremum minimum distance (d_2) between sample points.

$$Sr_h = R_h \sqrt{(d_1 d_2)} \quad (14)$$

where R_h is a scaling factor, which in this work was $R_h = 1$. However, it was found that the adaptive sampling approach outlined above was insufficient to progressively refine the solution without significant adaptive sample points. Therefore the adaptive sampling approach was modified with the addition of an exploitative term

$$C_b = (|\nabla^2 J| + \epsilon)(1 - h)^2 b^3 \quad (15)$$

where b is a distance based function and $0 \leq b \leq 1$. The b function is also smooth and produced via an RBF kernel in the same manner as h . Instead of using all sampled points, the function merely utilises a subset of p best fitness values taken from the full known sample set, where p is the number of samples in the subset. The support radius for the b function is the maximum distance between a subset sampled point to the full sample set. The b function will have a value of $b = 1$ at any subset sample point and will decay away from the subset of p points to zero at the furthest reaches of the domain (relative to the subset points). Thus the original adaptive sampling approach can be seen as primarily explorative (when $b = 1$ throughout the design space), whilst the latter is exploitative ($0 \leq b \leq 1$). This can be demonstrated using two analytical functions in the following subsections.

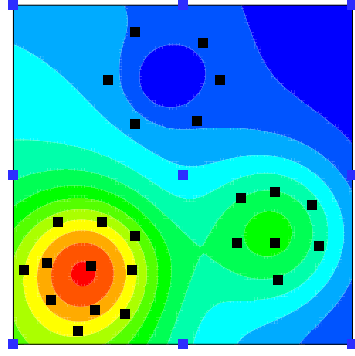
VI.B.1. Franke's Function

Franke's function is a bivariate function, consisting of two Gaussian peaks of differing heights and a smaller trough. In this instance, the adaptive sampling is undertaken using a grid evaluation seeking only the maxima location, with the trough refinement ignored. The function is defined using

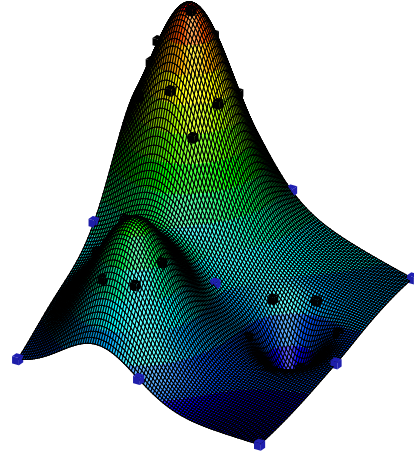
$$\begin{aligned}
f(\mathbf{x}) = & 0.75 \exp \left(-\frac{(9x_1 - 2)^2}{4} - \frac{(9x_2 - 2)^2}{4} \right) + 0.75 \exp \left(-\frac{(9x_1 + 1)^2}{49} - \frac{9x_2 + 1}{10} \right) \\
& + 0.5 \exp \left(-\frac{(9x_1 - 7)^2}{4} - \frac{(9x_2 - 3)^2}{4} \right) - 0.2 \exp \left(-(9x_1 - 4)^2 - (9x_2 - 7)^2 \right)
\end{aligned} \tag{16}$$

between $x_i \in [0, 1]$.

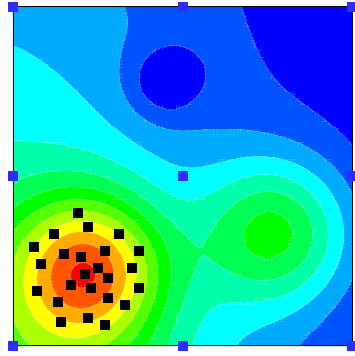
After an initial 9 sample points, 24 adaptive sample points are introduced (sequentially, rather than in batches). The results for using both adaptive sampling techniques is given in figure 7.



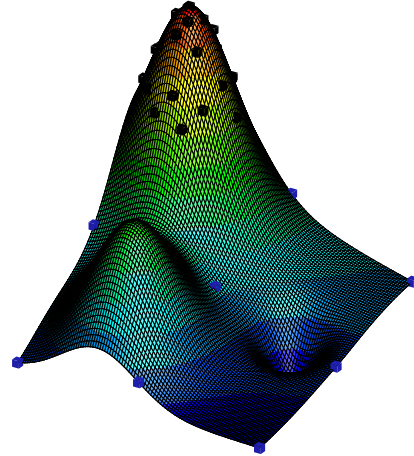
(a) C Approach (2d)



(b) C Approach (3d projection)



(c) C_b Approach (2d)



(d) C_b Approach (3d projection)

Figure 7: Comparison of C and C_b adaptive sampling approaches using Franke's Function

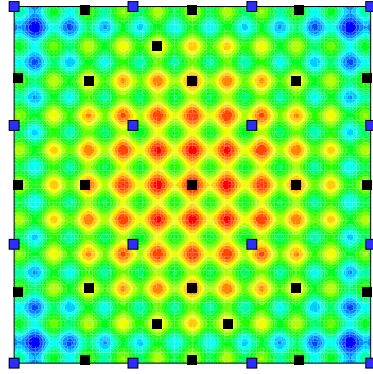
From the figure, it is clear that the C_b approach is highly concentrated around the primary peak. Less obviously, it is also more rapid in identifying the exact peak location during the adaptive cycle.

VI.B.2. Rastrigin Function

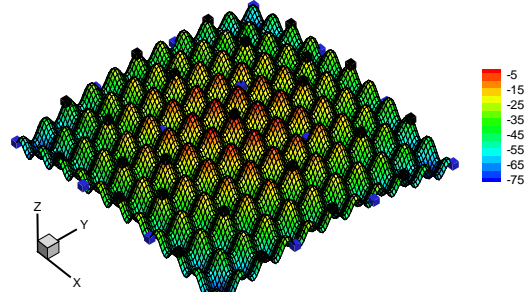
The two-dimensional variant of the n -dimensional Rastrigin function is a more complex problem, with a plethora of local minima and maxima. It is defined by

$$f(\mathbf{x}) = -An - \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i)) \tag{17}$$

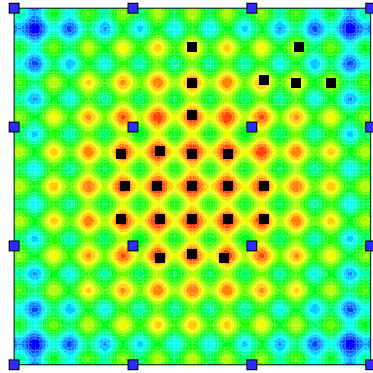
where $A = 10$, $n = 2$, $x_i \in [-5.12, 5.12]$. The global maximum ($f(\mathbf{x}) = 0$) is located at $\mathbf{x} = 0$. After an initial 16 sample points, 24 adaptive sample points are introduced sequentially. As shown in figure 8, both adaptive approaches capture the overall global maximum, although the C_b function has again clearly covered all potential local maxima within proximity.



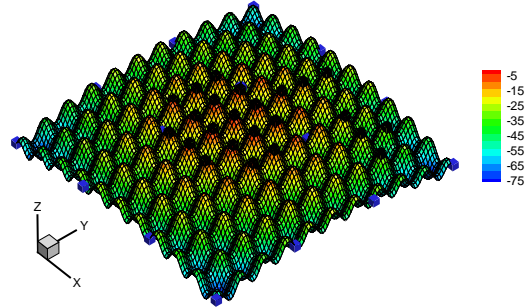
(a) C Approach (2d)



(b) C Approach (3d projection)



(c) C_b Approach (2d)



(d) C_b Approach (3d projection)

Figure 8: Comparison of C and C_b adaptive sampling approaches using Rastrigin's Function

Both adaptive strategies offer advantages, with C providing an initial adaptive coverage, and C_b refining the final solution.

VI.C. VG Optimization Results

The results of optimizing the VG surrogate model for the objective function given by equation 1 are given in table 4, along with the clean values and the values that are the lowest objective values from the latin hypercube sample. To increase the adaptive sampling rate, a number of updates per iteration were undertaken in parallel. This only requires updating the separation function before determination of the next update location. The number of updates per iteration and the number of iterations was prescribed in advance. Two rounds of explorative adaptive sampling using five samples in each round were followed by one round of exploitative sampling.

The results clearly demonstrate that the optimized value is a considerable improvement in the separation experienced on the aerofoil and therefore a reduction in buffet may be expected. The optimization process

via differential evolution has also improved the performance of the VG compared to just taking the best sample tested in the latin hypercube, therefore vindicating the optimization process. Comparing to the clean geometry, it is particularly impressive that the process developed here, of design space sampling, surrogate-based optimization and adaptive sampling, has been able to eliminate separation at a highly loaded condition and achieved a decrease in the drag on the aerofoil of 7% relative to best result achieved by the Latin hypercube sampling alone.

Table 4: Design space results

	J	C_D	Separation	h	AR	θ	c	s
<i>Clean</i>	0.0246	0.0190*	11.1%*	-	-	-	-	-
<i>Min. LHC</i>	0.0204	0.0203	0.08%	0.81%	3.84	16.89°	55.6%	8.8
<i>Optimized (Surrogate)</i>	0.0189	-	-	0.85%	3.64	8.42°	21.5%	5.8
<i>Optimized (CFD)</i>	0.0189	0.0189	0.0%					

* oscillatory result: time-averaged

Streamlines proximal to the VG and trailing edge of the airfoil are given in figure 9. From the figure, the large separation region present in the clean geometry has been entirely eliminated using the optimum VG configuration; see figure 3a for the clean geometry streamlines. Figure 9c shows the vortex extent, as defined by the $\lambda_2 = 0$ criterion between the minimum LHC sample and the optimum solution. This criterion denotes the intermediate eigenvalue of the symmetric tensor $\mathbf{S}^2 + \mathbf{\Omega}^2$ where $\mathbf{\Omega} = \frac{1}{2}[\nabla \mathbf{u} - (\nabla \mathbf{u})^T]$ is the vorticity tensor and $\mathbf{S} = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$ the rate of strain tensor. Negative values of the λ_2 criterion denote the vortex. The induced vortex of the optimum design has eliminated separation at the trailing edge of the airfoil, as shown in figure 9b, although a small region is still present in the minimum LHC sample.

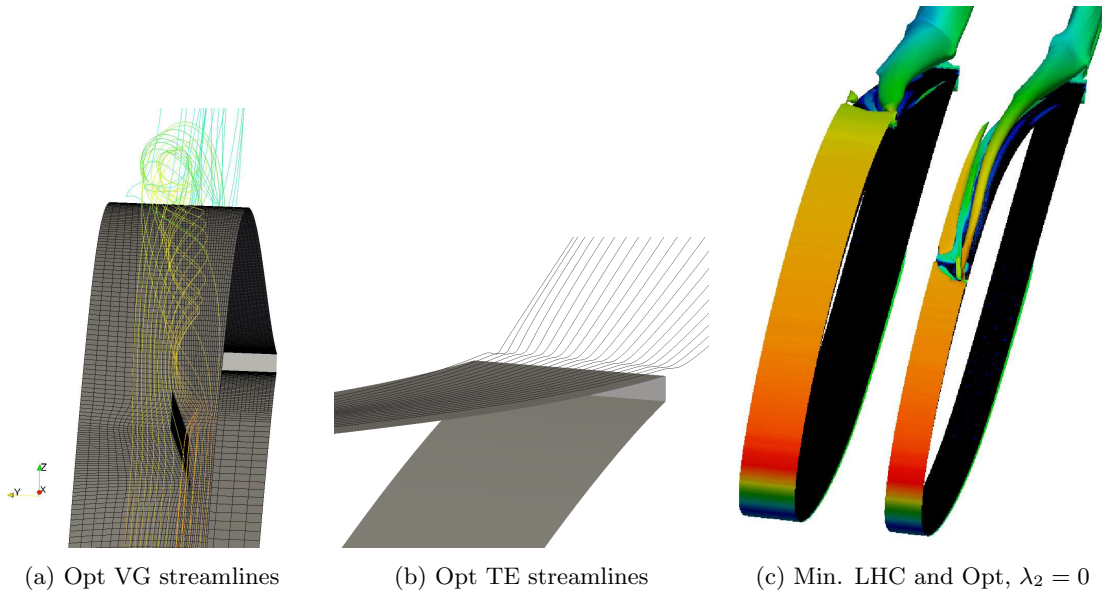


Figure 9: Induced vortex impact downstream of VG

The high performance of the optimization algorithm can be demonstrated by considering the value of the gradients of the objective with respect to changes in the design variables at the optimized point. An optimum solution for this type of problem is one where the gradient values approach a very small number. Table 5 shows these gradient values and demonstrates the high performance of the optimization algorithm, producing a solution that is optimum to within a very small tolerance, and an improvement on the minimum sample from the Latin Hypercube, which is not at an optimum solution as defined by the gradient.

Table 5: Gradients at minimum LHC sample and optimum solution

	Min. LHC	Optimized
$\partial J/\partial h$	6.2×10^{-3}	-2.3×10^{-10}
$\partial J/\partial AR$	-1.3×10^{-2}	1.2×10^{-9}
$\partial J/\partial \theta$	3.0×10^{-3}	-3.8×10^{-11}
$\partial J/\partial c$	3.2×10^{-3}	7.1×10^{-10}
$\partial J/\partial s$	-4.2×10^{-3}	1.5×10^{-9}

VII. Conclusions

The quantification of the effects of flow around a tiltrotor-like aerofoil with vortex generators (VGs) and subsequent design of these VGs has been considered. The highly loaded and thick wing of a tiltrotor is particularly susceptible to early onset buffet, so boundary layer control is critical and, hence, upper surface flow control devices are an important consideration. An analysis package has been presented that uses the open source library, OpenFOAM, to obtain high-fidelity viscous flow simulation, and a custom mesh generator to autonomously obtain high quality structured meshes.

Design of counter rotating VGs has been considered using, firstly, a Latin Hypercube sampling approach to obtain design space samples for five variables; VG height, VG aspect ratio, VG angle, chord-wise location, and inter- and intra-pair spacing ratio. A surrogate model that uses radial basis functions has been developed to produce an approximation to a high-fidelity design space but still preserves the important flow physics required for design. An optimization algorithm based on differential evolution, which is a global search algorithm, was used to fully interrogate the surrogate model design space. With the initial sample points defined via an optimised Latin Hypercube, adaptive sampling using a combined objective function Laplacian/space-filling criterion was then employed in order to locate a global optimum VG design for the high loading manoeuvring condition. The objective function blends the requirement for drag minimization with reduction of separation.

The best Latin Hypercube result obtained reduced the separation from 11.1% to 0.08%, but increased the drag by 6.8% relative to the clean geometry. Following the surrogate-based design optimization, the optimum result obtained has eliminated separation entirely at $C_L = 1.3$ whilst decreasing the drag by 7% relative to the minimum Latin Hypercube sample and with a marginal decrease in drag relative to the clean geometry.

Acknowledgements

The authors would like to thank David Tring and Nigel Scrase of AgustaWestland, for their support of and contribution to this work. The financial support, via the HiPerTilt project, of AgustaWestland and Innovate UK (Technology Strategy Board) is also gratefully acknowledged. Finally, the authors would like to acknowledge that this work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol.

References

- ¹M.D. Maisel, D.J. Giulianetti, and D.C. Dugan. The history of the XV-15 tilt rotor research aircraft: From concept to flight. Technical Report SP 2000-4517, NASA, 2000.
- ²M.D. Weiberg, J.A. Maisel. Wind-tunnel tests of the XV-15 tilt rotor aircraft. Technical Report TM 81177, NASA.
- ³J. Abras and R. Narducci. Analysis of cfd modeling techniques over the mv-22 tiltrotor. In *66th American Helicopter Society International Annual Forum*, Phoenix, Arizona, 2010.
- ⁴C-S. Yao, J. C. Lin, and B. G. Allan. Flow-field measurements of device-induced embedded streamwise vortex on a flat plate. In *1st AIAA Flow Control Conference*, St. Louis, Missouri, 2002. AIAA Paper 2002-3162.
- ⁵T.C. Tai. Effect of midwing vortex generators on v-22 aircraft forward-flight aerodynamics. *Journal of Aircraft*, 40(4):623–630, 2003.
- ⁶B. Konig, E. Fares, and S. Nolting. Fully-resolved lattice-boltzmann simulation of vane-type vortex generators. In *7th AIAA Flow Control Conference*, Atlanta, Georgia, 2014. AIAA Paper 2014-2795.
- ⁷A. Jirasek. Design of vortex generator flow control in inlets. *Journal of Aircraft*, 43(6):1886–1892, 2006.

- ⁸Y. Kuya, K. Takeda, X. Zhang, and A. I. J. Forrester. Multifidelity surrogate modeling of experimental and computational aerodynamic data sets. *AIAA Journal*, 49(2):289–298, 2011.
- ⁹N. Namura, S. Obayashi, and S. Jeong. Efficient global optimization of vortex generators on a super critical infinite-wing using kriging-based surrogate models. In *52nd Aerospace Sciences Meeting and Exhibit*, National Harbor, Maryland, 2014. AIAA Paper 2014–904.
- ¹⁰JS. Yi, C. Kim, and B. J. Lee. Adjoint-based design optimization of vortex generator in an S-shaped subsonic inlet. *AIAA Journal*, 50(11):2492–2507, 2012.
- ¹¹T. J. Mackman, C. B. Allen, M. Ghoreyshi, and K. J. Badcock. Comparison of adaptive sampling methods for generation of surrogate aerodynamic models. *AIAA Journal*, 51(4):797–808, 2013.
- ¹²T. J. Mackman and C. B. Allen. Investigation of an adaptive sampling method for data interpolation using radial basis functions. *International Journal for Numerical Methods in Engineering*, 83:915–938, 2010.
- ¹³R. L. T. Bevan, D. J. Poole, C. B. Allen, and T. C. S. Rendall. Simulation and surrogate-based design of rectangular vortex generators for tiltrotor aircraft wings. In *33rd AIAA Applied Aerodynamics Conference*, Dallas, Texas, 2014.
- ¹⁴J. W. Hamstra, D. N. Miller, and P. P. Truax. Active inlet flow control technology demonstration. In *22nd International Congress of Aeronautical Sciences*, pages Harrogate, UK, 2000. ICAS-2000-6.11.2.
- ¹⁵B. G. Allan, C-S. Yao, and J. C. Lin. Numerical simulations of vortex generator vanes and jets on a flat plate. In *1st AIAA Flow Control Conference*, St Louis, Missouri, 2002. AIAA Paper 2002–3160.
- ¹⁶E. Wik and S. T. Shaw. Numerical simulation of micro vortex generators. In *2nd AIAA Flow Control Conference*, Portland, Oregon, 2004. AIAA Paper 2004–2697.
- ¹⁷J. Dandois, V. Brunet, P. Molton, J.-C. Abart, and A. Lepage. Buffet control by means of mechanical and fluidic vortex generators. In *5th AIAA Flow Control Conference*, Chicago, Illinois, 2010. AIAA Paper 2010–4975.
- ¹⁸F. K. Lu, A. J. Pierce, Y. Shih, C. Lin, and Q. Li. Experimental and numerical study of flow topology past micro vortex generators. In *40th AIAA Fluid Dynamics Conference and Exhibit*, Chicago, Illinois, 2010. AIAA Paper 2010–4463.
- ¹⁹Q. Li and C. Liu. Declining angle effects of the trailing edge of a microramp vortex generator. *Journal of Aircraft*, 47(6):2086–2095, 2010.
- ²⁰C. Memory, D. O. Snyder, and J. Bons. Numerical simulation of vortex generating jets in zero and adverse pressure gradients. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2008. AIAA Paper 2008–558.
- ²¹Y. Ito, M. Murayama, and K. Yamamoto. High-quality unstructured hybrid mesh generation for capturing effects of vortex generators. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, Texas, 2013. AIAA Paper 2013–554.
- ²²K. J. Forster and T. R. White. Numerical investigation into vortex generators on heavily cambered wings. *AIAA Journal*, 52(5):1059–1071, 2014.
- ²³M. Kerho and B. Kramer. Enhanced airfoil design incorporating boundary layer mixing devices. In *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2003. AIAA Paper 2003–211.
- ²⁴M. Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In T. J. Mueller, editor, *Low Reynolds Number Aerodynamics*, pages 1–12. Springer Berlin Heidelberg, 1989.
- ²⁵M. Drela and M.B. Giles. Viscous-inviscid analysis of transonic and low reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.
- ²⁶P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *Recherche Aerospaciale*, 1:5–21, 1994.
- ²⁷F. R. Menter. Two-equation eddy-viscosity turbulence model for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994.
- ²⁸J. Dacles-Mariani, G. G. Zilliac, J. S. Chow, and P. Bradshaw. Numerical/experimental study of a wingtip vortex in the near field. *AIAA Journal*, 33(9):1561–1568, 1995.
- ²⁹M. I. Yaras and A. D. Grosvenor. Evaluation of one- and two-equation low-Re turbulence models. part II–vortex-generator jet and diffusing s-duct flows. *International Journal for Numerical Methods in Fluids*, 42(12):1321–1343, 2003.
- ³⁰W. G. Kunik. Application of a computational model for vortex generators in subsonic internal flows. In *AIAA/ASME/SAE/ASEE 22nd Joint Propulsion Conference*, Huntsville, Alabama, 1986. AIAA Paper 1986–1458.
- ³¹N. E. May. A new vortex generator model for use in complex configuration CFD solvers. In *19th AIAA Applied Aerodynamic Conference*, Anaheim, California, 2001. AIAA Paper 2001–2434.
- ³²E. E. Bender, B. H. Anderson, and P. J. Yagle. Vortex generator modeling for Navier-Stokes codes. In *American Society of Mechanical Engineers*, 1999. FEDSM 99-6929.
- ³³R. V. Chima. Computational modeling of vortex generators for turbomachinery. In *ASME TURBO EXPO*, Amsterdam, Netherlands, 2002. GT-2002-30677.
- ³⁴A. Jirasek. Vortex-generator model and its application to flow control. *Journal of Aircraft*, 42(6):1486–1491, 2005.
- ³⁵S. Lee, E. Loth, and H. Babinsky. Normal shock boundary layer control with various vortex generator geometries. *Computers and Fluids*, 49(1):233–246, 2011.
- ³⁶H. Shan, L. Jiang, C. Liu, M. Love, and B. Maines. Numerical study of passive and active flow separation control over a naca0012 airfoil. *Computers and Fluids*, 37(8):975–992, 2008.
- ³⁷H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, and V. Yakhot. Extended boltzmann kinetic equation for turbulent flows. *Science*, 301(5633):633–636, 2003.
- ³⁸C. B. Allen. CHIMERA volume grid generation within the EROS code. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 214(3):125–141, 2000.
- ³⁹C. B. Allen. Towards automatic structured multiblock mesh generation using improved transfinite interpolation. *International Journal for Numerical Methods in Engineering*, 74(5):697–733, 2008.

- ⁴⁰T. C. S. Rendall and C. B. Allen. Unified fluid–structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559, 2008.
- ⁴¹T. C. S. Rendall and C. B. Allen. Efficient mesh motion using radial basis functions with data reduction algorithms. *Journal of Computational Physics*, 228(17):6231–6249, 2009.
- ⁴²M. D. McKay, R. J. Beckman, 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*, 21(2):239–245, 1979.
- ⁴³M. D. Morris and T. J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43:381–402, 1995.
- ⁴⁴R. Jin, W. Chen, and A. Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134:268–287, 2005.
- ⁴⁵D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
- ⁴⁶N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41:1–28, 2005.
- ⁴⁷A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45:50–79, 2009.
- ⁴⁸G. E. P. Box and N. R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, 1987.
- ⁴⁹D. S. Broomhead and D. Lowe. Multivariable function interpolation and adaptive networks. *Complex Systems*, 2(3):321–355, 1988.
- ⁵⁰A. Sobester, S. J. Leary, and A. J. Keane. On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, 33(1):31–59, 2005.
- ⁵¹T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247(4945):978–982, 1990.
- ⁵²D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 53(5):159–162, 1952.
- ⁵³J. Sacks, S. B. Schiller, and W. J. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.
- ⁵⁴D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- ⁵⁵M. Buhmann. *Radial Basis Functions*. Cambridge University Press, 1st edition, 2005.
- ⁵⁶H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 1st edition, 2005.
- ⁵⁷J. E. Hicken and D. W. Zingg. Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA Journal*, 48(2):400–413, 2010.
- ⁵⁸F. Bisson, S. K. Nadarajah, and D. Shi-Dong. Adjoint-based aerodynamic optimization framework. In *52nd AIAA Aerospace Sciences Meeting*, National Harbor, Maryland, 2014. AIAA Paper 2014–0412.
- ⁵⁹G. K. W. Kenway and J. R. R. A. Martins. Multi-point high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, 2014.
- ⁶⁰J. Kennedy and R. Eberhart. Particle swarm optimization. In *1995 IEEE International Conference on Neural Networks*, Perth, Australia, 1995.
- ⁶¹E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi. GSA: A gravitational search algorithm. *Information Sciences*, 179:2232–2248, 2009.
- ⁶²R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, ICSI, UC Berkeley, 1995. TR-95-012.
- ⁶³M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.
- ⁶⁴F. Neri and V. Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33:61–106, 2010.
- ⁶⁵J. Brest and M. S. Maucec. Population size reduction for the differential evolution algorithm. *Applied Intelligence*, 29(3):228–247, 2008.