FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
Computational Engineering and Design

Design-in-Context: Exploiting the Increasing Accessibility of Global FE Models to Improve the Mechanical Design Process of Engineering Subsystems

by

Hau Kit Yong

Thesis for the degree of Doctor of Philosophy

May 2021

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
Aeronautical and Astronautical Department

Doctor of Philosophy

DESIGN-IN-CONTEXT: EXPLOITING THE INCREASING ACCESSIBILITY OF GLOBAL FE MODELS TO IMPROVE THE MECHANICAL DESIGN PROCESS OF ENGINEERING SUBSYSTEMS

by Hau Kit Yong

The established component structural design process is based on evaluations of high fidelity FE models of the local structure while incorporating estimates of the structural behaviour of the rest of the engine in the form of free body loads and displacements. However, these boundary conditions become invalid once a design change is made, and to update them involves a lengthy validation procedure where a coarse mesh of the proposed component design is generated and subsequently integrated into a low fidelity model of the whole engine. In the present work, the existing component design process is improved through a combination of exploiting new finite element modelling technologies, a resourceful use of multi-fidelity surrogate modelling methods, and taking concepts from concurrent design methods, to develop a set of reliable and efficient distributed design strategies. A novel component design-in-context method is proposed to enable components to design directly with respect to physical constraints in the residual system using information from embedded global FE models. The method is found to reduce the degree of system-level infeasibility for an engine component that is designed in isolation by up to 2.7%, with only a moderate 11% increase in simulation cost. A novel safe integration method which is based on a system-level surrogate model augmented with a confidence level is also developed. The method is able to maintain high system-level feasibility when multiple component design-in-context workflows are carried out concurrently, reducing the probability of rework by as much as 5%.

Contents

List of Figures

List of Tables

Nomenclature

CAD Computer Aided Design

CAE Computer Aided Engineering

CFD Computational Fluid Dynamics

CSSO Concurrent Subspace Optimisation

FE Finite Element

FEA Finite Element Analysis

FEM Finite Element Model

GSE Global Sensitivity Equations

HP High Pressure

LP Low Pressure

MA Medial Axis

MAC Modal Assurance Criterion

MANTLE Modeling and Analysis of finite elements in the Neutral Plane

MAT Medial Axis Transform

MDA Multidisciplinary Design Analysis

MDF Multidisciplinary Design Feasible

MDO Multidisciplinary Design Optimisation

MO Medial Object

NASTRAN NASA Structure Analysis

WEM Whole Engine Model

Declaration of Authorship

I, Hau Kit Yong, declare that the thesis entitled *Design-in-Context: Exploiting the Increasing Accessibility of Global FE Models to Improve the Mechanical Design Process of Engineering Subsystems* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:
  - Yong, Hau Kit, Leran Wang, David J. J. Toal, Andy J. Keane, and Felix Stanley. 'Multi-Fidelity Kriging-Based Optimization of Engine Subsystem Models with Medial Meshes'. In Proceedings of the 2018 ASME Turbo Expo, Vol. 2D: Turbomachinery. Oslo, Norway: American Society of Mechanical Engineers (ASME), 2018. https://doi.org/10.1115/GT2018-76148,

– Yong, Hau Kit, Leran Wang, David J. J. Toal, Andy J. Keane, and Felix Stanley. 'Multi-Fidelity Kriging-Assisted Structural Optimization of Whole Engine Models Employing Medial Meshes'. Structural and Multidisciplinary Optimization 60, no. 3 (11 May 2019): 1209–26. https://doi.org/10/gf4pt8,

– Yong, Hau Kit, Leran Wang, David Toal J. J., Andy J. Keane, and Felix Stanley. 'A Medial-Object-Based Design-in-Context Approach for the Structural Design of Engine Subsystems'. In Proceedings of the 2019 AIAA Propulsion and Energy Forum. Indianapolis, IN, USA: American Institute of Aeronautics and Astronautics (AIAA), 2019. https://doi.org/10.2514/6.2019-3829.

Signed:.....

Date:.....

Acknowledgements

There are countless people to whom I have become indebted during my PhD, but I would like to start with my supervisors, Dr. David J. J. Toal, and Prof. Andy J. Keane. I am thankful for Dr. Toal's guidance and encouragement, and especially as a voice of assurance as I struggled through the later parts of my programme while coping with the effects of a pandemic. I have also been inspired by Prof. Keane's breathe of knowledge and attention to detail, having had the pleasure to listen to his discussions ever since I was an undergraduate student in his lectures, to attending his presentations as a delegate of the university at Rolls-Royce.

I am honoured to have been a part of the Rolls-Royce UTC at the university. Dr. Leran Wang has been a huge help and a great friend, always unwavering in his assistance as I wet my feet as a novice researcher. Dr. Ivan Voutchkov has provided invaluable guidance and technical help, always entertaining my never-ending questions about OPTIMAT or the cluster. Dr. Xu Zhang, Dr. Murat Hakki Eres, and Dr. Jennifer Forrester have also provided me great support over these past four years, along with all the great students I've worked with. I'd also like to thank Dr. Felix Stanley of Rolls-Royce. His broad range of experience made him a great resource and I am very appreciative of the steadfast confidence he has shown in my work.

Last but certainly not least, I will always be grateful for the love and support from my family, and the sacrifices that my parents made to make this possible. No words can capture the amount of appreciation I have for them as they walked with me through this part of my life, so I shall not attempt to do so!

Chapter 1

Introduction

1.1 The Role of Whole Engine Finite Element Models

Gas turbine engines are complex systems where ever increasing demands to reduce engine weight and improve fuel consumption has made it necessary for engine components to be tightly integrated with each other. It is thus necessary to build engine-level models in order to capture both the physics of inter-component interactions and also any emergent behaviour that would otherwise be lost if the analysis was done on each component in isolation. An assortment of such models are built, each characterized by the requirements of the life-cycle stage of a particular engine programme and its associated activities.

For the purposes of structural analysis using the finite element (FE) method, such engine-level models are normally referred to in industry as Whole Engine Models (WEM). A WEM is typically an assembly of coarse FE models of the components in the engine's load path, although the selection of which components to include in the WEM, as well as the individual fidelities of each component FE model, differ based on the type of analysis that is performed.

One of the most common applications of WEMs is to conduct investigations of dynamic engine structural failure events. These WEMs necessarily include a 3D shell-and-beam model of the major components in the engine load path but will often be combined with analysis-specific features. Studies on engine-airframe interactions, for instance, will require models of engine integration components such as pylons, nacelles, and the thrust reverser unit, while fan blade-out simulations will require detailed 3D FE models of the disk, several fan blades, and other elements of the axial retention system (Heidari et al. (2008); Sengoz et al. (2015)). Some other applications include tip clearance evaluations that involve transient thermomechanical simulations of a WEM consisting of a 2D axisymmetric model of the rotor and stator overlaid with 3D effects such as blade deformation (Arkhipov et al. (2009); Bordo et al. (2013); Boeller et al. (2018)).

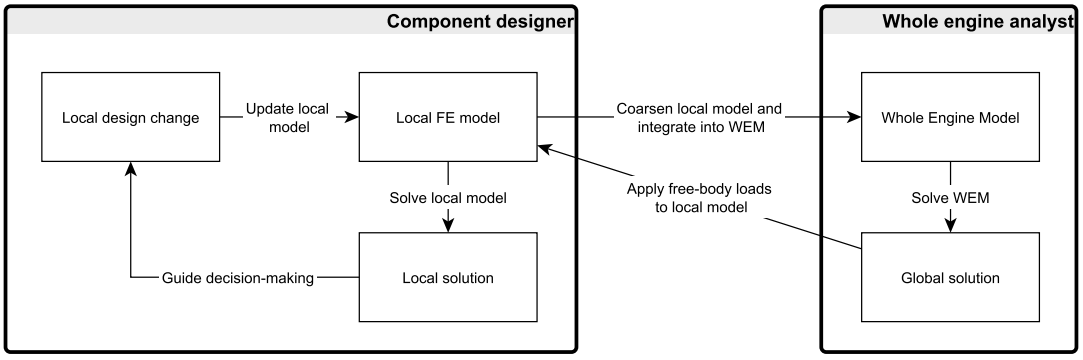


Figure 1.1: The traditional component design workflow. For every analysis of a component design change, the component designer needs to liaise with the whole engine analyst to obtain the updated free body loads for its local FE model. In a realistic engine design programme, however, outdated loads are often used instead due to time constraints.

Besides modelling system-level behaviour, WEMs are also used to both feed and validate component design changes. When used in this context, such FE models are also known as 'global' FE models to indicate its scope. For example, in the design of static structures, WEMs are used to understand and select the optimal configuration for the load-bearing parts. Once this configuration has been frozen and the engine programme proceeds to detail design, the results from these WEM simulations serve as a database of free-body loads and displacements from which component designers can take as inputs to conduct independent local analyses (Dharmasaroja et al. (2017)).

Figure 1.1 depicts the traditional component design workflow and the types of interactions that exist between a component designer and a whole engine analyst. The established workflow enables higher fidelity FE models of the component to be used while incorporating estimates of the structural behaviour of the rest of the engine without having to model the geometry explicitly. Once a design change has been made, the conventional validation approach then is to perform system-level checks by generating a coarse mesh of the proposed subsystem design and integrating it into the WEM (Minnicino et al. (2004)).

As good as this localized modelling approach is at accelerating analysis turnaround times at the component level, the simulation results are susceptible to a number of errors that the average component designer will find difficult to control (Sracic and Elke (2019)). WEMs will not be able to represent high stress gradients across component boundaries faithfully due to their coarseness. The approach makes use of St. Venant's Principle, which states that if the forces acting on a small portion of the surface of an elastic body are replaced by another statically equivalent system of forces acting on the same portion of the surface, the redistribution of loading produces substantial changes in the stresses locally, but has a negligible effect on the stresses at distances which are large in comparison with the linear dimensions of the surface on which the forces are changed.

In effect, only regions sufficiently far away from the where the interface loads are applied will produce accurate stress fields. Significant differences in the mesh resolution between the WEM and local FE models can also introduce considerable interpolation errors.

Perhaps the most serious issue of all from the perspective of a design engineer, however, is that the simulation results are valid only for the very narrow band of design changes that do not upset the equilibrium of the WEM simulation. This means that the only component designs that that can be assessed accurately and in a time frame short enough to drive design decisions are those that do not have a significantly different stiffness from the baseline. For design proposals that do not adhere to this restriction, the true loading seen by the local structure is unknown, and the component FE model is only able to provide qualitative information such as the trend of the response with respect to the changes that have been made. Such information is still valuable during the initial exploratory phases of the design process, but when the final design has to satisfy explicitly defined requirements, quantitative measures of the accuracy will eventually be needed to ensure that those constraints are met.

The two-step nature of the modelling process also exposes designers to a 'validation debt' where the time required to check their designs against a WEM prohibits them from pre-emptively taking the necessary corrective measures in their local iterations. It is unfortunately not uncommon for an engine programme to arrive at the end of a major design iteration when a final WEM is built and solved only to find out that certain components have failed to meet their requirements. The organization is then put into a firefighting mode where it is forced to decide between reverting the problematic components to an earlier compatible state and lose months of work, or trying to restore feasibility through a rework procedure that may involve soliciting compromises from the remaining compliant components at the risk of breaking them.

1.2 The Case for Using Global FE Models to Drive Component Design

If the component-to-WEM integration process can be streamlined, component designers will be more confident in their knowledge of the structural behaviour of their domain and will be more confident when making local design changes. One of the most significant steps that can be taken as part of this streamlining effort is to reduce the two-step process to a single step, by building a single FE model that meets the modelling priorities at both the local and global level, and which only needs to be run once to produce the necessary results. Such a model can be created by inserting the detailed component model within the global FEM in situ to form what is referred to in the present work as an 'embedded' global FE model. Due to the lower fidelity in the non-local regions of the embedded model, only the trends of the responses in these regions can be captured, as

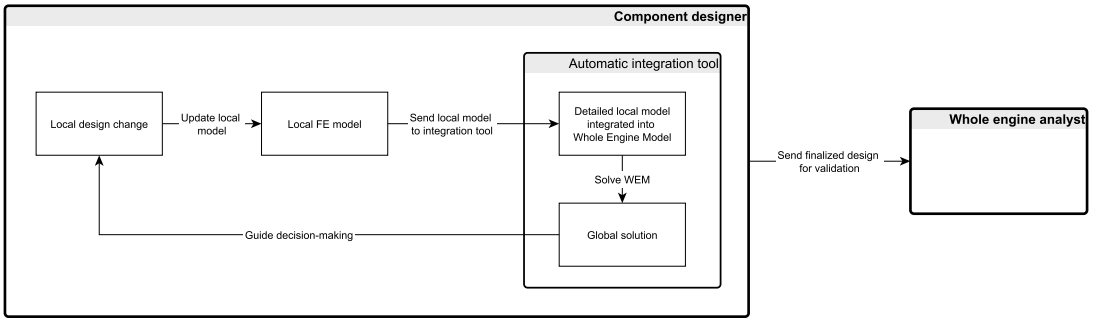


Figure 1.2: The component design-in-context workflow. In the proposed method, the component designer uses an integration tool to automatically embed a detailed component FE model into a low fidelity FE model of the whole engine. This approach circumvents the need for frequent communication between the component designer and the whole engine analyst, thereby reducing the cost of evaluating local design decisions.

with the interface load approach. However, the key advantage of the current approach is that the behaviour of the inaccuracies depend solely on the modelling fidelity and thus can be modelled and controlled.

The method can be regarded as part of the field of integrated global-local modelling which broadly involves mixing models of different fidelities, either by substituting a higher-fidelity local model into a simple global model (Klein et al. (2017, 2018)) or by a reduction of the surrounding region of a model of interest (Kumar et al. (2017)). The distinction is made between in situ and integrated methods for finite element models because the high-fidelity local model is placed directly within the overall global model. In spite of its usefulness, this in situ approach has not been prevalent in the engineering design community and descriptions of its applications in the literature are exceedingly rare. The reason for this is, again, the difficulty of the integration process. The step of reducing the local model has been removed but the activity of incorporating it into the global model with the appropriate contact formulations still requires the expertise of an analyst. The method has thus mostly been relegated to doing the same system-level validation studies of subsystem design changes that a WEM is responsible for, but when the analyst does not have enough time to coarsen the local model.

It is apparent that if the entire integration process can be completed without the division of tasks between the designer and the analyst, true simulation-driven design with these in situ models can be achieved. To this end, Rolls-Royce has developed MANTLE, a library of finite element pre-processing tools. The MANTLE library provides both mesh coarsening and integration capabilities in one package. This technology opens the door to design-in-context approaches where component designers can actively account for both local and global requirements in the design decisions. Figure 1.2 illustrates such an approach.

1.3 The Need for a Reliable Approach for Integrating Component Designs

As the scope of information that is available to the component designer increases, there is a natural inclination for increasing the scope of the design task. In a conventional concurrent design environment, the maximum attainable performance of a component is restricted not only by the physical limitations of the component structure but also by 'interface requirements' that act to prevent the component from inadvertently deteriorating the performance of its neighbouring structures. The values for these artificial requirements are typically determined through a systems engineering practice and can often be poorly prescribed. By making available a tool such as MANTLE that increases the accessibility of system-level structural information, the astute component designer may look towards searching for designs that violate the interface requirements but otherwise satisfies the physical requirements in a larger context.

Such endeavours are inherently useful as they tell the component designer how local decisions affect the behaviour in the residual system, as opposed to the opaqueness of the interface requirements. However, if the component designer then decides to drive local design decisions based on this non-local information directly, there is a risk that the resulting component design will violate those same requirements upon assembly. This can occur because the global FE model assumes that the residual system is static and does not model non-local preferences. There is thus a need for a system-level integration approach that can reliably combine discrete component designs together into a feasible system, while preserving the design freedom that is afforded to each component designer by the availability of global FE models.

1.4 Thesis Objectives and Overview

The main goal of this thesis is the development of methods for improving the accessibility and utility of component-level design optimizations in gas turbine engine development. This is primarily achieved by the development of a finite element modelling workflow that enables global FE models to be generated quickly and are able to provide context to component-level design decisions and in turn, enable them to increase the scope of their design activities. The proposed strategies should lead to the establishment of a design environment that is highly flexible and allows novel designs to be explored, yet provide some assurance that the decisions taken will not lead to infeasibilities in the residual structure.

In order to begin to develop such methods, it is important to understand the application of finite element methods for structural design, what makes the generation of global FE models challenging, and what the inaccuracies are; these points are addressed

in turn in Chapter 2. Chapter 3 then discusses the surrogate modelling approach, motivating the development of a novel multi-fidelity optimization approach mixing solid and mid-surface meshes, as well as a novel design-in-context approach that uses the aforementioned embedded global FE models. Chapter 4 presents a method for safely integrating arbitrary sets of component designs that are generated through concurrent local optimizations. The effect that different combinations of the component design approach and the system-level integration approach has on the convergence characteristics of an overall design process is investigated. Finally Chapter 5 concludes the thesis.

Chapter 2

Accelerated Generation of Global Finite Element Models

2.1 Whole Engine Modelling

Only through studying the engine's structural behaviour in its entirety are we able to distillate the synergistic interactions of its subsystems and components. This has driven engine manufacturers to establish dedicated groups of engineers whose responsibility is to build global finite element models of the engine, models that are often referred to in industry as Whole Engine Models (WEMs). This concentration of expertise is required to handle the complex process of transforming the engine from a detailed CAD assembly into a low degree-of-freedom CAE assembly of sufficient accuracy. However, a large amount of man-hours is still required to build these FE models. Their turnaround times thus exceed most decision-making timescales at the component level, so they are often only used to validate component designs. There is therefore a motivation to streamline the WEM generation process, often through automation or an improvement in its robustness. A recent survey of automatic geometry simplification methods can be found in Thakur et al. (2009). Of these methods, dimensional reduction is perhaps the most widely used for aerospace structures as they are predominantly thin-walled.

2.2 Dimensional Reduction of Solid CAD Geometry

2.2.1 Medial Object-Based Mid-Surface Modelling

Introduced in Blum (1967) for biological shape recognition, the medial axis (MA) can be defined for a solid as the locus of centres of an inscribed sphere of maximal diameter as it rolls within the boundaries of the solid. Taking a more computationally implementable

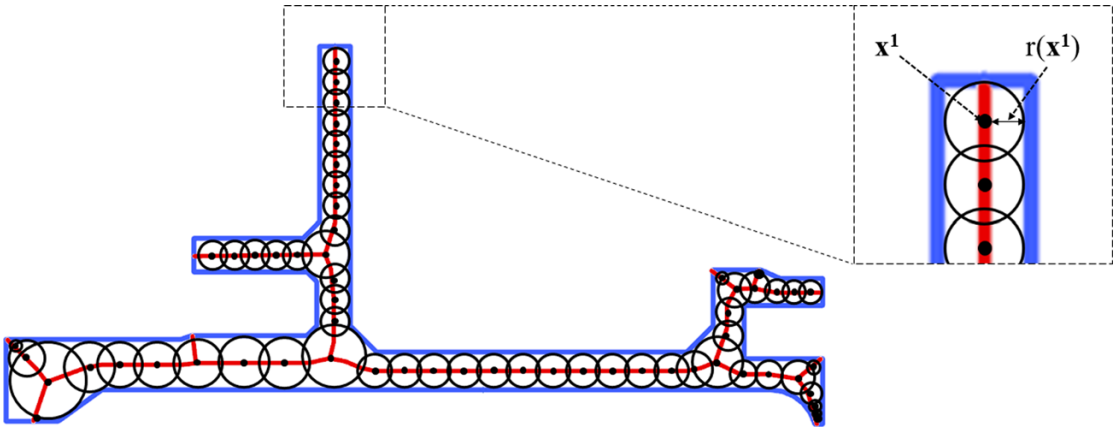


Figure 2.1: The medial axis transform for a HP compressor stage casing. The figure contains a 2D cross-section of a HP compressor stage casing (blue curves), its medial axis (red curves), several maximally-inscribed discs (black circles), and the radius function $r(\mathbf{x})$. Similarly, for 3D geometry, the boundary lines and medial axis are replaced with boundary faces and medial surfaces, respectively.

perspective, the MA can also be defined as the set of points interior to the solid that are equidistant from two or more points on the solid's boundary.

The medial axis transform (MAT) consists of the MA together with a radius function that maps points on the MA to the radius of its maximally inscribed sphere. Figure 2.1 shows an example MAT for an engine component. The MAT provides a simple and compact representation of shape that preserves many of the topological and size characteristics of the original shape. It is a complete shape descriptor, that is, the transformation is invertible and the shape of the original domain can be reconstructed. A MAT of a CAD model that is a CAD model itself on which meshes can be generated is referred to as a medial object (MO). The generation and application of such skeletal representations of geometric models are fields of active research within the computer graphics and image-processing communities (Leymarie and Kimia (2008)). Detailed descriptions of their properties and a survey of the state-of-the-art in skeleton generation can be found in Tagliasacchi et al. (2016).

The maximally-inscribed disc definition has been largely replaced with concepts that lend themselves towards easier algorithmic implementation. Gürsoy and Patrikalakis (1991) described the medial axis in terms of minimum distances. For any closed geometry, a minimum distance-to-boundary value can be computed for every internal point. The MA is then defined as the locus of the set of internal points that have non-unique minimum distance values. The grassfire analogy proposed in Blum (1973) and further developed in Kimia et al. (1995) is also a popular definition. The boundary of a geometry is "set on fire" in a controlled fashion such that the flame front at all locations propagate normally into the geometry at a uniform speed. Flame fronts originating from different parts of the boundary will eventually meet at the medial axis and die out. Each point on the medial axis holds information about the flame front 'arrival' time, which could then

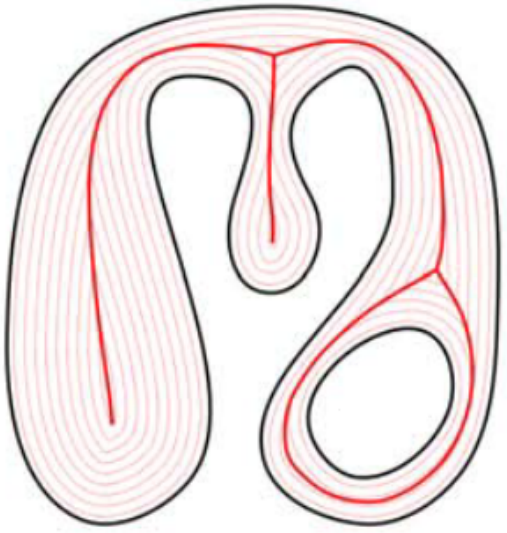


Figure 2.2: The medial axis grassfire (adapted from Tagliasacchi et al. (2016)).

be multiplied by the propagation speed to compute its equivalent of a maximal radius. Figure 2.2 shows the progress of the flame front at several discrete times.

The medial axis is closely related to the closest neighbourhood problems in computational geometry, especially the well known Voronoi diagram (Berg et al. (2008)). A Voronoi diagram is a geometric structure that takes a set of 'sites' as input and sub-divides a region such that every point in the region is assigned to the nearest site. For example, we can take a set of n points defined on a plane, $P := \{p_1, p_2, \ldots p_n\}$, to be the set of sites. The resulting Voronoi diagram, $\mathrm{Vor}(P)$, consists of n cells, one for each site, with the property that a point q lies in the cell corresponding to the site p_i if and only if the Euclidean distance $\mathrm{dist}(q, p_i) < \mathrm{dist}(q, p_j)$ for each $p_j \in P$ with $j \neq i$. The MA is a special case of Voronoi diagrams that uses both points and line segments, specifically the vertices and edges of a polygon, as sites to sub-divide the interior of the polygon itself. Figure 2.3 compares the MA to the Voronoi diagram for a simple 2D shape. They are identical to each other except in non-convex regions.

In a CAE context, the medial object can be treated as a mid-surface and thus represents a useful source of geometry on which shell meshes can be constructed. Shell meshes play a critical role in reducing the computational cost of running simulations on thin-walled geometries which are commonly found in gas turbine engines. Shell meshes exploit the engineers' theory of bending, which in its basic form asserts that stresses on opposite faces of thin objects are equal and opposite. Shell meshes can also take advantage of St. Venant's principle which asserts that modelling discrepancies between solids and 3D shells in localized regions are insignificant as long as the stress distribution in these regions are not of interest and only the overall stiffness of the structural model is desired. The total number of elements in a shell mesh can be significantly lower than an equivalent

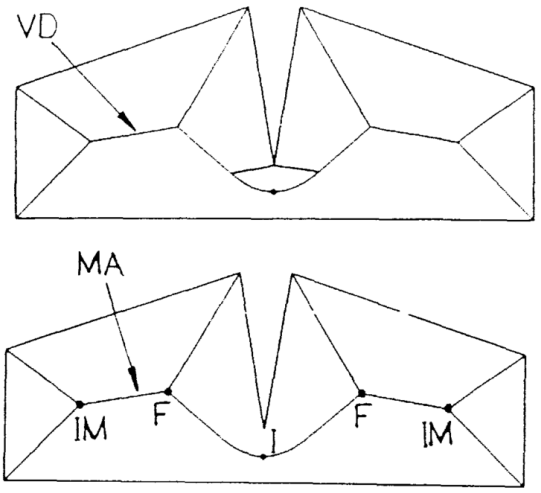


Figure 2.3: The Voronoi diagram, VD, versus the medial axis, MA (adapted from Gürsoy and Patrikalakis (1991)). Three types of branch points are labelled. The intermediate branch point, IM, indicates where three or more MA branches intersect. The initial and final branch points, I and F, indicate the locations of minimum and maximum thickness over the entire shape, respectively.

solid tetrahedral mesh that has to adhere to a minimum elements-through-thickness count to maintain mesh quality.

Bearing in mind the computational cost reductions that mid-surface meshes can offer in simulations, their full potential to accelerate the design process is truly shown when they are re-used in design optimization studies on models with static topologies. Their re-usability stems from how mesh regeneration is not necessary if the only modifications allowed are symmetric thickness changes which are common in late-stage sizing optimizations. Such changes would preserve the symmetric properties of the medial object and would thus be unchanged for the modified design. For example, if the effect of varying the thickness of a flange on the stiffness of an engine casing is to be investigated, the thickness data for each element in that flange can be directly changed by simply searching for the flange elements in the simulation data file and overwriting them with the updated values. Figure 2.4 shows a simulation deck before and after a thickness modification has been performed on a group of elements belonging to a polygon face.

The cost reductions offered by these mid-surface meshes thus become increasingly significant as the number of optimization iterations increases. Some of the limitations of these kinds of rapid geometry modifications are also immediately obvious. Symmetric thickness changes to surfaces in the engine that are located in the gas path are not desirable as the fluid flow would be altered, requiring recalculation of the boundary conditions from expensive CFD analyses. Decisions that alter the topology of the model, such as shifting the position of the thrust linkages on an engine casing (Toal et al. (2014)) or changing the number of stator vanes in a compressor, can also not be realised. These

```
CQUAD4,6760,2,7238,6987,6986,7237,0.0,0.0,+
+,,0,6.822,6.835,3.171,3.17
CQUAD4,6761,2,7238,7240,6989,6987,0.0,0.0,+
+,,0,6.835,3.0,3.0,6.822
CQUAD4,6762,2,6989,7240,7241,6990,0.0,0.0,+
+,,0,3.0,3.0,3.0,3.0
CQUAD4,6763,2,7241,7239,6988,6990,0.0,0.0,+
+,,0,3.0,3.0,3.0,3.0
CQUAD4,6764,2,7239,7242,6991,6988,0.0,0.0,+
+,,0,3.0,3.006,3.006,3.0
CQUAD4,6765,2,6991,7242,7243,6992,0.0,0.0,+
+,,0,3.057,3.057,3.006,3.006
CQUAD4,6766,2,6992,7243,7227,6976,0.0,0.0,+
+,,0,12.792,12.761,3.057,3.057
CQUAD4,6767,2,7244,7227,6976,6993,0.0,0.0,+
+,,0,4.078,12.792,12.761,4.076
```

```
CQUAD4,6760,2,7238,6987,6986,7237,0.0,0.0,+
+,,0,6.822,6.835,3.171,3.17
CQUAD4,6761,2,7238,7240,6989,6987,0.0,0.0,+
+,,0,3.75,3.75,3.75,3.75
CQUAD4,6762,2,6989,7240,7241,6990,0.0,0.0,+
+,,0,3.75,3.75,3.75,3.75
CQUAD4,6763,2,7241,7239,6988,6990,0.0,0.0,+
+,,0,3.75,3.75,3.75,3.75
CQUAD4,6764,2,7239,7242,6991,6988,0.0,0.0,+
+,,0,3.75,3.75,3.75,3.75
CQUAD4,6765,2,6991,7242,7243,6992,0.0,0.0,+
+,,0,3.75,3.75,3.75,3.75
CQUAD4,6766,2,6992,7243,7227,6976,0.0,0.0,+
+,,0,3.75,3.75,3.75,3.75
CQUAD4,6767,2,7244,7227,6976,6993,0.0,0.0,+
+,,0,4.078,12.792,12.761,4.076
```

Figure 2.4: Overwriting CQUAD4 element thicknesses in a NASTRAN simulation deck. The modified thicknesses are highlighted in red.

changes would require the medial object to be re-computed, so in these scenarios the amount of time saved would depend on how efficient the MAT algorithms are.

Having looked at the strengths of the method, the weaknesses of mid-surface meshes need to be addressed. Figure 2.5 shows the junctions in a medial object-based mid-surface mesh where the shell elements have been extruded by their thicknesses. There is considerable overlap in the extruded volumes due to the high number of medial points, each with a different element orientation, that is needed to capture junction details. This can cause a discrepancy in the mass distribution from the original geometry and lead to inaccuracies in modal simulation results. The mass of each element is computed by multiplying its extruded volume with the material density, and the total mass of the mid-surface mesh is just the sum of all element masses. For uniform geometries like flat plates and cylinders, the extraneous mass would be negligible as the element alignment will result in less overlap. For static simulations, the overlapping elements can also lead to extra thickness at the junctions. This phenomena, however, actually improves the accuracy of the simulation as they can represent the effects of fillets when compared to traditional mid-surface models that do not have this modelling capability (Pollard et al. (2017)).

The medial object will also produce geometric artefacts called 'flaps' at the location of every convex point on the model boundary. A flap is a special case of surfaces in the medial object that has a thickness which reduces to zero as the convex point is approached. These surfaces are not what engineers would normally regard as a good mid-surface as they do not represent the topology of the underlying geometry, and they would also exacerbate the mass discrepancy problem. Treatment of flaps is typically done by pruning them from the medial object and then extending the affected surfaces towards the boundaries of the original geometry. This is less of an issue for this project as the MANTLE medial object functions are able to bypass the creation of these branches at the geometry's end-faces based on the user pre-emptively excluding the end-faces from the extraction process (Figure 2.6). Some flaps will still exist at locations where there

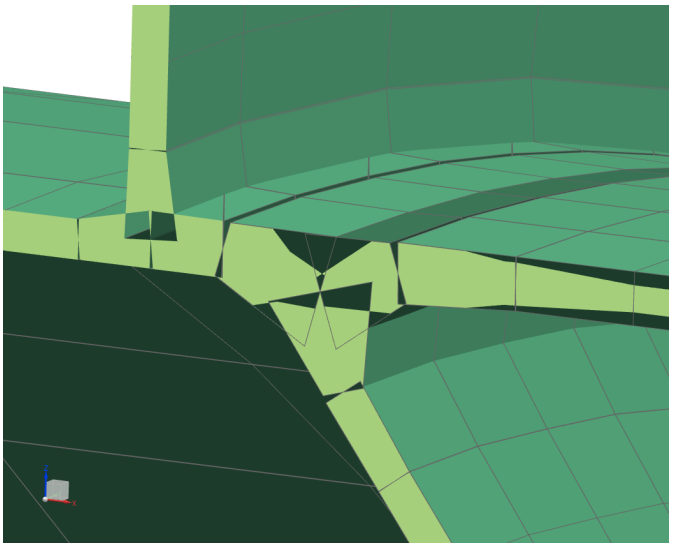


Figure 2.5: Extra thickness at junctions in the medial object.

is a change in the local curvature but these flaps can just be pruned without extension. The quality of the medial object is very sensitive to surface imperfections like free edges and intersecting surfaces which can be an issue if stringent requirements on CAD model quality are not enforced.

2.2.2 Medial Object-Based Mixed-Dimensional Modelling

The desire to automatically generate FEMs for a wider variety of geometries that have a mixture of thick and thin regions has motivated research into mixed-dimensional modelling (Robinson et al. (2011)). The medial object also plays a crucial role in this approach, although they are used to identify and decompose the geometry into simpler and more regularly-shaped units, instead of directly meshing them. Dimensional reduction operators are then applied separately on each region but only on those that satisfy some geometric requirements that have been shown to cause minimal accuracy loss from the simplification. The automatic model segmentation capabilities enabled by the MO have also been exploited for automatic hexahedral-dominant meshing, where they are

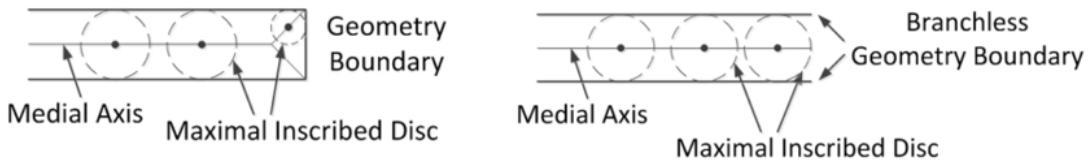


Figure 2.6: Flaps at convex locations in the model boundary, and their removal by the exclusion of the end-faces in the medial object extraction process (adapted from Wang et al. (2014)). The flaps are only present in the left-hand figure. They are the diagonal lines that split off from the medial axis and meet a corner of the geometry boundary.

used to decompose the geometry into sweepable regions (Price et al. (1995); Sun et al. (2018a,b)).

Mixed-dimensional modelling originated from the development of automatic methods for coupling finite elements of different dimensions (Monaghan (2000), McCune et al. (2000), Shim et al. (2002)). Donaghy et al. (2000) used medial axis information to idealize a 2D FEM into a mixed 1D-2D FEM. An identification process was used to identify uniform slender regions, which are characterised by their high aspect ratio and low taper ratio. The aspect ratio was calculated by dividing the shortest bounding edge by the maximum disc diameter in a particular region, while the taper ratio was defined as the maximum rate of change of the maximum disc diameter with respect to the medial edge length. User-defined critical values can be set for these geometric measures depending on the desired degree of model reduction. A model dominated by 1D elements is thus generated when the critical aspect ratio is low. Simulation results showed that by keeping the regions around joints and other discontinuities in their original representations, stress concentrations can be captured well. However, it can be difficult to determine *a priori* suitable critical values for a coveted degree of accuracy. A single critical value for each geometric measure would also not be optimal, as the sizes of regions with interesting structural activity would likely be quite different throughout the model.

The mixed-dimensional modelling approach has been further extended to produce 2D-3D models from asymmetric 3D models. In Chong et al. (2004), the medial object is used to divide the geometry into "primitives", which are simple shapes like cubes and cylinders. The decision to dimensionally-reduce a primitive to its 2D form is made based on a thickness tolerance value. Cutting faces for geometry splitting are determined heuristically using a combination of edge concavity information, island/boss feature recognition, and medians that were generated by collapsing the nearest edges. Mid-surfaces are then extracted from the medians. For non-planar geometries like curved surfaces, bounding shapes are created from the midpoints of the vertices of the surface on which grid points were subsequently generated. The bounding shape is then deformed to match the contour of the curved surface and the mapped grid points represent the mid-surface. For stitching widely-separated patches, edges were generated in the middle of the gaps. The mid-surfaces are then extended to meet at these edges. Alternative mixed-dimensional model generation methods, such as those based on feature information, can be found in Belaziz et al. (2000), Lee (2005) and Cao et al. (2012).

Methods to handle automatic coupling of the 2D and 3D regions were also developed in Robinson et al. (2011). St. Venant's principle was used to calculate suitable inset distances, which is how far a region of a higher dimension should extend into a region of a lower dimension, so that the stress variations can be captured properly. Mixed-dimensional models of an intermediate engine casing were used in free-free modal analyses and also in static stress analyses with circumferential loads. The number of

DoFs in the mixed-dimensional model was 8% of an equivalent solid model. Discrepancies of approximately 8% for the maximum modal frequency, and approximately 6% for the torque response, were reported.

A more recent study on the accuracy of these mixed-dimensional models can also be found in Nolan et al. (2014) where a mixed-dimensional engine intercasing model was validated against coarsely-meshed and finely-meshed tetrahedral FEMs in a free-free modal analysis. The mixed-dimensional model generation time was reported to be significantly longer than the tetrahedral FEM generation time, taking 63 minutes compared to 1 minute for the coarse tetrahedral model. The analysis time, however, was much shorter, taking 5 minutes for the mixed-dimensional model compared to 14 minutes for the coarse model. Across 20 mode shapes, the maximum error in the corresponding natural frequencies for the mixed-dimensional model was 10% while the error for the coarse mesh was 7%. These results demonstrate the potential for such dimensionally-reduced models for quickly obtaining FEA results but they also illustrate the difficulty in efficiently extracting the medial object.

Although mixed-dimensional models are generally more accurate than pure mid-surface meshes in terms of stress and mass, the design freedom offered by the latter can be far greater, as changes made to solid-meshed regions in the mixed-dimensional model cannot be updated without re-meshing and re-coupling with the shell-meshed regions.

2.3 Face Pair-Based Mid-Surface Modelling

Having looked at the methods for dimensional reduction using the medial object, attention has to be directed to the face-pairing approach because of how ubiquitous it is in commercial CAE software. Some methods are based on identifying geometry pairs in the solid geometry, such as edge-pairs or face-pairs, while others use volume-based decomposition methods to split a model into simpler sub-geometries (Woo (2014)).

Mid-surface generation by face-pairing was first proposed in Rezayat (1996). The method starts by iterating over every face in the geometry and identifying end-faces using an aspect-ratio based approach. These end-faces were removed from a set of 'active' faces, and the remaining faces in the set were grouped into pairs using a ray-casting algorithm. Adjacency graphs were then automatically created to capture the topology of the model. Information from these graphs was used to facilitate the generation of the complete mid-surface through a combination of geometric interpolation and 2D Boolean operations. The face-pairing step also generates bi-directional associativity data between the part and the mid-surface, thus boundary conditions for finite element analysis can be projected to the mid-surface naturally and analysis results from the mid-surface can similarly be projected back onto the original model.

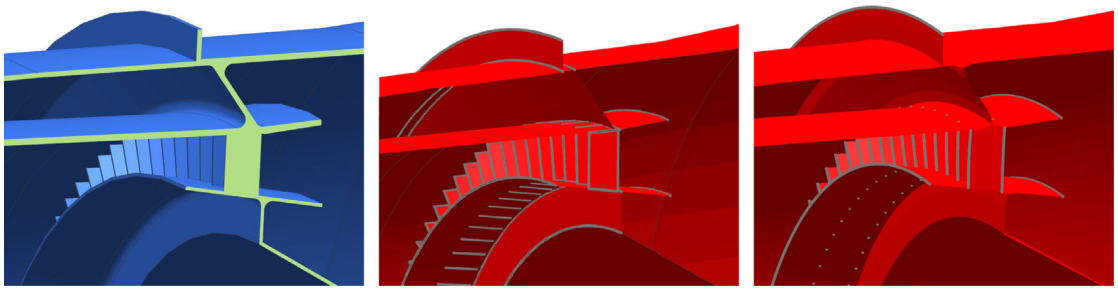


Figure 2.7: Comparison of mid-surfaces for a HP compressor stator model (left). Free edges are highlighted in grey. The face-pairing approach (mid) as implemented in Siemens NX9 Modelling produces patches of mid-surfaces that remain disconnected even after the automatic stitching operations have been applied. The medial object approach as implemented in MANTLE produces a fully connected medial object.

The face-pairing approach is popular because of its speed relative to the medial object-based approaches, although the end products for most geometry of even intermediate complexity are incomplete mid-surfaces that require a significant amount of manual corrections. The approach is not robust against geometries with junctions where opposite faces are not clearly defined. Disconnected mid-surface patches are thus produced for regions away from these junctions and need to be stitched together, often manually, before the model can be meshed. The current iteration of automatic stitching operations regularly fail at locations where the gaps between neighbouring patches are too large, or even in revolved geometries where there are small differences in the mid-surface curvatures (Figure 2.7). The face-pairing approach also produces mid-surfaces that differ based on the simplification criterion used (Figure Kulkarni et al. (2017)). Some work has been dedicated to developing methods for quantifying the quality of a mid-surface model using metrics such as its topological similarity to the original solid model and Hausdorff distances (Lockett and Guenov (2008)). However, these are geometric comparison indices.

2.4 The MANTLE Library

2.4.1 Previous Work

In design optimization studies, simplified models are often used to quickly locate regions of interest in the design space. Furthermore, when coupled with results from high-fidelity models, a good balance between accuracy and cost can be established. The goal of having a dimensional reduction algorithm that is fully-automatic, computationally inexpensive and robust against geometric complexity, has thus spurred the development of MANTLE, a library of proprietary Rolls-Royce medial object generation functions and finite element pre-processing tools that is written in the MATLAB environment.

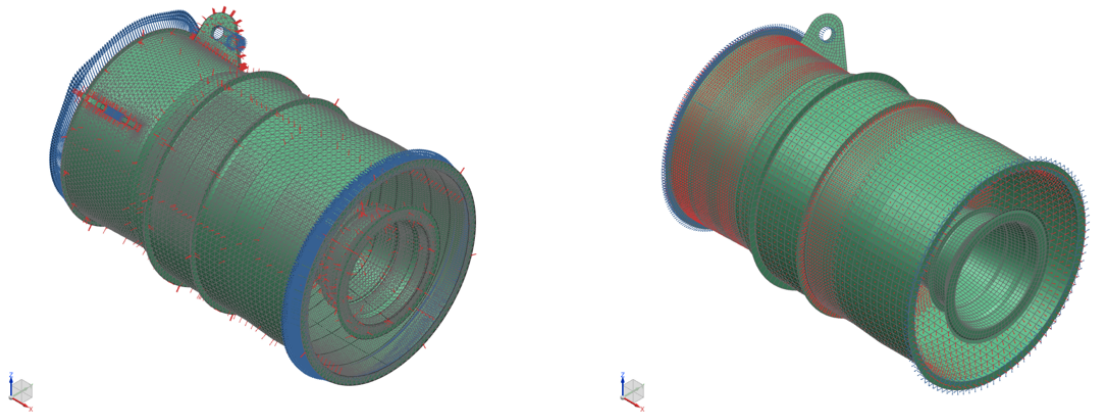


Figure 2.8: Representation of the same set of boundary conditions when applied to a tetrahedral mesh and a mid-surface mesh.

MANTLE was first proposed in Stanley (2010) and at the time of writing consists of three main functions: MANTLE-2D, MANTLE-3D and MANTLE-Core. MANTLE-2D and MANTLE-3D are the primary functions for the medial object generation of axisymmetric and asymmetric geometries, respectively, while MANTLE-Core is used as a tool for setting up engineering workflows. MANTLE-2D and MANTLE-3D call sub-functions for medial point cloud generation, duplicate node removal, and medial surface topology preservation. The capabilities of MANTLE-Core include the setting up of simulation parameters, such as the definition of boundary conditions on tagged sections of a geometry and the assembly of meshes for complex geometries and larger subsystems.

Compared to most medial object generation algorithms which take the geometry boundary as input, MANTLE computes the MO from a triangulation of the geometry's surfaces instead. Using a surface mesh as the input provides several advantages. First, this approach eliminates the problem of flaps at the end-faces since these faces can be left un-meshed and thus are ignored by the MO generation functions. By providing the user with control over the set of active faces, parallel computation of the MO is also enabled by simultaneously processing sub-divisions of the geometry. These MO segments can then be re-assembled using the MANTLE-Core function to obtain the complete MO.

The second advantage is that by associating nodes on the mid-surface mesh with nodes on the surface mesh, boundary conditions, which are often applied on the CAD model surfaces, can be transferred automatically to the mid-surface mesh (Figure 2.8). This is equivalent to the bi-directional associativity property in mid-surface models.

The MANTLE library has been demonstrated on multiple gas turbine component geometries. Wang et al. (2014) generated mid-surface meshes of a combustion chamber outer casing and a compressor intercasing and compared their deformations against equivalent tetrahedral meshes under simple load cases. Displacement errors of under

15% were observed while the solution time for a mid-surface mesh was shown to be, on average, 70% of the solution time of a tetrahedral mesh. The reported cost savings were noted to be lower than expected as MANTLE was at its infancy and the resolution of the mid-surface meshes were higher than necessary in large uniform regions. Wang et al. (2017) used MANTLE to generate mid-surface meshes of a realistic engine model and demonstrated the mesh assembly workflow for producing a Whole Engine Model. Joins consisting of rigid spider elements (specifically "RBE3" elements in the NASTRAN FE solver) were automatically generated between components. The resulting WEM contained 27 components, 26 of which were converted to their mid-surface form. The WEM was used in a mass-stiffness trade-off study.

2.4.2 A MANTLE Mid-Surfacing Workflow

A FE pre-processing workflow based on the MANTLE library was developed for the engine component geometries in the present work. Figure 2.9 illustrates the application of this workflow on a HP compressor casing. Improvements to the MANTLE library were made, including an element-wise thickness remapping function and a function that automatically transfer boundary conditions from component meshes to the WEM. These improvements were written in MATLAB while applications for automating the FE simulations were written in C# using the Siemens NX9 Open API.

To make the overall MO generation more efficient, the solid model is first decomposed into segments after a visual interrogation of the model for geometric patterns. At first glance, the casing appears to be asymmetric, due to the two thrust linkages at the front, and the presence of stator vanes between the compressor and the combustion chamber. Passing this model straight into the MANTLE-3D function would cause the MAT computation time to be extremely high, since it is highly dependent on the number of complex junctions in the geometry.

On closer inspection, the stator vanes were observed to be evenly-spaced, with a 4-degree separation between every vane. A reduction of up to 98.9% in the MAT computation time can thus be achieved from this decomposition. The MANTLE-Core function can then be used to duplicate this pattern around the casing centre axis to get the complete MO. The casing segment containing the thrust linkages, on the other hand, was treated separately as the spacing between the geometric features are different. Figure 2.10 shows how the casing geometry was split to accelerate the MAT computation.

The next step in the mid-surfacing procedure is geometry tagging. Tagging is used to prevent the creation of flaps and to enable the setting of boundary conditions that would normally be defined on the solid CAD model to be automatically transferred to the mid-surface mesh. To exclude a face from the surface mesh, the current method of operation is to simply change the colour property of that face to a pre-specified colour.

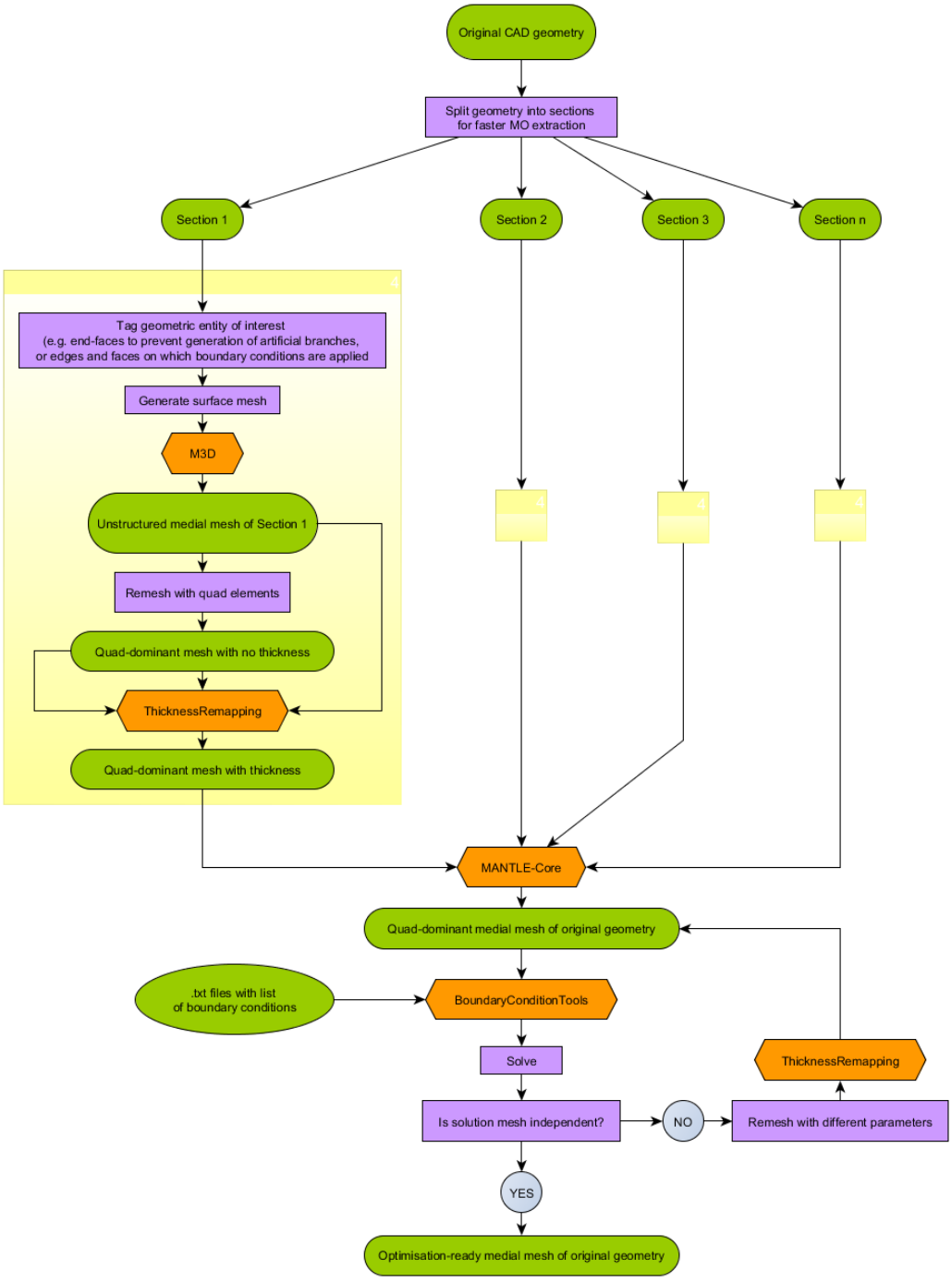


Figure 2.9: An overview of the MANTLE mid-surfacing workflow. The symbols in the figure are colour-coded as follows: Green symbols are computer files such as geometry and data, purple symbols are processes that require manual input, and orange symbols are processes that have been automated by the MANTLE software. The figure highlights the areas in which the MANTLE software has accelerated the traditional FE pre-processing workflow. A detailed description of each step in the process is provided in Section 2.4.2.

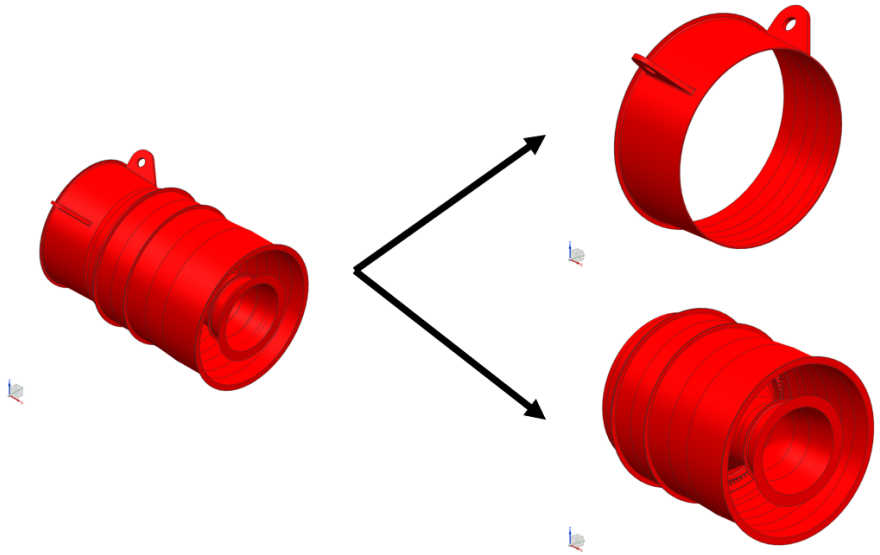


Figure 2.10: Splitting a casing geometry into semi-axisymmetric parts to accelerate the medial axis transform computation.

To tag an edge or a face for a boundary condition, the user need only give a name to that face which indicates that it is a feature of interest. At the time of writing, the MANTLE-3D function is not able to map boundary condition tags from two opposing polygon faces onto the same point on the mid-surface mesh. A workaround was devised where the smaller of the two opposing faces is tagged. A single, condensed boundary condition computed by taking the resultant of all local load vectors is then applied to this tagged face.

Once tagging is complete, surface meshes of the decomposed solid segments are generated and passed to the MANTLE-3D function for MAT computation and a subsequent triangulation of the resulting medial point cloud to create the initial mid-surface. This mid-surface is unstructured and may contain missing elements and elements with poor aspect ratios due to the stability of the algorithm in its current implementation. Figure 2.11 shows some defects near a junction in the initial mid-surface. The mesh can be fixed by filling in the holes with new elements and subsequently fitting polygon faces to the surface as a temporary geometry for any re-meshing operations.

Once a quad-dominant mesh of sufficient element quality has been generated, thicknesses can be mapped to the mesh. The previous re-meshing steps would effectively rewrite the node and element information in the original mid-surface mesh, which would lead to the loss of any thickness information that had been computed by the MANTLE-3D function. A thickness remapping function was thus written in MATLAB to overcome this problem. For each node in a new coarsened mid-surface mesh, the program would search for the nearest node in the initial mid-surface mesh using a quadtree k-nearest-neighbour search and subsequently transfer this node's thickness to the coarsened mid-surface mesh. No interpolation was done to account for spatial differences between the two nodes, and as

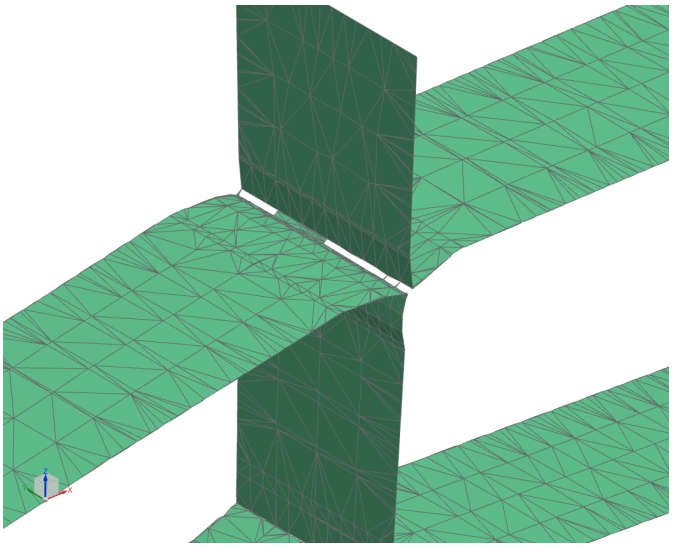


Figure 2.11: Defects in the initial mid-surface mesh from MANTLE-3D, such as missing elements and sliver elements.

such the suitability of this method depends on the density of the initial mid-surface mesh. However, this was not an issue for the models in the present work as the MANTLE-3D function tends to generate very dense initial mid-surface meshes.

Finally, the MANTLE-Core function is used to duplicate the mid-surface segments and translate them appropriately. Figure 2.12 shows the sequence of geometries and meshes created in the MANTLE mid-surfacing workflow.

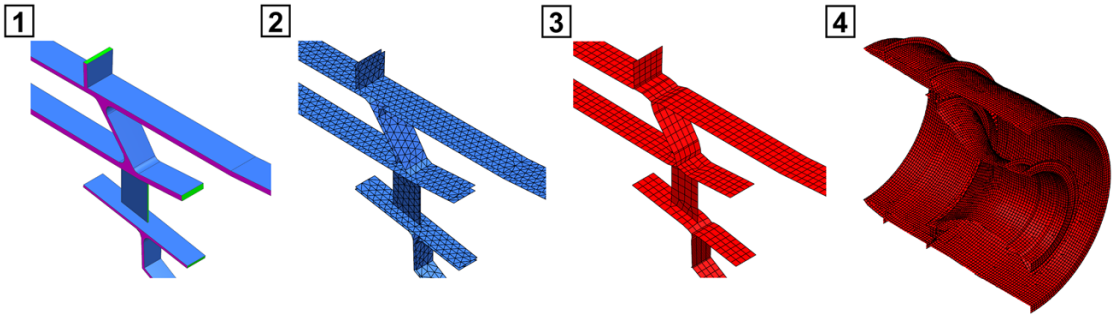


Figure 2.12: The sequence of geometries and meshes in the MANTLE mid-surfacing workflow for the stator segment of a HP compressor casing. (1) The geometry is reduced to a single stator vane segment. The end-faces (shown in green) and the split faces (shown in pink) have been tagged. (2) A surface mesh of the solid segment is generated. The tagged faces are observed to have been excluded. (3) MANTLE-3D takes the surface mesh as input and generates its mid-surface mesh while preserving the tagging information. (4) MANTLE-Core uses the tags to identify split-faces and subsequently revolves the mid-surface mesh of the segment into the complete mid-surface mesh of the full casing.

2.5 Validation of Mid-Surface Meshes

In this section, the accuracy of the MANTLE-generated mid-surface meshes will be demonstrated using two studies. The first study is a linear static analysis on a simple Whole Engine Model, where the deformations in the tetrahedral and shell meshes at several common coordinates are compared. The second study is a normal modes analysis on the LP turbine casing from the aforementioned WEM, where natural frequencies and mode shapes are compared. The choice of running the vibration simulation on just the LP turbine casing was made due to the large amount of memory and disc space required to do the post-processing of results.

The metrics used to describe the resources needed for successful runs were obtained from the performance monitoring .f04 file. The .f04 file contains information such as listings of memory and disk usage statistics, summaries of physical file I/O activity, as well as a time-logged summary of the module execution sequence, useful for determining bottlenecks in the solution process. Indeed, for simulations on realistic engine models, understanding this information is crucial for making good judgements on the computational resources needed so that extra resources can be pre-allocated without having to wait for the solver to return a failed result.

Both of the current studies were performed on a workstation containing a 4-core Intel Xeon E5-4640 CPU clocked at 2.40GHz and 256GB of installed memory.

2.5.1 Linear Static Analysis

The geometry used in this study is the CRESCENDO engine as used in Toal et al. (2014). Figure 2.13 is a lengthwise split view of the engine geometry and its 4 components. The model was constructed such that its longitudinal axis is parallel to the global x-axis, while its transverse plane is parallel to the global y-z plane.

Tetrahedral and shell Whole Engine Models were generated using the original solid CAD geometry and its mid-surface, respectively. The load case was obtained from a sample analysis that was previously performed on the same geometry in an industrial study using a proprietary thermomechanical tool from Rolls-Royce. The load case corresponds to the engine at cruise condition. Thrust in the form of uni-directional forces are applied to every compressor stage, while static pressure loads act on every surface. The WEMs are constrained in the y- and z- axes at the fan casing and LP turbine casing mounts, and in the x-axis at the HP compressor casing thrust lugs. The simulations were executed in NX 9 Nastran within the SOL 101 solver environment. In practice, the thermal loading is modelled in order to capture the tip clearances accurately. Inputs to such a thermomechanical model include temperature-dependent material properties and parameters describing the heat transfer within the engine such as air temperatures,

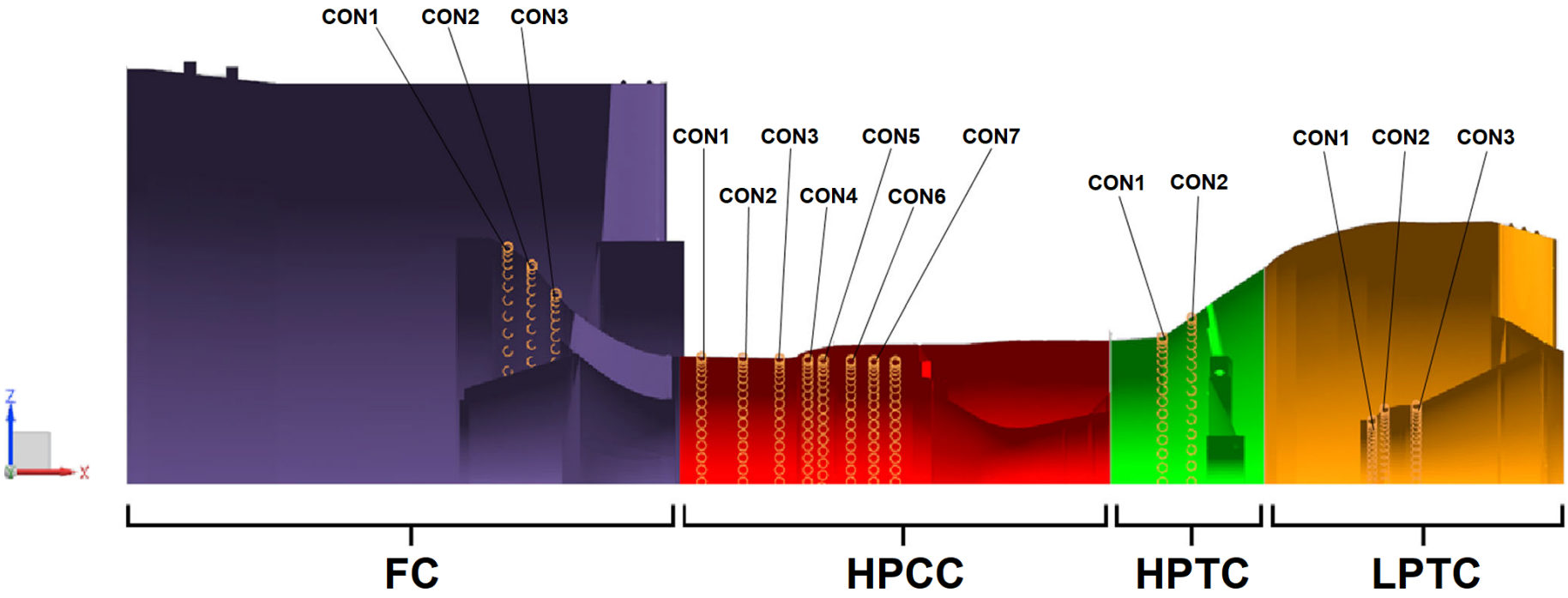


Figure 2.13: Section view of the CRESCENDO engine geometry, its components, and the locations of the output sets that were used for post-processing in the static structural analysis. There are four components: the fan casing (FC), the high-pressure compressor casing (HPCC), the high-pressure turbine casing (HPTC) and the low-pressure turbine casing (LPTC). The labels CONi represent the maximum radial displacements of each output set.

heat transfer coefficients, and flow directions. This additional simulation fidelity significantly affects engine deformation, but is not explored here in order to keep simulation costs low. The behaviour of mid-surface meshes when subjected to thermal boundary conditions will need to be explored in future work.

For this study, the displacements at a total of 960 points (64 points evenly distributed on the circumference of the WEM at 15 discrete x-positions) were extracted. The displacements were post-processed into a maximum local radial displacement value, $\max u_r$ for each x-position. These values are used as approximations of the maximum pinch point metric that is commonly used in tip clearance studies of gas turbine engine casings. The post-processing first involved fitting least-squares circles to the coordinates of the interrogated points after deformation for each of the 64-point sets. The radial displacement at each point is then computed as the radius of the post-deformation displacement points from the y-z plane origin minus the radius of the least squares circle. This least-squares circle approach was used to account for any rigid-body translations.

Mesh sensitivity studies were performed for each engine component independently. Table 2.1 shows the results of the comparison for the converged meshes. All percentage differences fall under the 15% mark as claimed for the MANTLE tool in Stanley (2010). The error in the HP and LP turbine casings are markedly higher than the other engine components. This is because the magnitude of the deformations in those components under the tested set of boundary conditions is low. Table 2.2 shows the computational resources that were consumed by both simulations, where the cost benefits of using mid-surface meshes can be clearly observed.

2.5.1.1 Normal Modes Analysis

Conducting mesh convergence studies using displacement results from static structural simulations is one of the most common approaches in industry as they are relatively simple to set up and are computationally affordable. However, when accuracy over a wider range of scenarios is of paramount importance, modal assurance criterion (MAC) values from a free-free modal analysis can be a better indicator of model accuracy (Chen (2001); Allemang (2002); Pastor et al. (2012); Towner and Band (2012)). The MAC is a statistical indicator that is used to represent the similarity in mode shapes between two modal data sets, and in addition to FEM-FEM validation it is also most commonly used to validate simulation results with results from physical experiments.

The MAC value between two mode shape vectors $\{\phi_r\}$ and $\{\phi_t\}$ is calculated as follows:

$$\mathrm{MAC}(r_i, t_j) = \frac{|\{\phi_r\}_i^T \{\phi_t\}_j|^2}{(\{\phi_r\}_i^T \{\phi_r\}_i)(\{\phi_t\}_j^T \{\phi_t\}_j)} \tag{2.1}$$

Table 2.1: Results from the linear static analysis validation study.

maxu_r (mm)	Solid mesh	Mid-surface mesh	% Error
FC			
FC.CON1	2.2738	2.2583	0.6792
FC.CON2	2.0428	2.0390	0.1879
FC.CON3	1.8941	1.8685	1.3468
HPCC			
HPCC.CON1	1.4249	1.3841	2.8631
HPCC.CON2	1.3055	1.2825	1.7605
HPCC.CON3	1.2249	1.1749	4.0779
HPCC.CON4	1.1979	1.1453	4.3988
HPCC.CON5	1.0547	1.0082	4.4050
HPCC.CON6	0.9820	0.9326	5.0314
HPCC.CON7	0.8847	0.8526	3.6289
HPTC			
HPTC.CON1	0.2414	0.2692	11.4998
HPTC.CON2	0.1524	0.1300	14.7104
LPTC			
LPTC.CON1	0.1170	0.1066	8.8958
LPTC.CON2	0.1936	0.1795	7.3063
LPTC.CON3	0.3205	0.2952	7.8777

This is a normalized scalar product of the two mode shape vectors. The subscripts r and t denote the reference and test data sets respectively, while i and j are mode numbers. $\{\phi_t\}_j$ thus refers to the modal vector of the j^{th} mode shape from the test data set. The superscript T is a conjugate transpose operator. The results from comparing n modal vectors in the two models can be arranged into an $n \times n$ MAC matrix.

The MAC value can range from 0 (no correlation in mode shapes) to 1 (perfect modal correlation). An accurate model would thus produce a MAC matrix that has values near 1 in the main diagonal and near-zero values everywhere else, which indicates that only the mode shapes in the same mode number are well-correlated. Main diagonal MAC values of > 0.9 are normally used as a benchmark in the industry to account for numerical noise and spatial-aliasing when not enough unique node pairs have been identified. The actual pairing of the mode shapes are based on comparisons of the natural frequencies and is done separately before the MAC calculations.

The geometry used in the current study is the CRESCENDO LPCC. The tetrahedral and shell meshes were generated using the settings that were determined through the

Table 2.2: Computational resources expended for the linear static analysis validation study.

Resource	Solid mesh	Mid-surface mesh
Element count	2.7938E+06	3.5164E+05
Node count	5.5868E+06	3.3746E+05
Solution time (hours)	7.5828	0.6508
Max memory usage (GB)	19.9692	0.7513
Max disk usage (GB)	90.2837	39.8240
Total I/O (GB)	555.9546	189.6931

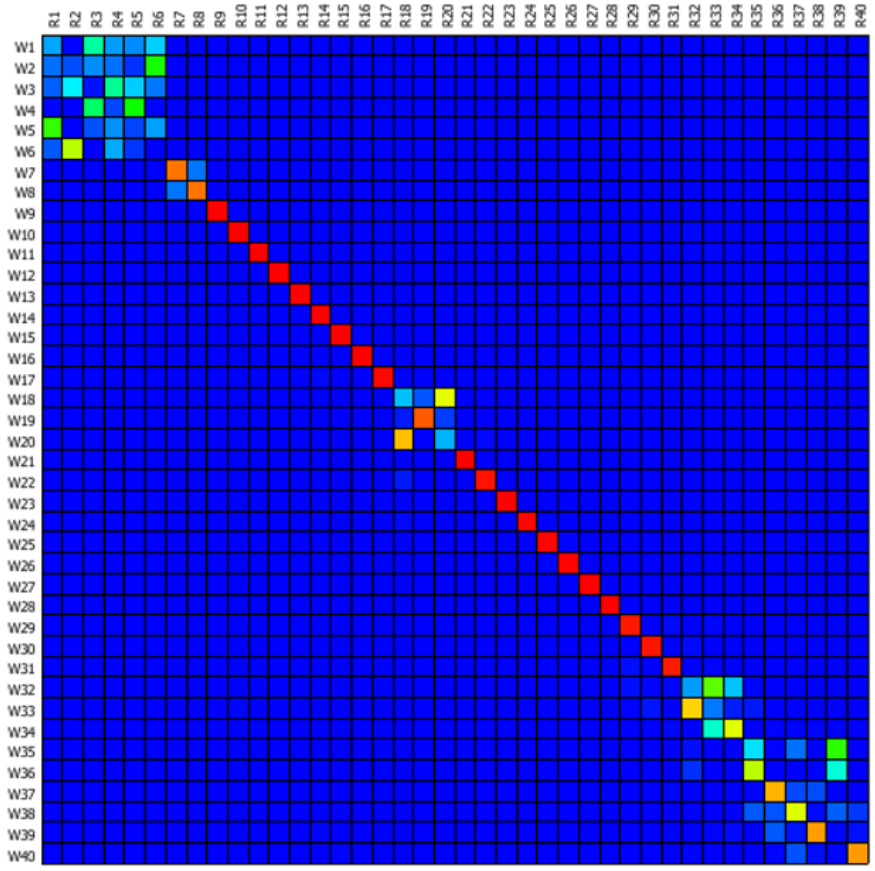


Figure 2.14: The Modal Assurance Criteria (MAC) matrix for the first 40 modes of the CRESCENDO LPCC. R and W represent the tetrahedral and shell mesh modes, respectively.

mesh convergence study in Section 2.5.1. The simulations were run in the NX 9 Nastran SOL 103 solver environment for the first 40 modes under free-free conditions (no loads and constraints). Figure 2.14 shows the MAC matrix plotted as a heat map. Table 2.3 shows the MAC values on the main diagonal along with the natural frequency errors of each mode.

Table 2.3: Comparison of the natural frequencies for the critical modes from a free-free modal analysis of a solid and shell mesh of the CRESCENDO LP turbine casing.

Mode	MAC	Solid mesh (Hz)	Shell mesh (Hz)	% Difference
7	0.89	176.60	178.55	1.10
8	0.89	176.68	178.68	1.13
9	1.00	292.08	291.70	0.13
10	1.00	446.05	452.98	1.55
11	1.00	446.56	453.50	1.56
12	1.00	794.33	813.85	2.46
13	1.00	813.87	834.56	2.54
14	1.00	876.62	891.13	1.66
15	1.00	876.98	891.68	1.68
16	1.00	885.95	901.37	1.74
17	1.00	926.73	931.42	0.51

The MAC for the first 6 modes have near-zero values due to the 6 rigid body modes (3 translational and 3 rotational) in unconstrained 3D structures. The natural frequencies of these six modes are also almost zero, indicating no internal displacements due to rigid body motion. The number of mode shapes that have to be considered varies by component but for compressor casings specifically, the range of rotor RPMs during operation is the determining factor. In this study, a limit of 60000 RPM, or 1000 Hz, was used. The highest mode number of interest is thus mode 17.

All main diagonal MAC values in this mode number range satisfy the > 0.9 condition for good model accuracy except for modes 7 and 8 which fall just short. In these circumstances, the modal correlation may still be regarded as sufficiently high if the overall MAC matrix shows clear identification of the mode pairs. This can be seen in Figure 2.14 where all off-diagonal MAC values are approximately zero.

Chapter 3

Component Design-in-Context with Global Finite Element Models

3.1 Surrogate Modelling for Optimization

The previous chapter demonstrated that the medial object-based mid-surfacing and finite element model integration capabilities provided by the MANTLE software package can be used to generate global FE models rapidly. The objective now is to investigate how component designers can exploit this larger scope of structural information, especially for generating designs that can meet both local and global requirements. A Design-in-Context method is thus proposed. The method uses simulation results from an embedded global FE model to drive surrogate-based optimizations at the component level. The scope of the optimization case studies in the present work is limited to single-objective, sizing-type optimization problems with continuous variables. Multi-objective problems would be a relatively better representation of actual engineering design studies, but for the purpose of demonstrating the principles of the proposed method, the current scope should suffice.

Setting up the optimization problem first involves the assembly of design variables, $x_1, x_2, \ldots, x_n$ into a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. For continuous problems, this vector belongs to the subset X of the n-dimensional real space $\mathbb{R}^n$ as defined by the bounds on each x_i. The subset X is commonly referred to as the design space. A measure of goodness or an *objective function* can then be quantifiably expressed in terms of the design variables, that is, $f(\mathbf{x})$. A realistic problem also often involves several constraints in the form of functional relations between the design variables, such as $h(\mathbf{x}) = 0$ and $g(\mathbf{x}) < 0$, that shall be satisfied for the design to be acceptable. The values of $\mathbf{x}$ that satisfy all constraints comprise the feasible domain of the problem.

An optimization problem with a single objective can thus be stated as

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}), \\ &\text{s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ &\quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{aligned} \tag{3.1}$$

where $\mathbf{h} = (h_1, h_2, \ldots, h_p)^T$ is a set of p equality constraints, and $\mathbf{g} = (g_1, g_2, \ldots, g_q)^T$ is a set of q inequality constraints.

The functions f, $\mathbf{h}$ and $\mathbf{g}$ can be expressed in different forms. They are most often given as explicit algebraic expressions that are derived either from the fundamental equations and laws in engineering and science, or from a curve-fitting procedure to empirical data. Additionally, their evaluation may involve complex calculations within a *numerical simulation* that can only be solved using a computer program, such as the FE models that have been discussed thus far. Concerns with using simulations to drive optimizations include long running times and the lack of explicit gradient information. Short design cycles also mean that optimizations are often terminated prematurely, so an approach that also leaves the design engineer with a greater understanding of the model to stimulate subsequent work is highly desirable.

Surrogate modelling is one such approach. The idea behind surrogate modelling is to invest a proportion of one's computational budget up-front for building fast mathematical approximations to the slow-running simulations. An outline of the procedure is shown in Figure 3.1. Surrogate models can also be used to bridge between various levels of accuracy provided by simulation models of varying fidelity, an example being the mixed-dimensional in-situ models and full-shell Whole Engine Models. An important consideration in the approach is the selection of the surrogate model structure. The Kriging method and its multi-fidelity auto-regressive variant, Co-Kriging, have been chosen for the test cases in the current work for their suitability in approximating smooth engineering functions and the usefulness of its error estimator in enabling efficient exploration of a design space.

3.2 The Surrogate Modelling Method

3.2.1 Generating the Initial Data Set

Surrogate models are essentially used to quickly estimate the mapping $\mathbf{x} \rightarrow f(\mathbf{x})$. Before the surrogates have the ability to make these fast predictions, they must be trained with some observational data of the same form, i.e. $\{\mathbf{x}^{(i)} \rightarrow f(\mathbf{x}^{(i)}) | i = 1, \ldots, n\}$, where n is

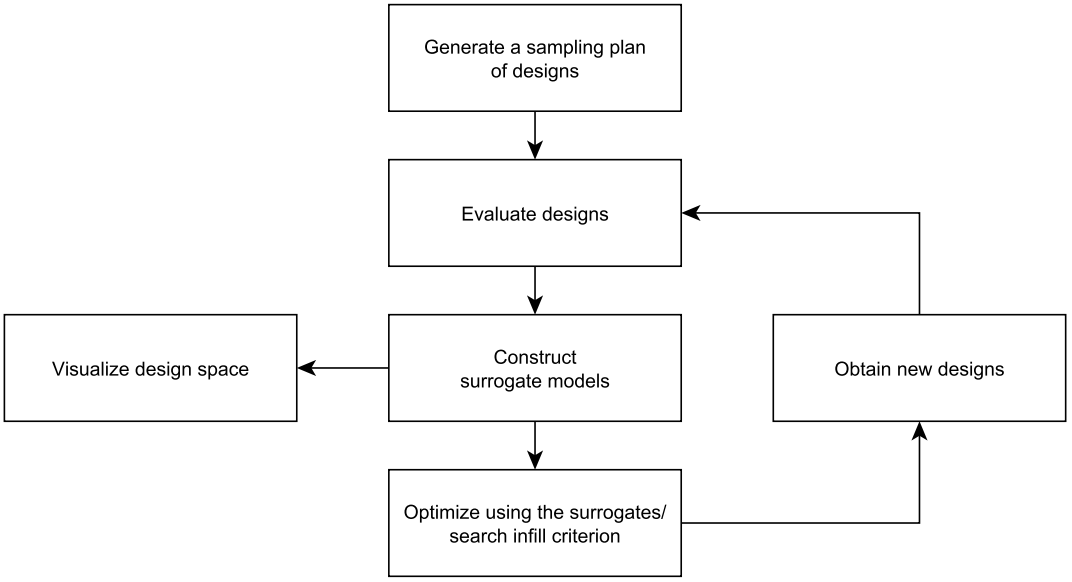


Figure 3.1: The surrogate modelling method. An initial sampling plan of designs is generated using a DOE method and is subsequently evaluated for objective and constraint response data (Section 3.2.1). A surrogate model is then constructed for each response of interest (Section 3.2.2). The surrogates can then be used to provide fast approximations of expensive computer simulations in an optimization process. A number of infill criteria are used to guide the optimization process, and a set of promising designs are obtained (Section 3.2.3). These new designs are evaluated and added to the existing dataset, and the method returns to the surrogate construction step. This loop is repeated until a termination condition is reached.

the number of points in a sampling plan. As obtaining this training data set involves running the simulation $f(\mathbf{x})$ several times, thoughtful selection of the sample locations is needed.

In general, the sample locations should be dispersed uniformly over the design space so that the surrogate model need not extrapolate over large distances to make its predictions. This process of choosing sample locations is typically carried out using design of experiment (DOE) methods. One option is to use a space-filling Latin Hypercube (Keane and Nair (2005)) which produces sampling plans where each sample is the only one in its containing hyperplane.

Through testing different surrogate model settings on a variety of optimisation problems, Jones et al. (1998) recommends a training data set of size $10d$, though for simpler responses $4d$ can be sufficient. The ultimate determining factor, however, is the computational budget that is available. If the surrogate model is to be updated with new points, a rule of thumb is to allocate a third of the budget for the sampling plan (Sobester et al. (2004)).

3.2.2 Constructing the Surrogate Model

Having run the simulations at the sample points, a surrogate model can be constructed. The surrogate model construction process consists of two stages. First, a modelling type is chosen. This is normally done with some idea of what the response should look like. If the governing physical equations are largely linear, simple linear approximations that are cheap to both construct and use for predictions would suffice. However, if the function is highly irregular and multi-modal, using an over-simplified surrogate structure could cause the search to miss promising regions among unattractive zones. Conversely, using complex surrogate structures that have a large number of parameters could lead to over-fitting, where the noise that is present in the simulations corrupts the underlying response.

Surrogate models can be expressed in the generic form $\hat{f}(\mathbf{x}, \mathbf{w})$, where $\mathbf{x}^{(i)}, i = 1, ..., n$ represents the sample locations and $\mathbf{w}$ represents a vector of parameters to be tuned. The conventional way to do this is to minimise a suitable error or loss function. Let the training data set be,

$$\{(\mathbf{x}^{(1)}, f(\mathbf{x})^{(1)}), (\mathbf{x}^{(2)}, f(\mathbf{x})^{(2)}), ..., (\mathbf{x}^{(n)}, f(\mathbf{x})^{(n)})\}. \tag{3.2}$$

The outputs from an un-tuned surrogate model would then be

$$\{(\mathbf{x}^{(1)}, \hat{f}(\mathbf{x}, \mathbf{w})^{(1)} \pm \epsilon^{(1)}), (\mathbf{x}^{(2)}, \hat{f}(\mathbf{x}, \mathbf{w})^{(2)} \pm \epsilon^{(2)}), ..., (\mathbf{x}^{(n)}, \hat{f}(\mathbf{x}, \mathbf{w})^{(n)} \pm \epsilon^{(n)})\} \tag{3.3}$$

where ϵ represents the prediction errors at the sample locations. The tuning process is then centred around minimizing these errors.

There is no guarantee that matching the surrogate model predictions to the true response at the n number of discrete locations means that the predictions would be accurate everywhere else (James et al. (2013)). However, if n is sufficiently large and the sample points are well-distributed over the design space, one can at least be confident that the extrapolation distance would not be a main contributor to the surrogate's inaccuracy.

Here two methods for estimating $\mathbf{w}$ will be presented: the maximum likelihood approach, and the cross-validation approach. The proprietary optimization toolbox OPTIMATv2 (Toal et al. (2008, 2009, 2011)) that was used to solve all the optimization problems in the present work uses the maximum likelihood approach to build Kriging models, and the cross-validation approach to build Gaussian radial basis function models.

Maximum Likelihood Estimation

In statistics, the *likelihood* can be described as a quantitative measure of the extent to which a given data set supports a set of parameters in a parametric model. The concept of likelihoods is normally taken as the inverse of the concept of probabilities, where probabilities quantify predictions but likelihoods quantify the level of trust in a model. For example, given a set of values for $\mathbf{w}$, $p(f(\mathbf{x})|\mathbf{w})$ is the probability that we would obtain $f(\mathbf{x})$ given model parameters $\mathbf{w}$. However, when we are building a surrogate, $\mathbf{w}$ is often unknown, and instead we have a finite number of observations $f(\mathbf{x})$. The likelihood is thus the probability of obtaining $\mathbf{w}$ given $f(\mathbf{x})$. This can be expressed as

$$P(\{f(\mathbf{x})^{(1)}, ..., f(\mathbf{x})^{(n)}\}|\mathbf{w}) = \prod_{i=1}^{n} P(\{f(\mathbf{x})^{(i)}\}|\mathbf{x}^{(i)}, \mathbf{w}). \tag{3.4}$$

Assuming the errors ϵ can be represented as a normal distribution with standard deviation σ, the likelihood function becomes

$$P = \frac{1}{(2\pi\sigma^2)^{n/2}} \prod_{i=1}^{n} \left\{ \exp\left[-\frac{1}{2} \left(\frac{y^{(i)} - \hat{f}(\mathbf{x}, \mathbf{w})}{\sigma} \right)^2 \right] \epsilon \right\}. \tag{3.5}$$

We can convert these products into sums by taking the natural logarithm of Equation 3.5,

$$\sum_{i=1}^{n} \frac{\left[y^{(i)} - \hat{f}(\mathbf{x}, \mathbf{w}) \right]^2}{2\sigma^2} - n \ln \epsilon. \tag{3.6}$$

The model parameters are then obtained by minimising Equation 3.6.

Cross-Validation

In the cross-validation approach, a subset of the training data set is removed and the surrogate model is constructed using the remaining observations. The surrogate model can then be used to predict the outputs at the locations in the removed subset. Since the true output values are known, the prediction errors can be computed. This process repeats until all data points have been removed at some point in the procedure.

The sum of the prediction errors from all iterations is used to construct a cross-validation error function,

$$\mathcal{E}_{cv}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)} - \hat{f}^{-\zeta(i)}(\mathbf{x}^{(i)}, \mathbf{w})\right]^2. \tag{3.7}$$

The model parameters are then obtained by also minimizing Equation 3.7.

3.2.2.1 Kriging

Kriging (Sacks et al. (1989)) is a popular surrogate modelling approach for facilitating engineering design studies that are based on deterministic computational responses. It is particularly useful for predicting responses from numerical simulations as it only makes assumptions about the smoothness of the response surface but not the degrees of freedom. The Kriging formulation also lends itself to the derivation of a prediction error metric which is useful for gauging the quality of the surrogate when high simulation costs make it difficult to do so otherwise (Keane and Nair (2005); Forrester et al. (2008); Sóbester et al. (2014)).

Consider a simple surrogate model of the response of interest, f, involving a m-order polynomial,

$$\hat{f}(x, m, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_m x^m = \sum_{i=0}^{m} w_i x^i. \tag{3.8}$$

This single-variable model can be extended to n variables using a linear combination of basis functions,

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} w_i \psi^{(i)}, \tag{3.9}$$

where ψ is defined as the set of terms of order not greater than m and consists of all possible combinations of variables.

In terms of describing continuous physical responses, radial basis functions are more suitable. A radial basis function can be expressed as,

$$\hat{f}(\mathbf{x}) = \mathbf{w}^T\boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x} - \mathbf{c}^{(i)}\|) = \sum_{i=1}^{n_c} w_i \psi(\mathbf{r}^{(i)}), \tag{3.10}$$

where $\mathbf{c}^{(i)}$ denote the basis function centres. ψ thus depends on the distance between a location of interest $\mathbf{x}$ and these centres. ψ can take on many forms, such as linear ($\psi(r) = r$), cubic($\psi(r) = r^3$), and Gaussian ($\psi(r) = e^{-r^2/(2\sigma^2)}$), among others.

The basis function for the Kriging method is expressed as,

$$\psi^{(i)} = \exp\left(-\sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j|^{p_j} \right). \tag{3.11}$$

The Kriging formulation can be considered to be a more general form of the Gaussian basis function, where the parameter $1/(\sigma)^2$ and the exponent or order 2 have been replaced with hyperparameters θ_j and p_j. θ_j and p_j control the spread of the basis function and the degree of smoothness around the basis centres respectively. The fundamental assumption for using surrogate models is that the response at two locations which are close together in the design space will have similar values, which is true for most engineering functions. Kriging is one of the few surrogate types that is the least reliant on this assumption, due to the presence of the hyperparameters.

The similarity in values, or correlation, between points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(l)}$ can be expressed as a basis function,

$$\text{cor}[f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(l)})] = \exp\Big(-\sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j^{(l)}| \Big). \tag{3.12}$$

A correlation matrix, $\mathbf{\Psi}$, can be formed from all possible pairs of points in the training data set,

$$\mathbf{\Psi} = \begin{pmatrix} \text{cor}[f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(1)})] & \cdots & \text{cor}[f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(n)})] \\ \vdots & \ddots & \vdots \\ \text{cor}[f(\mathbf{x}^{(n)}), f(\mathbf{x}^{(1)})] & \cdots & \text{cor}[f(\mathbf{x}^{(n)}), f(\mathbf{x}^{(n)})] \end{pmatrix}. \tag{3.13}$$

Let $\mathbf{1}\hat{\mu}$ be the mean of the training data set, where $\mathbf{1}$ is an $n \times 1$ vector of ones. A covariance matrix is thus,

$$\text{cov}(f) = \sigma^2 \mathbf{\Psi}. \tag{3.14}$$

A maximum likelihood approach can be used to find the values of θ_j and p_j which best return $f(\mathbf{x})$. At this point two types of likelihood functions can be derived based on our assumptions on the nature of $f(\mathbf{x})$. If we take $f(\mathbf{x})$ to have no error (from noise, modelling inaccuracies, or other sources), we can build a model that interpolates the training data set.

The maximum likelihood function for this approach can be written as,

$$\frac{1}{(2\pi\sigma^2)^{n/2}|\mathbf{\Psi}|^{1/2}}\exp\left[-\frac{(\mathbf{f}-\mathbf{1}\mu)^T\mathbf{\Psi}(\mathbf{f}-\mathbf{1}\mu)}{2\sigma^2}\right], \tag{3.15}$$

or in natural logarithmic form,

$$-\frac{n}{2}\ln(2\pi)-\frac{n}{2}\ln(\sigma^2)-\frac{1}{2}\ln|\mathbf{\Psi}|-\frac{(\mathbf{f}-\mathbf{1}\mu)^T\mathbf{\Psi}^{-1}(\mathbf{f}-\mathbf{1}\mu)}{2\sigma^2}. \tag{3.16}$$

Maximum likelihood estimates of μ and σ^2 can then be obtained by setting the partial derivatives of Equation 3.16 to zero, giving,

$$\hat{\mu}=\frac{\mathbf{1}^T\mathbf{\Psi}^{-1}\mathbf{f}}{\mathbf{1}^T\mathbf{\Psi}^{-1}\mathbf{1}}, \tag{3.17}$$

$$\hat{\sigma}^2=\frac{(\mathbf{f}-\mathbf{1}\mu)^T\mathbf{\Psi}^{-1}(\mathbf{f}-\mathbf{1}\mu)}{n}. \tag{3.18}$$

Substituting Equation 3.16 and Equation 3.18 back into Equation 3.15 gives an expression known as the concentrated ln-likelihood function,

$$-\frac{n}{2}\ln(\hat{\sigma}^2)-\frac{1}{2}\ln|\mathbf{\Psi}|. \tag{3.19}$$

Equation 3.19 is maximised to give the optimum hyperparameter values. As this function can be evaluated quickly, numerical optimisation techniques that directly search the hyperparameter space can be employed. In the current work, the OPTIMATv2 optimisation package that is used to run the surrogate-based optimisation processes uses a hybridised Particle Swarm Optimisation algorithm to maximise the hyperparameters.

3.2.2.2 Co-Kriging

When there are multiple analysis models of varying accuracies and evaluation costs, a multi-fidelity surrogate modelling approach can be used to make full use of the available data. In this work, the auto-regressive, multi-fidelity extension of Kriging, called Co-Kriging, is used (Kennedy and O'Hagan (2000); Forrester et al. (2007)). The auto-regressive model says that observations of the high-fidelity response are correct and that any inaccuracies in the Co-Kriging model are caused entirely by the low-fidelity response. This is equivalent to approximating the high-fidelity response by adding a scaled Gaussian process of the low-fidelity response to a Gaussian process of the difference between the two data sets. The reader is referred to Park et al. (2017) for a survey of multi-fidelity surrogate modelling methods, in which the Co-Kriging approach used here is classified as a Bayesian discrepancy framework with a non-informative prior.

As with any surrogate modelling process, we first start with a training data set that has been generated by a design of experiments. The difference for multi-fidelity surrogate models is that the data set would now have distinct groups of sample locations pertaining to simulations of different fidelities. Assuming that there are two simulation codes, the training data set would be

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_c \\ \mathbf{X}_e \end{pmatrix} = \begin{pmatrix} \mathbf{x}_c^{(1)} \\ \vdots \\ \mathbf{x}_c^{(n_c)} \\ \mathbf{x}_e^{(1)} \\ \vdots \\ \mathbf{x}_e^{(n_e)} \end{pmatrix} \tag{3.20}$$

where the subscripts e and c denote the expensive and cheap evaluations respectively.

Recall that Kriging considers the outputs at each of the sampling locations as realizations of stochastic processes. Using F to denote a randomly-distributed representation of the outputs, the evaluated sampling plan can be expressed as,

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_c(\mathbf{X}_c) \\ \mathbf{F}_e(\mathbf{X}_e) \end{pmatrix} = \begin{pmatrix} \mathbf{F}_c(\mathbf{x}_c^{(1)}) \\ \vdots \\ \mathbf{F}_c(\mathbf{x}_c^{(n_c)}) \\ \mathbf{F}_e(\mathbf{x}_e^{(1)}) \\ \vdots \\ \mathbf{F}_e(\mathbf{x}_e^{(n_e)}) \end{pmatrix}. \tag{3.21}$$

Consider the scenario where the cheap and expensive sampling plans share a common point $\mathbf{x}^{(i)}$. What would be a suitable way to utilise both of the random fields $\mathbf{F}_c(\mathbf{x}^{(i)})$ and $\mathbf{F}_e(\mathbf{x}^{(i)})$ in our calculations? A rational assumption that addresses this which would also simplify subsequent derivations is to let $\text{cov}\{Y_e(\mathbf{x}^{(i)}), Y_c(\mathbf{x})|Y_c(\mathbf{x})^{(i)}\} = 0, \forall \mathbf{x} \neq \mathbf{x}^{(i)}$. This assumption is part of the auto-regressive model from Kennedy and O'Hagan (2000). Here we treat the expensive evaluation at this point as the 'truth' and say that it cannot learn anything more from the cheap evaluation.

Continuing with the auto-regressive model, an approximation of the expensive analysis is represented by the product of a Gaussian process of the expensive response, Z_c, with a scaling factor ρ, and a Gaussian process of the difference between the cheap and expensive data, Z_d,

$$Z_e(\mathbf{x}) = \rho Z_c(\mathbf{x}) + Z_d(\mathbf{x}). \tag{3.22}$$

In the same vein as Equation 3.14, the covariance matrix between the cheap and expensive data can be expressed as,

$$C = \begin{pmatrix} \sigma_c^2 \mathbf{\Psi}_c(\mathbf{X}_c, \mathbf{X}_c) & \rho\sigma_c^2 \mathbf{\Psi}_c(\mathbf{X}_c, \mathbf{X}_e) \\ \rho\sigma_c^2 \mathbf{\Psi}_c(\mathbf{X}_e, \mathbf{X}_c) & \rho^2\sigma_c^2 \mathbf{\Psi}_c(\mathbf{X}_e, \mathbf{X}_e) + \sigma_d^2 \mathbf{\Psi}_d(vX_e, \mathbf{X}_e) \end{pmatrix}, \tag{3.23}$$

where $\mathbf{\Psi}_c$ and $\mathbf{\Psi}_d$ are in the same form in Equation 3.12.

With an extra correlation matrix there are now five hyperparameters to estimate: θ_c, θ_d, $\mathbf{p}_c$, $\mathbf{p}_d$ and ρ. As the cheap and expensive data are considered to be independent, the maximum likelihood estimations of the hyperparameters belonging to the cheap model can be computed using the same formulation as seen in Equation 3.15 to Equation 3.19.

The concentrated ln-likelihood function that is to be maximised in this process is,

$$-\frac{n_c}{2}ln(\hat{\sigma}_c^2) - \frac{1}{2}ln|\mathbf{\Psi}_c(\mathbf{X}_c, \mathbf{X}_c)|. \tag{3.24}$$

To estimate θ_d, $\mathbf{p}_d$ and ρ, we first define a difference model,

$$\mathbf{d} = \mathbf{f}_e - \rho\mathbf{f}_c(\mathbf{X}_e). \tag{3.25}$$

Equations 3.17 and 3.18 can then be re-used to calculate $\hat{\mu}_d$ and $\hat{\sigma}_d$ by replacing $\mathbf{f}$ with $\mathbf{d}$.

$\hat{\rho}_d$ can then be calculated by maximising,

$$-\frac{n_e}{2}ln(\hat{\sigma}_d^2) - \frac{1}{2}ln|\mathbf{\Psi}_c(\mathbf{X}_c, \mathbf{X}_c)|. \tag{3.26}$$

3.2.2.3 Practical Considerations

One complication from having simulation codes of different fidelities is what the ratio of cheap to expensive runs should be. This is governed by factors such as the cost ratio and how well-correlated the cheap data is to the expensive data. Toal (2015) investigated these factors regarding the appropriate use of Co-Kriging for optimisation. The results showed that the level of correlation between the cheap and expensive data sets significantly influences the accuracy of the multi-fidelity surrogate model, and this is especially true when the number of cheap points are high.

If the correlation is poor, adding more cheap points may decrease the accuracy of the Co-Kriging model. Adding expensive points could improve the accuracy of the Co-Kriging model, but a single-fidelity Kriging model of equivalent cost could be more accurate by comparison. Thus, the construction of Co-Kriging models on badly correlated data sets is not recommended. Finally, if the cheap and expensive evaluations are of similar cost (a 2:1 cost ratio was considered similar in this case), the use of cheap evaluations can only be recommended if the correlation is, again, high.

These observations have lead to the establishment of a set of guidelines for constructing Co-Kriging models which were considered in the process of designing the experiments in this report. The guidelines are:

1. The r^2 correlation between the cheap and expensive data should be > 0.9.

2. $0.1 < f_r < 0.8$, where f_r is $\dfrac{\text{Budget for cheap evaluations}}{\text{Total evaluation budget}}$.
3. The number of cheap data should always be slightly more than the number of expensive data. A lower bound for this condition was defined to be $f_r > \frac{1.75}{1+\frac{1}{C_r}}$, where C_r is the cost ratio of the cheap to expensive simulation codes.

For example, if the optimisation problem has 8 variables, we can set the budget to $10d$, equal to 80 expensive evaluations. Assuming $C_r = 0.2$, $\frac{1.75}{1+\frac{1}{C_r}}$ returns 0.2916. Thus, guidelines 2 and 3 recommends having a f_r value that is approximately 30% to 80% of the total available budget.

Of course, these are suggestions on what a Co-Kriging model should not be, and it is not quite obvious what the optimum parameters are as the range of f_r is quite large. An argument can be made that the higher the r^2 correlation, the higher the f_r should be, as there would be more evaluations available for surrogate model construction. In this example, $f_r = 0.7$ would be appropriate if $r^2 = 0.99$. $f_r = 0.7$ of the total budget of 80 expensive simulations would give a budget for the cheap simulations that is equivalent to 56 expensive simulations. The number of cheap simulations can then be computed by dividing this with C_r, giving $n_c = 280$, while $n_e = 24$.

Note that the r^2 correlations are often not known a priori. Furthermore, there has yet to be any studies on what budget divisions are most suitable if the training data set is a mixture of well-correlated and poorly-correlated responses. As such, some divergence from the presented guidelines is unavoidable but what should be followed most faithfully is the recommendation on having more cheap data than expensive data.

After establishing the number of cheap and expensive simulations, the next question is how should one go about choosing the points themselves? Here the exchange algorithm from Forrester et al. (2008) is used which selects a subset of points from the cheap data set that minimizes the Φ_q criterion from Morris and Mitchell (1995) through an iterative process. The Φ_q criterion ranks competing sampling plans, where a lower value signifies better space-filling properties,

$$\Phi_q(\mathbf{X}) = \left(\sum_{j=1}^{n} J_j d_j^{-q} \right)^{1/q}, \tag{3.27}$$

where n is the number of sampling points, $\{d_j, j = 1, ..., n\}$ is the list of unique distances between all possible pairs of sampling points, $\{J_j, j = 1, ..., n\}$ is the list of the number of pairs of sampling points separated by the distances in d_j, and q is an exponent which can be chosen to control the desired amount of 'space-filling-ness'. The common points

between the cheap and expensive datasets are then used to construct the difference model according to Equation 3.25.

3.2.3 Searching the Surrogate

As surrogate models are approximations of the true response, it would be useful to have a mechanism that can continually improve their accuracy throughout the optimization process. The most straightforward way to do this is to add more data to the training data set. The cost of each simulation, however, forces the optimizer to be more selective when choosing new locations to sample. The error estimation capabilities of the Kriging and Co-Kriging formulations provide a foundation from which multiple infill criteria can be rapidly derived to guide this selection. The locations at which these update criteria are at their extrema represent candidate locations for further simulations. In the current work, the OPTIMATv2 optimisation package that is used to run the surrogate-based optimisation processes uses a NSGA-II algorithm to search for these extrema.

3.2.3.1 Predicted-Based Exploitation

For Kriging models, once the correlation matrix $\mathbf{\Psi}$ is computed, the surrogate model now has the capability to make predictions at un-sampled points in the design space. $\mathbf{\Psi}$ has been trained to be consistent with the training data set, but now it also has to be consistent with the predictions at new points. A new point can be added to the training data set, giving $\tilde{\mathbf{f}} = [\mathbf{f}, f_{new}]^T$.

A vector of correlations between this new point and the rest of the points in the data set can be written as,

$$\tilde{\psi} = \begin{pmatrix} \mathrm{cor}[f(\mathbf{x}^{(1)}), f_{new}(\mathbf{x}_{new})] \\ \vdots \\ \mathrm{cor}[f(\mathbf{x}^{(n)}), f_{new}(\mathbf{x}_{new})] \end{pmatrix} = \begin{pmatrix} \psi^{(1)} \\ \vdots \\ \psi^{(n)} \end{pmatrix}. \tag{3.28}$$

The correlation matrix for this augmented data set can be expressed as,

$$\tilde{\mathbf{\Psi}} = \begin{pmatrix} \mathbf{\Psi} & \psi \\ \psi^T & 1 \end{pmatrix}. \tag{3.29}$$

Substituting Equation 3.29 into Equation 3.19 gives the augmented ln-likelihood function,

$$-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln|\hat{\mathbf{\Psi}}| - \frac{(\hat{\mathbf{f}} - \mathbf{1}\hat{\mu})^T\hat{\mathbf{\Psi}}^{-1}(\hat{\mathbf{f}} - \mathbf{1}\hat{\mu})}{2\hat{\sigma}^2}. \tag{3.30}$$

It is only necessary to include the last term in the maximization as it is the only term that depends on f_{new}.

To maximise this function, a partitioned inverse method is applied and the resulting function can then be differentiated with respect to f_{new} to give,

$$\left(-\frac{1}{\sigma^2(1-\psi^T\mathbf{\Psi}^{-1}\psi)}\right)(f_{new}-\mu) + \left(\frac{\psi^T\mathbf{\Psi}^{-1}(\mathbf{f}-\mathbf{1}\mu)}{\sigma(1-\psi^T\mathbf{\Psi}^{-1}\psi)}\right) = 0. \tag{3.31}$$

The Kriging predictor for f_{new} is thus,

$$\hat{f}(\mathbf{x}_{new}) = \mu + \psi^T\Psi{-}1(\mathbf{f}-\mathbf{1}\mu). \tag{3.32}$$

The Co-Kriging predictor can be similarly derived, and is expressed as,

$$\hat{f}_e(\mathbf{x}) = \hat{\mu} + c^T C^{-1}(\mathbf{f}-\mathbf{1}\hat{\mu}), \tag{3.33}$$

where the correlation terms in the Kriging predictor have been replaced with covariance terms.

Updating the Kriging model at the predicted optimum would allow the optimizer to quickly converge to an optimum value. This approach heavily favours exploitation which is useful if the response is expected to be unimodal. However, the optimiser would have a difficult time escaping from a basin of attraction which would not be desirable if better optima are located elsewhere in the design space. Using this metric early on in the optimisation loop when the surrogate model has the lowest accuracy could result in wasted evaluations.

3.2.3.2 Error-Based Exploration

A better infill strategy for early optimization iterations is to sample where the level of uncertainty in the Kriging prediction is high. This is an exploration-based method that,

over several iterations, increases the overall accuracy of the surrogate over the design space.

The mean squared error of a Gaussian-based prediction can be expressed as,

$$s^2(\mathbf{x}_{new}) = \sigma^2\left[1 - \psi^T\mathbf{\Psi}^{-1}\psi + \frac{1 - \mathbf{1}^T\mathbf{\Psi}^{-1}\psi}{\mathbf{1}^T\mathbf{\Psi}^{-1}\mathbf{1}}\right]. \tag{3.34}$$

Employing this strategy for too long, however, is tantamount to using a larger sampling plan. There is also no obvious guide to indicate when one should stop exploring and start exploiting the basins of attraction that have been found.

3.2.3.3 Balanced Exploitation and Exploration

By incorporating the Gaussian-based error from Equation 3.34 into the prediction values, an approach can be developed which allows the predictions to take multiple values from a normal distribution that is centred around the maximum likelihood estimation of $\hat{f}$ with a variance equal to the error. This provides a means to estimate an expected improvement value (Jones et al. (1998)) representing the magnitude of the potential improvement in the objective function at un-sampled locations, i.e. the expected improvement.

If we assume that the improvement is $I = f_{min}(\mathbf{x}) - \hat{f}(\mathbf{x}_{new})$, the probability of achieving this improvement can be calculated using,

$$P[I(\mathbf{x})] = \frac{1}{s\sqrt{2\pi}}\int_{-\infty}^{0}\exp\Big[-\frac{[I - \hat{f}(\mathbf{x}_{new})]^2}{2s^2}\Big]dI, \tag{3.35}$$

$$P[I(\mathbf{x})] = \frac{1}{2}\left[1 + \text{erf}\left(\frac{f_{min} - \hat{f}(\mathbf{x}_{new})}{s\sqrt{2}}\right)\right]. \tag{3.36}$$

The expected improvement is then given by,

$$E[I(\mathbf{x})] = (f_{min} - \hat{f}(\mathbf{x}_{new}))\Phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) + s\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right), \tag{3.37}$$

$$E[I(\mathbf{x})] = (f_{min} - \hat{f}(\mathbf{x}_{new}))\left[\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s\sqrt{2}}\right)\right] + s\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{f_{min} - \hat{f}(\mathbf{x})}{2s^2}\right], \tag{3.38}$$

where Φ and ϕ represent the cumulative distribution function and probability density function respectively, and erf is the error function.

This approach strikes a good balance between exploration and exploitation, but the location of maximum expected improvement can be difficult to find after several update iterations as they become very small.

3.2.3.4 Constrained Search

The probability of feasibility criterion, $P[F(\mathbf{x})]$, represents the probability that the predicted value from the surrogate model of a constraint is greater than its specified upper bound. It is similar to the probability of improvement criterion, $P[I(\mathbf{x})]$, where instead of using the current best design as the dividing point in probability space, we use the constraint limit value $g_{\min}$,

$$P[F(\mathbf{x})] = \frac{1}{\hat{s}\sqrt{2\pi}}\int_0^\infty e^{-(F-\hat{g}(\mathbf{x})^2/(2\hat{s})^2)}\mathrm{d}G, \tag{3.39}$$

where g is the constraint function, F is the measure of feasibility $G(\mathbf{x}) - g_{\min}$, $G(\mathbf{x})$ is a random variable and $\hat{s}$ is the variance of the Kriging model of the objective.

The probability that a new infill point both improves on the current best point and is also feasible is then,

$$P[I(\mathbf{x}) \cap F(\mathbf{x})] = P[I(\mathbf{x})]P[F(\mathbf{x})], \tag{3.40}$$

since these are independent models.

We can also use the probability that a point will be feasible to formulate a constrained expected improvement. We simply multiply $E[I(\mathbf{x})]$ by $P[F(\mathbf{x}) > g_{\min}]$,

$$E[I(\mathbf{x}) \cap F(\mathbf{x})] = E[I(\mathbf{x})]P[F(\mathbf{x})]. \tag{3.41}$$

Note that this search for $E[I(\mathbf{x}) \cap F(\mathbf{x})]$ is unconstrained since the action of the constraint has been merged directly into the objective function.

In effect, the constrained expected improvement falls away at the sample points as in simple expected improvement. Additionally. the constrained expected improvement drops as the constraint boundary is approached but allows for infeasible points to be sampled to improve the overall model quality (for both the objective and the constraint). As more update points are added the search space is widely sampled and the contours of the constrained expected improvement becomes more closely aligned with the actual constraint boundary. This method of computing the expected improvement is used to solve the constrained optimization problems in the current work.

3.3 Multi-Fidelity Surrogate-Based Optimization with Solid and Mid-Surface Meshes

In this section, two case studies are presented to demonstrate the multi-fidelity surrogate modelling approach with solid and mid-surface meshes.

The first case study is an optimization of the CRESCENDO HP compressor casing for mass reduction under stiffness constraints. This case study is a simple 2D problem that allows the benefits of the multi-fidelity approach to be fully grasped through inspection of the resulting response surface plots. The results from this case study also serve as a useful benchmark for the design-in-context case studies in Section 3.4, where it will be shown that the optimum casing design obtained here violates the stiffness constraints in its neighbouring components.

The second case study is an optimization of the CRESCENDO whole engine for mass reduction under stiffness constraints. This case study is used to assess the effectiveness of the multi-fidelity approach for higher-dimensional problems that involve complex, multi-component finite element assemblies.

3.3.1 Case study overview

In the current work, sizing optimization in the manner of modifying the thicknesses of planar structural members is performed. This is a relatively minor type of geometry adjustment but has been shown to be effective in reducing weight in thin-walled structures while preserving much of the stiffness (Taylor (2006)). The design variables in each case study are thus the thicknesses of select structural members in the geometry under consideration. Some care is taken to only select structural members not in the gas path so that the fluid-structure interactions can be neglected.

Although not included as part of the simulation cost, some differences in the time required to perform the finite element pre-processing steps between the solid and mid-surface meshes can be observed. For evaluations of the mid-surface mesh, only a single

discretization is required. Thickness changes can be realized by simply identifying within the simulation deck the shell elements that belong to the mid-surface of a structural member, and rewriting the thickness values of those elements (see Figure 2.4). On the other hand, to materialize the thickness changes in the solid mesh, the original solid geometry had to be updated and re-meshed. The direct modelling tool in Siemens NX is used to apply these thickness changes to the solid CAD model. The subsequent re-meshing step is then carried out with the assumption that the results from the mesh convergence studies performed in Section 2.5.1 remains valid within the scope of geometry changes, so the same mesh settings can be applied. It is thus considerably more expensive to prepare the high-fidelity model.

Since all of the case studies involve a comparison between strategies that employ low-fidelity FEMs in some form, it would be helpful to have a preliminary assessment of the accuracy and cost of these low-fidelity simulations over a design space in order to see how they affect the final optimization results. This can be regarded as an extension of the single-point validation study in Section 2.5.1 but to a uniform distribution of designs.

To this end, a large space-filling DOE of low- and high-fidelity simulations is generated to provide this data for each case study. The size of the space-filling DOE is chosen to be 20 times the problem dimensionality. The accuracy of the low-fidelity simulations is then presented in terms of a Pearson's correlation coefficient, r^2, a root mean square error (RMSE) and a maximum absolute error (MAE). The errors are normalized against the range of observed values. The costs of the low- and high-fidelity simulations are also presented as bootstrapped mean values. These mean costs are then used to compute the cost ratio between the simulation fidelities. It should be noted that the data from these comparisons are for expository purposes; the typical design engineer does not normally have the luxury of time or computational resources to perform so many simulations ahead of his/her optimization study.

3.3.2 Optimization of a High-Pressure Compressor Casing

The first case study involves an optimization of the HP compressor casing (HPCC) from the CRESCENDO engine. Figure 3.2 shows the HPCC geometry. The objective is to minimize the HPCC mass, HPCC.OBJ, within a small relaxation of its stiffness. The design variables are the thicknesses HPCC.VAR1 and HPCC.VAR2. HPCC.VAR1 is the thickness of the flange-rib substructure, and HPCC.VAR2 is the thickness of the rib separating the void box from the combustor region. These structural members were selected as design variables because their thickness modification does not affect the gas path. The values of the design variables for the baseline design are 3 mm. Lower and upper bounds for the design variables are set to $\pm 50\%$ of the baseline values. A total of seven stiffness constraints, HPCC.CON1 to HPCC.CON7, are present in the current study. The upper bounds of the constraints are set to a 1% increase over the

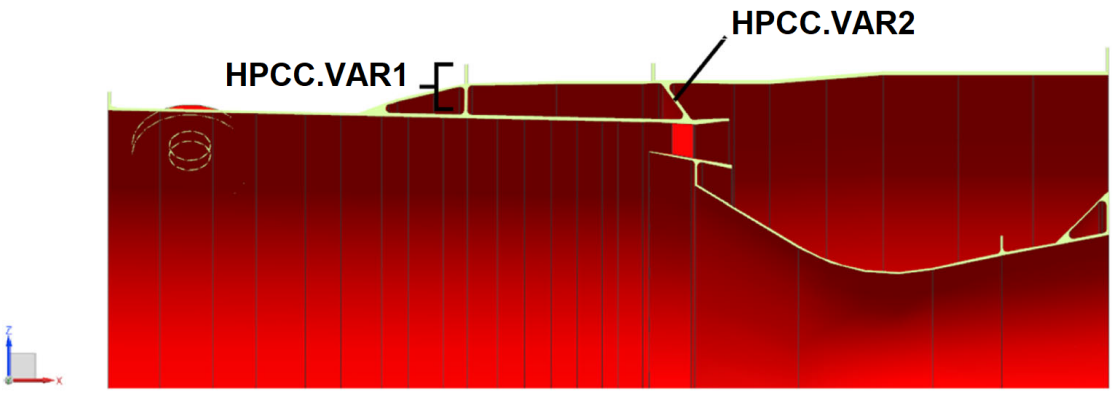


Figure 3.2: Section view of the CRESCENDO HPCC geometry.

maximum radial displacement values of the baseline design. This relaxation in the allowed displacement is used to examine how much weight saving can be achieved if a small compromise in stiffness is allowed.

The mass objective is obtained by taking the product of the CAD model's volume and its material density. Since both the solid and mid-surface simulations use the same CAD model as their starting geometry, their evaluations of the mass are identical. This response is thus omitted from the accuracy assessments. The stiffness constraints are obtained by post-processing the displacement results as described in Section 2.5.1.

The geometry is meshed in both solid and mid-surface form using the converged meshing parameters established in Section 2.5.1. Since the HPCC is an independent local model, it needs to be supplied with free body displacements derived from the results of a global FE model simulation of the whole engine. The results used in this derivation is simply taken from the validation study in Section 2.5.1. The displacements at the interfaces between the HPCC and its neighbouring components were exported as a field into a .csv file. This file is then imported into the solid and mid-surface meshes of the HPCC in isolation and applied to the same location as an enforced displacement boundary condition. Siemens NX automatically interpolates the displacements from the imported field to the nodes on the target face. These boundary conditions are, however, not updated for every HPCC design change to reflect the new engine equilibrium as it would be expensive to do so. There is thus an expectation of some consistency error in the simulation results when compared to whole engine simulations.

A large 40-point space-filling DOE is generated for the preliminary accuracy and cost assessments of the mid-surface mesh. Table 3.1 shows the results of the accuracy assessment. The results show that the mid-surface mesh is a suitable source of low-fidelity information according to the Co-Kriging guidelines stated in Section 3.2.2.2. The r^2 values for all responses are above the recommended 0.9 threshold, with the constraint HPCC.CON1 as the least well-correlated response at a r^2 of 0.9902.

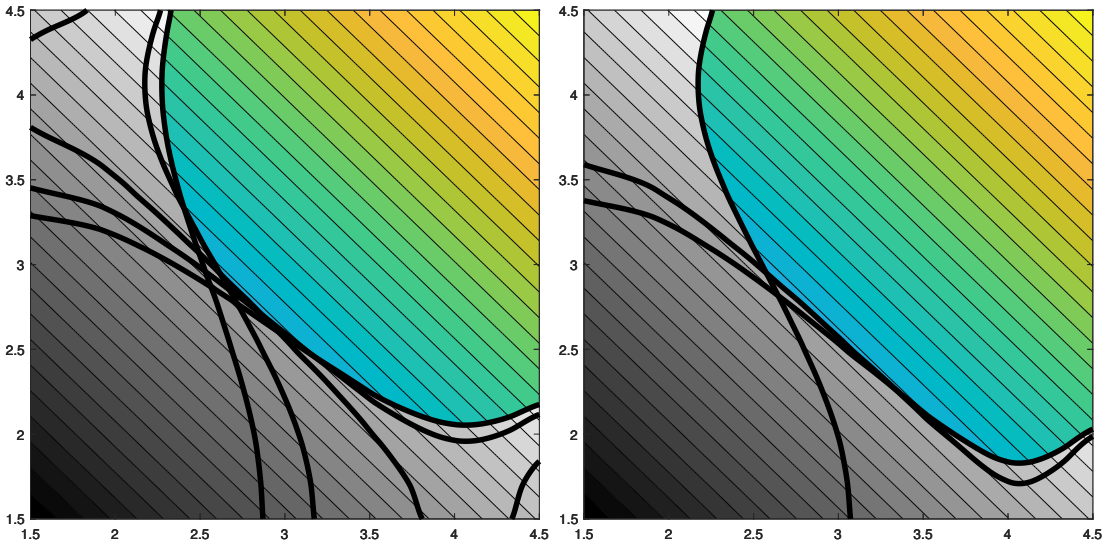
Table 3.1: Comparison of maximum radial displacement values between linear static analyses of solid and shell meshes of the CRESCENDO HPCC.

Output	r^2	% RMSE	% MAE
HPCC.CON1	0.9902	59.92	66.82
HPCC.CON2	0.9910	38.25	47.61
HPCC.CON3	0.9930	63.47	69.71
HPCC.CON4	0.9943	79.45	82.51
HPCC.CON5	0.9944	66.64	71.08
HPCC.CON6	1.0000	63.40	71.31
HPCC.CON7	0.9954	46.52	51.20

The errors are, however, significant enough to discourage solitary use of the mid-surface mesh in a single-fidelity fashion unless the constraint bounds are set to their mid-surface equivalents. Recall that the stiffness constraints in the present case study are percentage increases of responses over the baseline design. The validation study in Section 2.5.1 has shown that the difference between the solid and mid-surface evaluations of these responses for the baseline is non-negligible. It was discovered from a test run that if the constraint bounds in a mid-surface only optimization was set to the values in the solid-only optimization, the entire design space became infeasible.

This requirement seems obvious but is less so when a mixing of bounds is necessary for the design-in-context case studies. It is also clear that changing the bounds is only straightforward if they are stated relative to a given baseline design. Constraint bounds stated in absolute values such as “tip clearance cannot exceed 1 mm” would require a pointless search for the design whose constraints match those values exactly, in order for that design to be evaluated using a mid-surface mesh. Finally, even with the proper bounds, the optimum design that is arrived at with a mid-surface only approach is still subject to the inaccuracies of the surrogate model and thus is not guaranteed to be feasible. The optimum designs would need to be validated using an equivalent solid mesh.

Being a two-variable problem, plots of the response surfaces can be generated for a visual comparison. Figure 3.3 shows the response surfaces of the Kriging models that have been constructed from the large space-filling DOE. Differences in the constraint level curves are noticeable, even with the adjusted constraint bounds. For the solid mesh evaluations, the feasible domain is bounded by the constraints HPCC.CON1, HPCC.CON3, HPCC.CON5, HPCC.CON7 and HPCC.CON8. For the mid-surface mesh evaluations, however, the feasible domain is bounded by the constraints HPCC.CON2, HPCC.CON5, and HPCC.CON7.



(a) Tetrahedral mesh. (b) Mid-surface mesh.

Figure 3.3: Response surface of the CRESCENDO HPCC space-filling DOE.

Table 3.2: Cost comparison between linear static analyses of solid and mid-surface meshes of the CRESCENDO HPCC. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

Mesh	Solution time (hr)	
	μ	95% CI
Solid	2.9124	2.6123 - 3.2517
Mid-surface	0.2695	0.2288 - 0.3001

Table 3.2 shows the results of the cost assessment. The means are used to calculate a cost ratio for converting high-fidelity simulations into an equivalent number of low-fidelity simulations. The cost ratio for this case study is thus 10.81.

Optimization strategies were then devised with these considerations in mind. Three strategies were devised to test the performance of the multi-fidelity approach against reference single-fidelity approaches. Strategy I is a single-fidelity Kriging-based optimization that uses solid HPCC evaluations as data. The optimization is started with a 10-point Latin Hypercube DOE, and in each update iteration a maximum EI point is evaluated. Strategy II is a single-fidelity Kriging-based optimization that uses mid-surface HPCC evaluations as data. The optimization is started with the same 10-point Latin Hypercube DOE as in Strategy I, and in each update iteration a maximum EI point is evaluated. Strategy III is a multi-fidelity Co-Kriging-based optimization that uses both solid and mid-surface HPCC evaluations as high- and low-fidelity data, respectively. The optimization is started with a 10-point low-fidelity DOE equivalent to the DOEs in the first two strategies, and a 3-point high-fidelity DOE that is generated by taking an optimum space-filling subset of the low-fidelity DOE. In each update iteration,

3 maximum EI points are evaluated using the low-fidelity model, as well as 1 imputed maximum EI point that is evaluated using the high-fidelity model.

Table 3.3 summarizes these strategies. A maximum EI value of 0.01 kg was used as the convergence criteria for all optimizations. Each strategy was then repeated with ten differently seeded Latin Hypercube DOEs of the same sizes to obtain some statistics.

Table 3.4 summarizes the optimization results. The average mass reduction achieved by the multi-fidelity approach of Strategy III is on par with the reduction achieved by the expensive solid mesh only approach of Strategy I, but only at 26.64% cost. The multi-fidelity approach was also able to identify the most active constraints, constraints HPCC-CON6 and HPCC-CON7, and stay within their bounds. The mid-surface only approach of Strategy II returns an optimum design with a lower mass, but failed to identify the most active constraints. The constraint violation was discovered only at the end of the optimizations when the optimum designs were validated using an equivalent solid mesh.

Table 3.3: Description of optimization strategies in the CRESCENDO HPCC case study.

Parameter	Strategy I	Strategy II	Strategy III
HF FE model	Solid	Mid-surface	Solid
LF FE model	N/A	N/A	Mid-surface
Surrogate model	Kriging	Kriging	Co-Kriging
DOE method	Latin hypercube	Latin hypercube	Latin hypercube
# HF DOE points	10	10	3
# LF DOE points	N/A	N/A	10
Update method	Maximum EI	Maximum EI	Maximum EI
# HF updates	1	1	1
# LF updates	N/A	N/A	3

Table 3.4: Optimization results for the CRESCENDO HP compressor casing case study.

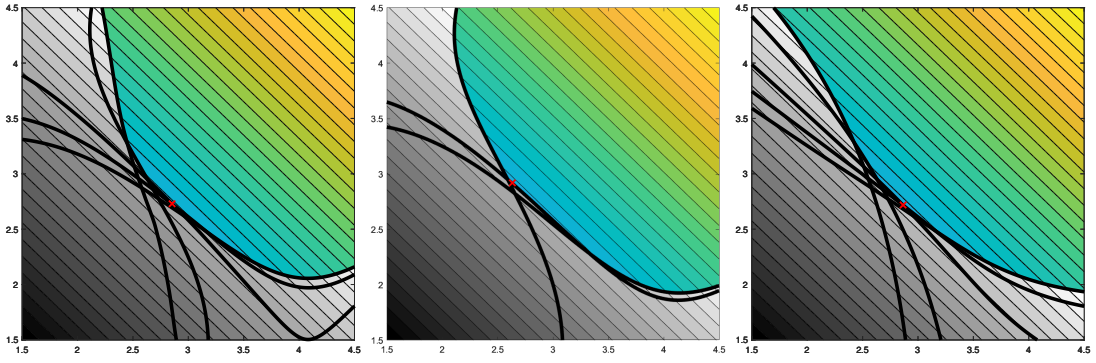
Result	Strategy I	Strategy II	Strategy III
Mass reduction (%)	5.58	5.61	5.55
Infeasibility (%)	0	0.03	0
Relative cost (%)	100	9.25	26.64

Plots of the response surfaces at the end of the optimizations in each strategy can be generated for a visual comparison. Figure 3.4 shows the response surfaces of the surrogate models that have been constructed from all evaluations in the optimizations. The Co-Kriging predictions of the constraint level curves in the multi-fidelity approach of Strategy III are relatively accurate in the region of the optimum design. The same cannot be said for the Kriging predictions in the mid-surface only approach of Strategy II. The surrogate model failed to predict the presence of constraints HPCC-CON1, HPCC-CON3, and HPCC-CON6.

Figure 3.5 plots the search histories for all three strategies. The search histories are presented as a function of the equivalent number of low-fidelity simulations and include the cost of both the initial DOE and any subsequent updates. The cost conversions are performed using the cost ratios in Table 3.2. This conversion is somewhat more complicated for the multi-fidelity approach. The method taken here is to plot the progression of the high-fidelity simulations only but to augment each simulation's cost with the average cost over all simulations in both fidelities. For example, a final data set of 20 solid mesh simulations and 40 mid-surface mesh simulations with a relative cost of 20% would give an overall cost of 140 mid-surface mesh simulations. This overall cost is then spread over the 20 solid mesh simulations, such that each one is treated as 7 mid-surface mesh simulations. The cost savings provided by the Co-Kriging approach can again be clearly observed.

3.3.3 Optimization of a Full Engine

Having illustrated the benefits of the multi-fidelity approach for optimizing an engine component, the approach is now applied to a whole engine optimization study to demonstrate its effectiveness in solving realistic design problems involving a global FEM. The second case study thus involves an optimization of the full CRESCENDO engine. Figure 3.6 shows the engine geometry. The objective is to minimize the whole engine mass,



(a) Strategy I (b) Strategy II (c) Strategy III

Figure 3.4: Response surfaces of the CRESCENDO HPCC optimization problem. These response surfaces correspond to the first repeated run.

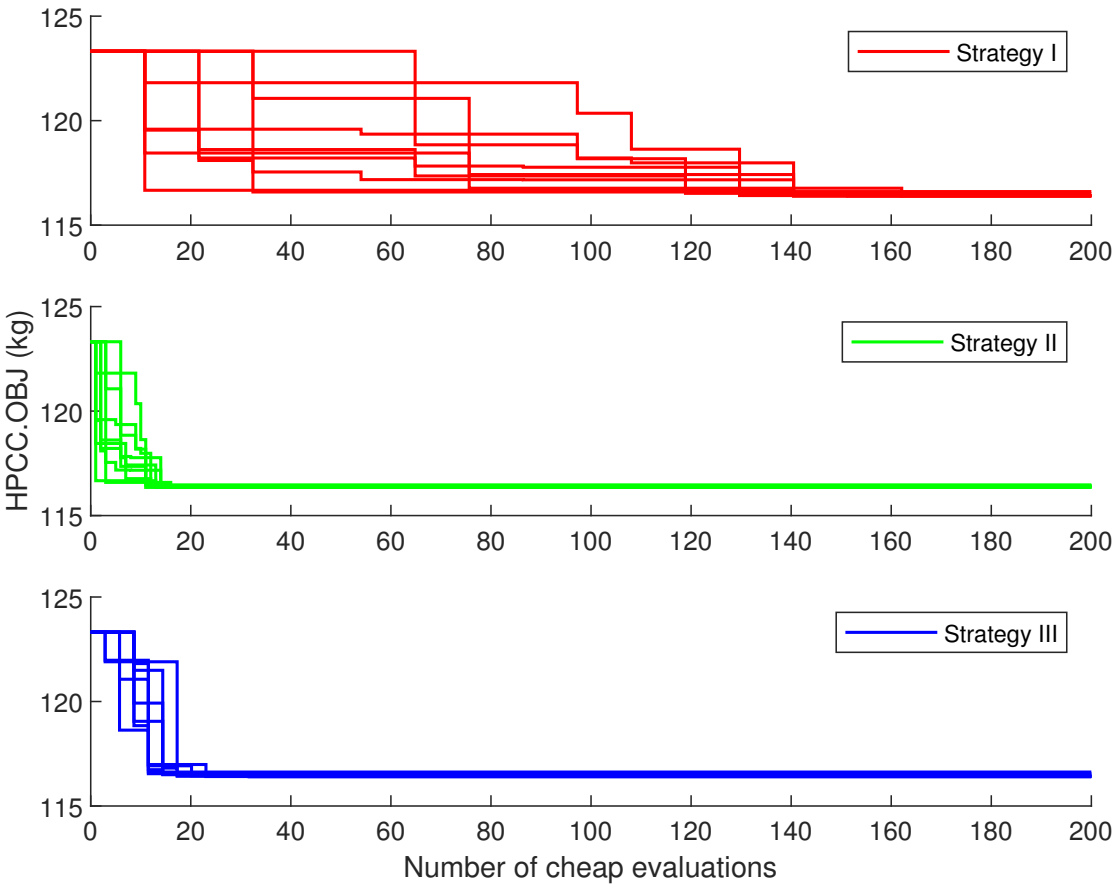


Figure 3.5: Optimization histories for the CRESCENDO HPCC case study.

ENGINE.OBJ, within a small relaxation of its stiffness. There are now eight design variables spread along the engine. They are the thicknesses FC.VAR1 and FC.VAR2 in the fan casing, HPCC.VAR1 and HPCC.VAR2 in the HP compressor casing, HPTC.VAR1 and HPTC.VAR2 in the HP turbine casing, and LPTC.VAR1 and LPTC.VAR2 in the LP turbine casing. The values of the design variables for the baseline design are 3 mm. Lower and upper bounds for the design variables are set to $\pm 50\%$ of the baseline values.

A total of fifteen stiffness constraints are present in the current study. They are the maximum radial displacements FC.CON1 to FC.CON3 in the FC, HPCC.CON1 to HPCC.CON7 in the HPCC, HPTC.CON1 and HPTC.CON2 in the HPTC, and LPTC.CON1 to LPTC.CON3 in the LPTC. The upper bounds of the constraints are set to a 1% increase over the maximum radial displacement values of the baseline design.

The mass objective is again obtained by taking the product of the CAD model's volume and its material density. Its evaluation is again identical between the solid and mid-surface simulations. The stiffness constraints are obtained by post-processing the displacement results as described in Section 2.5.1.

The geometry is meshed in both solid and mid-surface form using the converged meshing

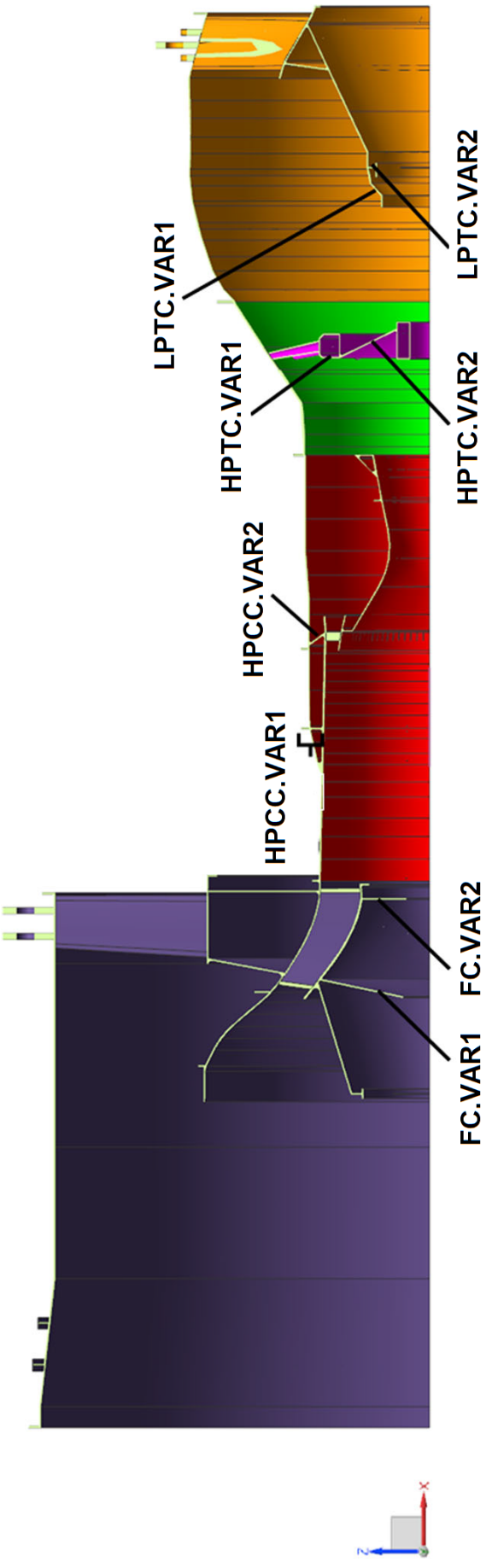


Figure 3.6: Section view of the CRESCENDO engine geometry.

Table 3.5: Comparison of maximum radial displacement outputs between linear static analyses of solid and shell meshes of the CRESCENDO engine

Output	r^2	RMSE	MAE
FC			
FC.CON1	0.9653	0.1545	0.2756
FC.CON2	0.9689	0.0605	0.1419
FC.CON3	0.9740	0.2220	0.3623
HPCC			
HPCC.CON1	0.9720	0.4978	0.6187
HPCC.CON2	0.9747	0.3371	0.4853
HPCC.CON3	0.9853	0.6022	0.7002
HPCC.CON4	0.9945	0.7816	0.8114
HPCC.CON5	0.9946	0.6504	0.6952
HPCC.CON6	0.9944	0.5589	0.6537
HPCC.CON7	0.9881	0.3837	0.4577
HPTC			
HPTC.CON1	0.9714	0.2961	0.4919
HPTC.CON2	0.9775	0.1999	0.2810
LPTC			
LPTC.CON1	0.9733	0.1295	0.2276
LPTC.CON2	0.9699	0.1778	0.2846
LPTC.CON3	0.9663	0.4254	0.5522

parameters established in Section 2.5.1. A large 160-point space-filling DOE is generated for the preliminary accuracy and cost assessments of the mid-surface mesh. Table 3.5 shows the results of the accuracy assessment. The results again show that the mid-surface mesh is a suitable source of low-fidelity information according to the Co-Kriging guidelines stated in Section 3.2.2.2. The r^2 values for all responses are above the recommended 0.9 threshold, with the constraint FC.CON1 as the least well-correlated response at a r^2 of 0.9653.

Compared to the accuracy results in Table 3.1, it can be observed that the r^2 for the HPCC responses for the current case study is markedly lower. This is due to the larger range of response values that are possible because of the increased number of design variables. The errors are again significant enough to discourage solitary use of the mid-surface mesh in a single-fidelity fashion unless the constraint bounds are set to their mid-surface equivalents.

Table 3.6 shows the results of the cost assessment. The means are used to calculate a cost ratio for converting high-fidelity simulations into an equivalent number of low-fidelity

simulations. The cost ratio for this case study is thus 10.18.

Three strategies were devised to test the performance of the multi-fidelity approach against reference single-fidelity approaches. Strategy I is a single-fidelity Kriging-based optimization that uses solid engine evaluations as data. The optimization is started with a 40-point Latin Hypercube DOE, and in each update iteration a maximum EI point is evaluated. Strategy II is a single-fidelity Kriging-based optimization that uses mid-surface engine evaluations as data. The optimization is started with the same 40-point Latin Hypercube DOE as in Strategy I, and in each update iteration a maximum EI point is evaluated. Strategy III is a multi-fidelity Co-Kriging-based optimization that uses both solid and mid-surface engine evaluations as high- and low-fidelity data, respectively. The optimization is started with a 40-point low-fidelity DOE equivalent to the DOEs in the first two strategies, and a 20-point high-fidelity DOE that is generated by taking an optimum space-filling subset of the low-fidelity DOE. In each update iteration, 3 maximum EI points are evaluated used the low-fidelity model, as well as 1 imputed maximum EI point that is evaluated using the high-fidelity model.

Table 3.7 summarizes these strategies. A maximum EI value of 0.01 kg was used as the convergence criteria for all optimizations. Each strategy was then repeated with ten differently seeded Latin Hypercube DOEs of the same sizes to obtain some statistics.

Table 3.8 summarizes the optimization results. The average mass reduction achieved by the multi-fidelity approach of Strategy III is on par with the reduction achieved by the expensive solid mesh only approach of Strategy I, but only at 23.86% cost. The multi-fidelity approach was also able to identify the most active constraints, constraints FC.CON1, FC.CON2 and FC.CON3, and stay within their bounds. The mid-surface only approach of Strategy II returns an optimum design with a lower mass, but failed to identify the most active constraints. The constraint violation was discovered only at the end of the optimizations when the optimum designs were validated using an equivalent solid mesh.

Figure 3.7 plots the search histories for all three strategies. The search histories are presented as a function of the equivalent number of low-fidelity simulations and include the cost of both the initial DOE and any subsequent updates. The cost conversions are

Table 3.6: Cost comparison between linear static analyses of solid and mid-surface meshes of the CRESCENDO engine. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

FE model	Solution time (hr)	
	μ	95% CI
Solid	8.3999	7.2801 - 9.0602
Shell	0.8252	0.5867 - 1.1167

Table 3.7: Description of optimization strategies used in the CRESCENDO engine case study.

Parameter	Strategy I	Strategy II	Strategy III
HF FE model	Solid	Mid-surface	Solid
LF FE model	N/A	N/A	Mid-surface
Surrogate model	Kriging	Kriging	Co-Kriging
DOE method	Latin hypercube	Latin hypercube	Latin hypercube
# HF DOE points	40	40	20
# LF DOE points	N/A	N/A	40
Update method	Maximum EI	Maximum EI	Maximum EI
# HF updates	1	1	1
# LF updates	N/A	N/A	3

performed using the cost ratios in Table 3.6. This conversion follows the method used in the HP compressor casing case study. The cost savings provided by the Co-Kriging approach can again be clearly observed.

3.4 Multi-Fidelity Surrogate-Based Optimization with Embedded Global Finite Element Models

The utility of mid-surface meshes within a multi-fidelity surrogate modelling approach has been demonstrated. The main conclusions are that mid-surface meshes on their own are moderately accurate with respect to solid meshes and can be used to predict the general trend of displacement-based responses. If the results from these simulations are used to train a single-fidelity surrogate model, and the surrogate model is used to predict design constraint values, small constraint violations can still occur at the predicted optimum. The accuracy of these constraint predictions and consequently the

Table 3.8: Optimization results for the CRESCENDO engine case study.

Result	Strategy I	Strategy II	Strategy III
Mass reduction (%)	1.84	1.98	1.80
Infeasibility (%)	0	0.23	0
Relative cost (%)	100	9.82	23.86

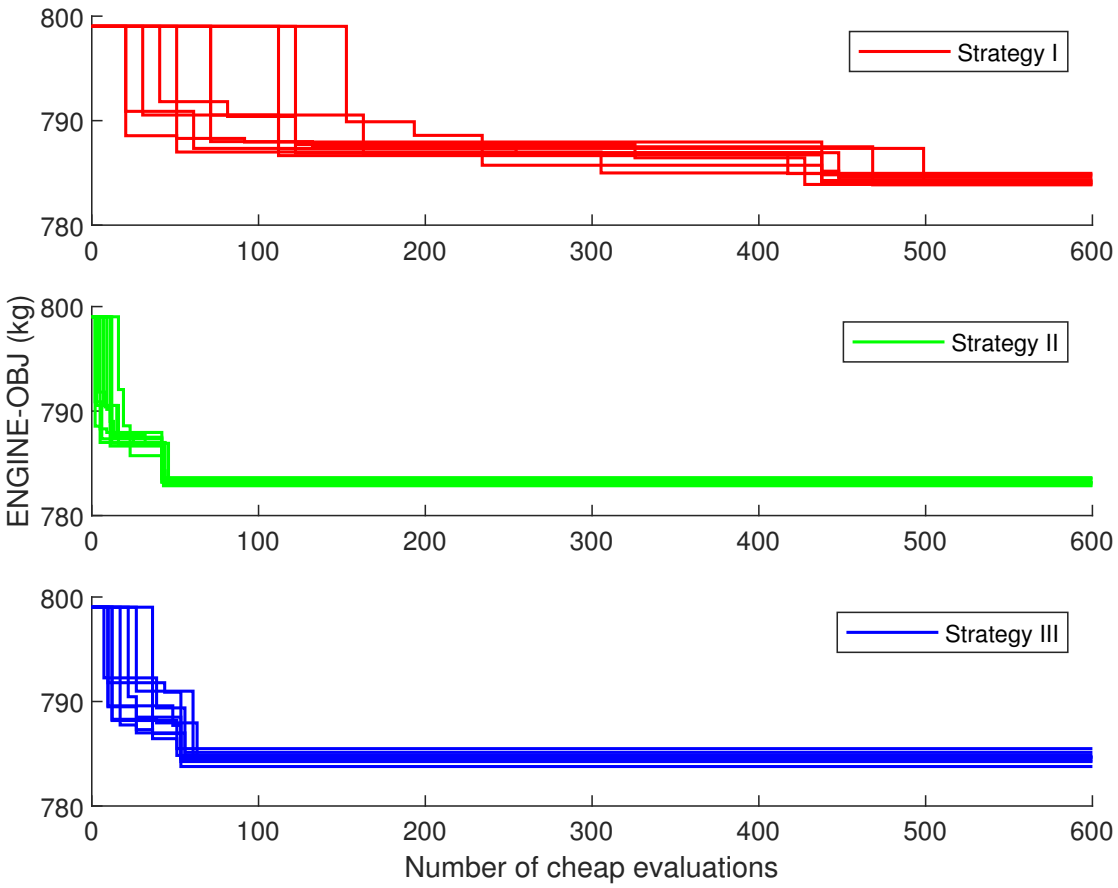


Figure 3.7: Optimization histories for the CRESCENDO engine case study.

reliability of these optimizations, however, can be greatly increased with a multi-fidelity approach employing both solid and mid-surface meshes.

These insights will now be used to devise a design-in-context strategy whereby engine components are optimized with respect to both local and non-local requirements. This is achieved by integrating a high fidelity mesh of a component of interest into global FE models of varying fidelity to create what will be referred to as an embedded global FE model. The motivation for this is to provide component designers with the ability to quickly infer the impact of their design decisions on the rest of the engine. The benefits of democratizing this system-level understanding also transfers to realistic design environments well.

For example, if it was discovered that design changes in one component significantly affects the structural behaviour in another, the design teams responsible for the two components can choose to collaborate more tightly and trade-off with each other in order to deliver performance improvements that would otherwise be impossible if the design activities were carried out in silos. In another scenario, requirements that are prescribed by systems engineers to prevent incompatibilities between concurrent component design

work early on in the design process are often not optimal. Results from Design-in-Context studies may then be used as a quantitative leverage by component designers to negotiate for better requirements.

In this section, two case studies are presented to demonstrate the properties of the design-in-context approach. The first case study is an optimization of the CRESCENDO HPCC for mass reduction under stiffness constraints in the whole engine. This case study is a simple 2D problem that allows the benefits of the design-in-context approach to be fully grasped through inspection of the resulting response surface plots. The second case study is a multi-fidelity version of the first case study that uses Co-Kriging models to improve the prediction of the global constraints.

3.4.1 Optimization of a High-Pressure Compressor Casing with Global Constraints

The first case study involves an optimization of the CRESCENDO HPCC as used in Section 3.3.2. The objective is to minimize the HPCC mass, HPCC.OBJ, within a small relaxation of the stiffness of the whole engine. The design variables are the thicknesses HPCC.VAR1 and HPCC.VAR2. The values of the design variables for the baseline design are 3 mm. Lower and upper bounds for the design variables are set to ±50% of the baseline values. A total of fifteen stiffness constraints are present in the current study. They are the maximum radial displacements FC.CON1 to FC.CON3 in the FC, HPCC.CON1 to HPCC.CON7 in the HPCC, HPTC.CON1 and HPTC.CON2 in the HPTC, and LPTC.CON1 to LPTC.CON3 in the LPTC. The upper bounds of the constraints are set to a 1% increase over the maximum radial displacement values of the baseline design.

A global FE model is constructed by inserting a solid HPCC mesh into the whole engine mid-surface mesh. The embedded global FE model is evaluated over the large 40-point space-filling DOE that was generated in Section 3.3.2. Table 3.9 shows the results of the cost assessment. The means are used to calculate a cost ratio for converting high-fidelity simulations into an equivalent number of low-fidelity simulations. The cost ratio for this case study is thus 1.37.

Three strategies were devised to test the performance of the design-in-context approach against reference approaches. Strategy I is a single-fidelity Kriging-based optimization that uses solid engine evaluations as data. The optimization is started with a 10-point Latin Hypercube DOE, and in each update iteration a maximum EI point is evaluated. Strategy II is a single-fidelity Kriging-based optimization that uses solid HPCC evaluations as data. The optimization is started with the same 10-point Latin Hypercube DOE as in Strategy I, and in each update iteration a maximum EI point is evaluated. Strategy III is a single-fidelity Kriging-based optimization that uses embedded global

Table 3.9: Cost comparisons between linear static analyses of an embedded global FE model containing a solid HPCC mesh, against a solid HPCC mesh in isolation. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

Mesh	Solution time (hr)	
	μ	95% CI
Embedded engine	3.9855	3.4232 - 4.3789
Solid HPCC	2.9124	2.6123 - 3.2517

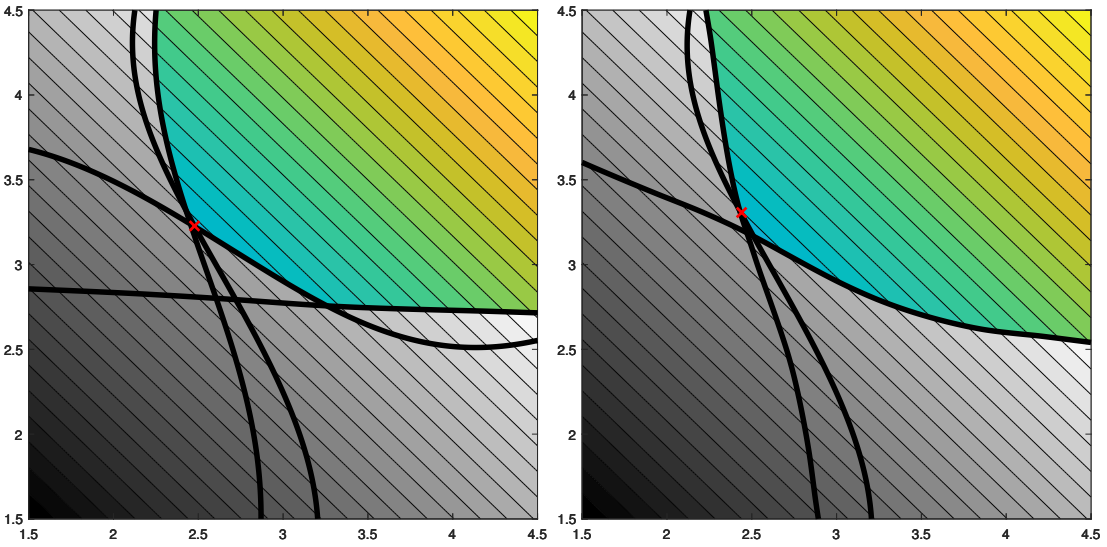
Table 3.10: Description of optimization strategies in the CRESCENDO HPCC design-in-context case study.

Parameter	Strategy I	Strategy II	Strategy III
FE model	Solid engine	Mid-surface HPCC	Embedded engine
Surrogate model	Kriging	Kriging	Kriging
DOE method	Latin hypercube	Latin hypercube	Latin hypercube
# DOE points	10	10	10
Update method	Maximum EI	Maximum EI	Maximum EI
# updates	1	1	1

model evaluations as data, where the solid mesh is of the HPCC in situ. The optimization is started with the same 10-point DOE used in the first two strategies, and in each update iteration a maximum EI point is evaluated.

Table 3.10 summarizes these strategies. A maximum EI value of 0.01 kg was used as the convergence criteria for all optimizations. Each strategy was then repeated with ten differently seeded Latin Hypercube DOEs of the same sizes to obtain some statistics.

Table 3.11 summarizes the optimization results. The average mass reduction achieved by the Design-in-Context approach of Strategy III is on par with the reduction achieved by the expensive solid mesh only approach of Strategy I, but only at 33.32% cost. The Design-in-Context approach was also able to identify the most active constraint HPTC.CON2, a constraint that belongs to the neighbouring HP turbine casing, but did not manage to stay within its bounds. Slight inaccuracies in the Kriging prediction lead to a maximum constraint violation of 0.18%. The Design-in-Isolation approach of Strategy II returns an optimum design with a lower mass, but failed to account for the global constraints because its FEM does not provide that information. The constraint



(a) Strategy I (b) Strategy III

Figure 3.8: Response surfaces of the CRESCENDO HPCC design-in-context optimization problem. These response surfaces correspond to the first repeated run.

violation was discovered only at the end of the optimizations when the optimum designs were validated using a solid mesh of the whole engine.

Plots of the response surfaces at the end of the optimizations in each strategy can be generated for a visual comparison. Figure 3.8 shows the response surfaces of the surrogate models that have been constructed from all evaluations in Strategy I and Strategy III. The response surface for Strategy II can be seen in Figure 3.4a.

Figure 3.9 plots the search histories for all three strategies. The search histories are presented as a function of the equivalent number of low-fidelity simulations and include the cost of both the initial DOE and any subsequent updates. The cost conversions are performed using the cost ratios in Table 3.2 and Table 3.6, where the cost of simulating the solid HPCC in isolation is taken as the reference low-fidelity simulation cost. This conversion follows the method used in the HP compressor casing case study. The cost savings provided by the Design-in-Context approach can be clearly observed.

Table 3.11: Optimization results for the CRESCENDO HPCC design-in-context case study..

Result	Strategy I	Strategy II	Strategy III
Mass reduction (%)	5.34	5.58	5.35
Infeasibility (%)	0	2.86	0.18
Relative cost (%)	100	22.27	33.32

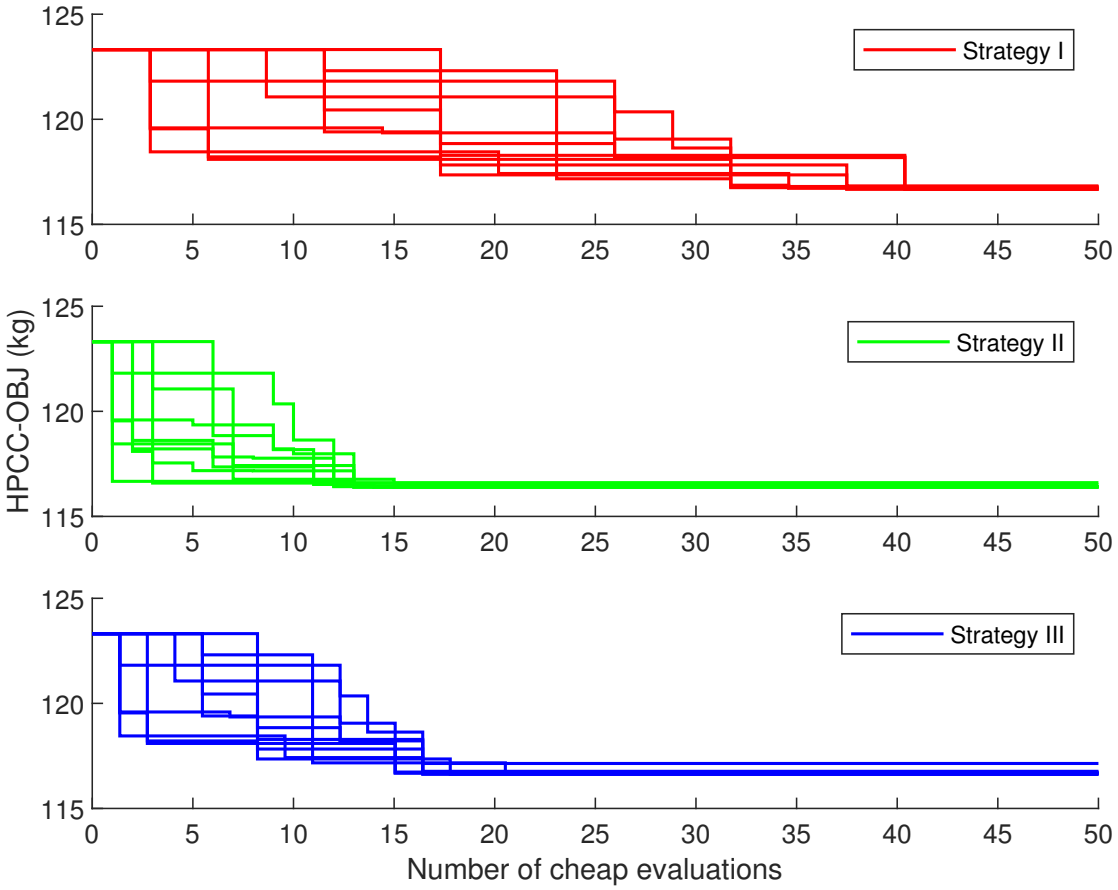


Figure 3.9: Optimization histories for the CRESCENDO HP compressor casing with global constraints case study.

3.4.2 Multi-Fidelity Optimization of a High-Pressure Compressor Casing with Global Constraints

The embedded global FE models have been shown to be a relatively accurate source of non-local structural behaviour. They did not completely eliminate violations of non-local constraints, because the parts of the global mesh used to evaluate these constraints are still mid-surfaces. However, they were still able to reduce what would be large non-local constraint violations in component design approaches that are not able to account for their presence in local design spaces. The case studies in Section 3.3 have demonstrated that a multi-fidelity approach with solid and mid-surface simulations can improve the prediction accuracy of these responses. These results will thus be extended to the component design-in-context approach in order to improve the reliability of the optimization results.

The second case study attempts to solve the problem as in the single-fidelity case in Section 3.4.1. Three strategies were devised to test the performance of the multi-fidelity Design-in-Context approach. Strategy I is a single-fidelity Kriging-based optimization that uses solid engine evaluations as data. The optimization is started with a 10-point

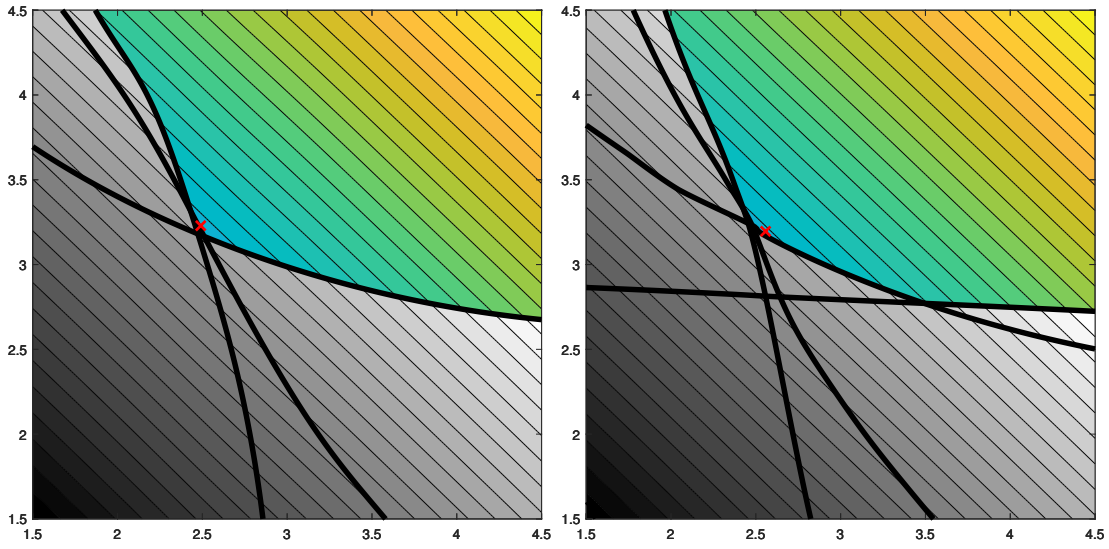
Table 3.12: Description of optimization strategies in the multi-fidelity CRESCENDO HPCC design-in-context case study.

Parameter	Strategy I	Strategy II	Strategy III
HF FE model	Solid engine	Embedded engine	Solid engine
LF FE model	N/A	Mid-surface engine	Embedded engine
Surrogate model	Kriging	Kriging	Co-Kriging
DOE method	Latin hypercube	Latin hypercube	Latin hypercube
# HF DOE points	10	10	3
# LF DOE points	N/A	N/A	10
Update method	Maximum EI	Maximum EI	Maximum EI
# HF updates	1	1	1
# LF updates	N/A	N/A	3

Latin Hypercube DOE, and in each update iteration a maximum EI point is evaluated. Strategy II is a multi-fidelity Co-Kriging-based optimization that uses embedded global model evaluations as high-fidelity data, and mid-surface engine evaluations as low-fidelity data. The optimization is started with a 10-point low-fidelity DOE equivalent to the DOEs in Strategy I, and a 3-point high-fidelity DOE that is generated by taking an optimum space-filling subset of the low-fidelity DOE. In each update iteration, 3 maximum EI points are evaluated used the low-fidelity model, as well as 1 imputed maximum EI point that is evaluated using the high-fidelity model. Strategy III is also a multi-fidelity Co-Kriging-based optimization but one that uses solid engine evaluations as high-fidelity data, and embedded global model evaluations as low-fidelity data. The update method in Strategy III is identical to Strategy II.

Table 3.12 summarizes these strategies. A maximum EI value of 0.01 kg was used as the convergence criteria for all optimizations. Each strategy was then repeated with ten differently seeded Latin Hypercube DOEs of the same sizes to obtain some statistics.

Table 3.13 summarizes the optimization results. The average mass reduction achieved by both multi-fidelity Design-in-Context approaches of Strategy II and Strategy III did not converge to the same level of mass reduction as in the expensive solid mesh only approach of Strategy I. The two strategies were still able to achieve a significant mass reduction at only a third of the computational cost. The multi-fidelity Design-in-Context approach though still not able to satisfy the most active constraint HPTC.CON2, the magnitude of the violation has remained low at 0.06%. This could be explained by how the benefits of the multi-fidelity approach is mostly concentrated within the HP compressor casing,



(a) Strategy II (b) Strategy III

Figure 3.10: Response surfaces of the multi-fidelity CRESCENDO HPCC design-in-context optimization problem. These response surfaces correspond to the first repeated run.

since the global constraints in both simulation fidelities are still evaluated using the mid-surface mesh. In contrast, the multi-fidelity Design-in-Context approach of Strategy III returned a feasible optimum design, albeit one that may be too conservative.

Plots of the response surfaces at the end of the optimizations in each strategy can be generated for a visual comparison. Figure 3.10 shows the response surfaces of the surrogate models that have been constructed from all evaluations in Strategy I and Strategy III. The response surface for Strategy II can be seen in Figure 3.4.

Figure 3.11 plots the search histories for all three strategies. The search histories are presented as a function of the equivalent number of low-fidelity simulations and include the cost of both the initial DOE and any subsequent updates. The cost conversions are performed using the cost ratios in Table 3.2 and Table 3.6, where the cost of simulating the solid HPCC in isolation is taken as the reference low-fidelity simulation cost. This conversion follows the method used in the HP compressor casing case study. The cost savings provided by both multi-fidelity Design-in-Context approaches can be clearly

Table 3.13: Optimization results for the multi-fidelity CRESCENDO HPCC design-in-context case study.

Result	Strategy I	Strategy II	Strategy III
Mass reduction (%)	5.34	4.80	4.22
Infeasibility (%)	0	0.06	0
Relative cost (%)	100	34.67	47.45

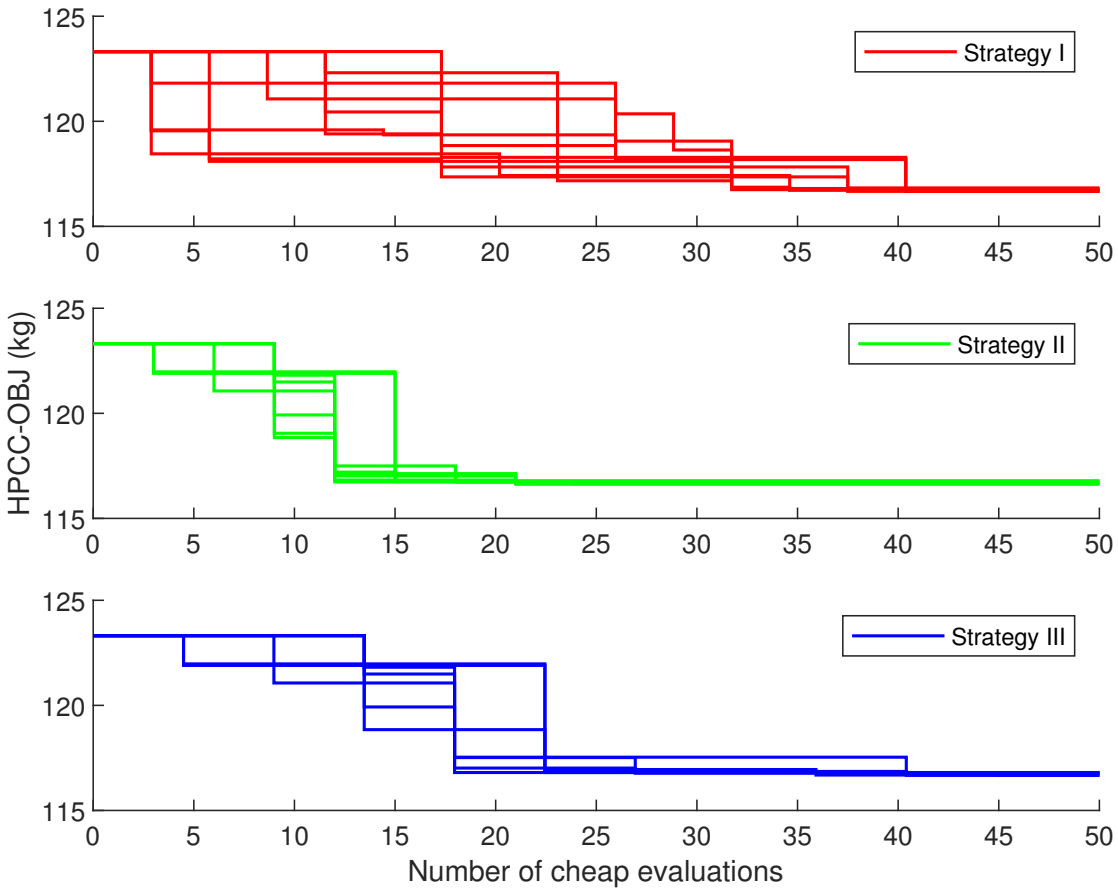


Figure 3.11: Optimization histories for the multi-fidelity CRESCENDO HP compressor casing with global constraints case study.

observed. Strategy III is naturally the more expensive Design-in-Context approach because it uses evaluations of the solid engine as its high-fidelity model. The question then needs to be asked if what seems like a small increase in reliability is worth the non-negligible increase in computational cost.

Chapter 4

Safe Integration of Concurrent Component Design Activities

4.1 Extension of the Component Design-in-Context Approach to a Concurrent Environment

We have shown in Chapter 3 how embedded global finite element models allow component designers to infer the impact of their design choices onto the requirements in their neighbouring components in a cost-effective manner. By scaling these non-local requirements to the range of the mid-surface portions of the global FE model used to predict them, or by using a multi-fidelity approach where these non-local requirements are occasionally evaluated with solid FE models, component designers can even include them directly into their driving set of requirements. An important follow-up question is how should the results from these component design-in-context activities be integrated into a concurrent design environment. Satisfaction of a constraint by multiple parties independently does not guarantee that the satisfaction will hold for the assembled solution.

In this chapter, a safe integration approach is proposed to address this issue by providing a systems integrator with information about how the component optimum designs should be adjusted in order to get a feasible assembled solution. This approach is also extended across multiple design iterations to explore how the combination of the component design-in-context approach and the proposed integration approach affects the performance of the product development lifecycle.

4.2 Approaches for Managing Concurrent Component Design Activities

In this section, the field of multidisciplinary optimization (MDO) is introduced with a focus on the concurrent subspace optimisation architecture (CSSO). The motivation for this is that the sub-problem formulation in the architecture shares many similarities with the design-in-context method, in the sense that constraints from the residual system are also modelled and included in sub-problem optimisations. There is thus an emphasis on prioritising the generation of a sequence of system-feasible designs that progressively improve, rather than investing the full design effort from the start towards searching for the best design possible with no recourse for early termination or changes in design goals. This property is unfortunately often overlooked when comparing CSSO against other MDO architectures in the literature which prioritize low numbers of function evaluations.

4.2.1 Multidisciplinary Design Optimization

Multidisciplinary optimisation (MDO) is concerned with the optimisation of complex, coupled systems (Martins and Lambe (2013)). Such systems are widely found in the aerospace industry where the fields of aerodynamics, structures, controls, and many others have to work in cohesion to produce an optimal system design. In its earliest days, much of MDO research was focused on developing *monolithic architectures*, where a single decision-maker is tasked with controlling all design variables at once. Due to the coupled nature of the engineering disciplines, there was a need to integrate the various computer codes into a single entity that automatically cycles through each discipline. Inputs and outputs are passed from one code to the next, until an equilibrium is reached, as signalled by a convergence of the residuals to a sufficiently small value. This analysis loop is called multidisciplinary design analysis (MDA), and the monolithic MDO architecture that uses it is called the multidisciplinary design feasible (MDF) architecture.

Here $\mathbf{f} = [f_1, f_2, \ldots, f_n]$ are the objective functions, where the subscript denotes the discipline number. Similarly, $\mathbf{g} = [g_1, g_2, \ldots, g_n]$ and $\mathbf{h} = [h_1, h_2, \ldots, h_n]$ represent the inequality and equality constraints respectively. The vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ contains all the design variables in the system.

The MDF approach can be formally expressed as,

$$\min_{\mathbf{x}} f(\mathbf{y}(\mathbf{x},\mathbf{y}),\mathbf{x}) \quad \text{while satisfying} \quad \begin{aligned} &\mathbf{g}(\mathbf{y}(\mathbf{x},\mathbf{y}),\mathbf{x}) \leq \mathbf{0} \\ &\mathbf{h}(\mathbf{y}(\mathbf{x},\mathbf{y}),\mathbf{x}) = \mathbf{0}, \end{aligned} \tag{4.1}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ are the state variables in the system, and they normally represent the outputs from an analysis. For example, changing the wing planform (a design variable, $\mathbf{x}$) will affect the lift distribution on the wing (a state variable, $y_1(\mathbf{x})$), and this will further affect the wing deflection ($y_2(y_1, \mathbf{x})$). Here the assumption is that there is only one system objective, for example the aircraft weight ($f(y_2(y_1, \mathbf{x}), y_1(\mathbf{x}), \mathbf{x})$).

The multidisciplinary feasible architecture has several disadvantages that prevent it from being applied in design studies of realistic engineering components:

1. The multidisciplinary design analysis process involves many iterations for convergence and its stability can be poor for tightly-coupled codes. It can be very expensive to solve the MDA for every design that is considered, especially when the individual discipline/component codes are also expensive to solve in isolation.
2. The complete set of design variables in the system is handled by a single optimizer, leading to high-dimensional problems which can be expensive to solve in the absence of explicitly defined derivatives or inaccurate estimations of them.
3. Significant time and effort is needed to integrate the different computer codes. Additional tools have to be written to convert the outputs from one code into a format that is compatible for another.
4. The role of lower-level design teams is effectively reduced to only conducting analyses and they have no say in the selection of new designs.

The multidisciplinary feasible architecture can be infeasibly expensive for late-stage design maturation, but it is important to note that its solution would theoretically be the true system optimum, as the single optimiser has control over every variable, while inter-component interactions are fully accounted for in the MDA. The question then becomes: how should this large problem be decomposed into smaller-sized problems that can be solved independently by multiple parties, while maintaining the coupling between component behaviour, such that the optimum found is equivalent to the MDF solution?

4.2.2 Concurrent Subspace Optimization (CSSO)

As research into MDO architectures matured, the need to have multiple decision-makers working in collaboration akin to realistic engineering design environments lead to the development of *distributed architectures*. One of the earliest distributed architectures is concurrent subspace optimization (CSSO). CSSO is a non-hierarchic MDO architecture which was designed to take advantage of the global sensitivity equations that were developed in Sobieszczanski-Sobieski (1990). In this architecture, the problem decomposition creates independent sub-problems, each with full control over a unique portion of

the overall design vector. For example, in a three-component system the decomposition would be,

$$\mathbf{x} \rightarrow \left\{ \begin{array}{c} x_1 \\ \vdots \\ x_m \\ \hline x_{m+1} \\ \vdots \\ x_n \\ \hline x_{n+1} \\ \vdots \\ x_p \end{array} \right\} \rightarrow \left\{ \begin{array}{c} \mathbf{x}_1 \\ \hline \mathbf{x}_2 \\ \hline \mathbf{x}_3 \end{array} \right\}. \tag{4.2}$$

4.2.2.1 Global Sensitivity Equations

The global sensitivity equations (GSE) are first-order gradients which describe the sensitivity of every state variable with respect to every design variable in a complex, tightly-coupled system. The GSEs are typically calculated at the system level and, when sent to the component level, allows the individual design subspaces to approximate any non-local functional relationships, including objective and constraint functions. Interestingly, it can be argued that the difficulty of computing the GSEs, along with the limited amount of information that is encapsulated in the GSEs, has lead to the stagnation of research into CSSO in recent years.

Two methods for generating the GSEs have been proposed: (1) GSE1 which is based on residual information, and (2) GSE2 which is based on sub-problem sensitivities. As the calculation of GSE2 is generally more convenient given the nature of existing engineering analysis software, the scope of the following discussion will be limited to GSE2.

Consider a system of three components A, B and C that are fully coupled to each other, meaning the evaluation of each component is dependent upon the outputs from every other component,

$$\begin{aligned} A((\mathbf{x}, \mathbf{y}_B, \mathbf{y}_C), \mathbf{y}_A) &= \mathbf{0}, \\ B((\mathbf{x}, \mathbf{y}_A, \mathbf{y}_C), \mathbf{y}_B) &= \mathbf{0}, \\ C((\mathbf{x}, \mathbf{y}_A, \mathbf{y}_B), \mathbf{y}_C) &= \mathbf{0}. \end{aligned} \tag{4.3}$$

The multidisciplinary analysis solution to Equation 4.3 will be denoted as $\mathbf{y}^* = [\mathbf{y}_A^*, \mathbf{y}_B^*, \mathbf{y}_C^*]$. By expressing each component's state variables as an explicit function of the design variables and the non-local state variables, first-order approximations of the neighbourhood of the MDA solution can be constructed using a Taylor series.

As an example, the first-order approximation for component A is,

$$\mathbf{y}_A = \mathbf{y}_A^* + \frac{\partial f_A}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial f_A}{\partial \mathbf{y}_B}\Delta\mathbf{y}_B + \frac{\partial f_A}{\partial \mathbf{y}_C}\Delta\mathbf{y}_C, \tag{4.4}$$

where f represents a generic component response.

Deriving the same approximation for the remaining components, we arrive at a linearised version of Equation 4.3,

$$\begin{aligned}
\mathbf{y}_A - \mathbf{y}_A^* - \frac{\partial f_A}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \frac{\partial f_A}{\partial \mathbf{y}_B}(\mathbf{y}_B - \mathbf{y}_B^*) + \frac{\partial f_A}{\partial \mathbf{y}_C}(\mathbf{y}_C - \mathbf{y}_C^*) &= \mathbf{0}, \\
\mathbf{y}_B - \mathbf{y}_B^* - \frac{\partial f_B}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \frac{\partial f_B}{\partial \mathbf{y}_A}(\mathbf{y}_A - \mathbf{y}_A^*) + \frac{\partial f_B}{\partial \mathbf{y}_C}(\mathbf{y}_C - \mathbf{y}_C^*) &= \mathbf{0}, \\
\mathbf{y}_C - \mathbf{y}_C^* - \frac{\partial f_C}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \frac{\partial f_C}{\partial \mathbf{y}_A}(\mathbf{y}_A - \mathbf{y}_A^*) + \frac{\partial f_C}{\partial \mathbf{y}_B}(\mathbf{y}_B - \mathbf{y}_B^*) &= \mathbf{0},
\end{aligned} \tag{4.5}$$

Making use of the implicit function theorem, the sensitivities of each function $f(\mathbf{x}, \mathbf{y}) = 0; \mathbf{y} = f(\mathbf{x})$ can be expressed as,

$$\left[\frac{\partial f}{\partial \mathbf{y}}\right]\left\{\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k}\right\} = -\left\{\frac{\partial f}{\partial \mathbf{x}_k}\right\}. \tag{4.6}$$

Equation 4.6 is always simultaneous, linear and algebraic regardless of the nature of the original governing equations. Its solution vector describes the first-order influence that each design variable has on every state variable in the system.

The global sensitivity equations for this 3-component system is thus,

$$\begin{bmatrix} I & -\frac{\partial f_A}{\partial \mathbf{y}_B} & -\frac{\partial f_A}{\partial \mathbf{y}_C} \\ -\frac{\partial f_B}{\partial \mathbf{y}_A} & I & -\frac{\partial f_B}{\partial \mathbf{y}_C} \\ -\frac{\partial f_C}{\partial \mathbf{y}_A} & -\frac{\partial f_C}{\partial \mathbf{y}_B} & I \end{bmatrix} \begin{Bmatrix} -\frac{\partial \mathbf{y}_A}{\partial \mathbf{x}_k} \\ -\frac{\partial \mathbf{y}_B}{\partial \mathbf{x}_k} \\ -\frac{\partial \mathbf{y}_C}{\partial \mathbf{x}_k} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\frac{\partial f}{\partial \mathbf{x}_k} \\ 0 \end{Bmatrix}. \tag{4.7}$$

When generalised for systems that have multiple couplings between components, the relevant element in the matrix in Equation 4.7 would be replaced with sub-matrices. For example, if component A has three outputs and it receives two state variables from component C, element (1,3) in the matrix would be replaced with the sub-matrix,

$$\begin{bmatrix} -\frac{\partial f_{A_1}}{\partial \mathbf{y}_{C_1}} & -\frac{\partial f_{A_1}}{\partial \mathbf{y}_{C_2}} \\ -\frac{\partial f_{A_2}}{\partial \mathbf{y}_{C_1}} & -\frac{\partial f_{A_2}}{\partial \mathbf{y}_{C_2}} \\ -\frac{\partial f_{A_3}}{\partial \mathbf{y}_{C_1}} & -\frac{\partial f_{A_3}}{\partial \mathbf{y}_{C_2}} \end{bmatrix}. \tag{4.8}$$

4.2.2.2 Component-Level Optimization

With the sensitivity equations derived, each component designer can now quantitatively measure the impact of changing a local design variable onto a non-local response. The foundations for an architecture that enables empathetic decisions to be made within decomposed sub-problems have thus been set.

Continuing from the derivations in Section 4.2.2.1, before component design activities can begin, an evaluation of the sensitivity equations at an initial system design $\mathbf{x}^* = [\mathbf{x}_A^*, \mathbf{x}_B^*, \mathbf{x}_C^*]$ is required. An optimisation problem for component A can then be stated as follows,

$$\begin{aligned} & \min_{\mathbf{x}_A} f(\mathbf{x}_A, \mathbf{x}_B^*, \mathbf{x}_C^*, \mathbf{y}_A, \hat{\mathbf{y}}_B, \hat{\mathbf{y}}_C, \mathbf{z} = \{\mathbf{r}, \mathbf{t}, \mathbf{s}\}), \\ \text{subject to} \quad & \hat{g}_A(\mathbf{x}_A, \mathbf{y}_A) \leq \max\{\hat{g}_A(\mathbf{x}_A^*, \mathbf{y}_A^*)\} s_{A_A}(1 - r_{A_A}) + (1 - s_{A_A}) t_{A_A}, \\ & \hat{h}_A(\mathbf{x}_A, \mathbf{y}_A) = 0, \\ & \hat{g}_{B_A}(\mathbf{x}_A, \mathbf{y}_A) \leq K_B(\mathbf{x}_A, \mathbf{y}_A) s_{B_A}(1 - r_{B_A}) + (1 - s_{B_A}) t_{A_B}, \\ & \hat{g}_{C_A}(\mathbf{x}_A, \mathbf{y}_A) \leq K_C(\mathbf{x}_A, \mathbf{y}_A) s_{C_A}(1 - r_{C_A}) + (1 - s_{C_A}) t_{A_C}. \end{aligned} \tag{4.9}$$

The system objective function f is minimized with respect to the vector of local variables $\mathbf{x}_A$. $\mathbf{y}_A$ is the vector of outputs from the local computer code, while $\hat{\mathbf{y}}_B$ and $\hat{\mathbf{y}}_C$ are approximations of the outputs from components B and C calculated by using the global sensitivity equations in a Taylor series,

$$\begin{aligned} \hat{\mathbf{y}}_B &= \mathbf{y}_B^* + (\frac{\partial \mathbf{y}_B}{d\mathbf{x}_A})(\Delta \mathbf{x}_A), \\ \hat{\mathbf{y}}_C &= \mathbf{y}_C^* + (\frac{\partial \mathbf{y}_C}{d\mathbf{x}_A})(\Delta \mathbf{x}_A). \end{aligned} \tag{4.10}$$

z is the vector of coordination parameters that is set by the system-level optimizer. A discussion of the method for setting the values of these parameters is presented in Section 4.2.2.3.

In the original CSSO architecture, the vector of equality and inequality constraint functions in each component are aggregated into scalar-valued functions $\hat{h}$ and $\hat{g}$, respectively,

in order to minimize the number of constraints in both the component optimizations and the system-level coordination. The overall problem size would thus only increase when the number of components increases, instead of when the number of constraint functions in each component increases. A Kreisselmeier-Steinhauser aggregation of the inequality constraint functions local to component A is,

$$K_A(\mathbf{x}_A, \mathbf{y}_A) = \frac{1}{\rho} \sum_{j=1}^{m_A} e^{\rho g_{A_j}}, \tag{4.11}$$

where g_{A_j} is the jth inequality constraint function in component A and m_A is the constraint count. This leads to $\hat{g}_A = K_A$ and the same method can be applied to obtain the aggregated equality constraint function, $\hat{h}_A$ as well. It is important to recall here that these local constraint functions take approximations of the outputs from the non-local computer codes as input.

On the other hand, the non-local constraint functions are aggregated using a linear approximation,

$$\hat{g}_{B_A}(\mathbf{x}_A, \mathbf{y}_A) = K_B(\mathbf{x}_A^*, \mathbf{y}_A^*) + \sum_{j=1}^{n_A} \frac{\partial K_A}{\partial x_j}(\mathbf{x}_A^*, \mathbf{y}_A^*)\Big(x_{A_j} - x_{A_j}^*\Big), \tag{4.12}$$

where n_A is the number of design variables in component A. Due to its linearity, this approximation is only valid in the vicinity of the initial system design $\mathbf{x}_A^*$. Move limits are thus often enforced for $\mathbf{x}_A$.

The partial derivatives in Equation 4.12 can be calculated as,

$$\frac{\partial K_B}{\partial x_j} = \Big(\sum_{j=1}^{m_B} e^{\rho g_{B_j}}\Big)^{-1} \Big(\sum_{j=1}^{m_B} \Big(e^{\rho g_{B_j}} \frac{\partial g_{B_j}}{\partial x_j}\Big)\Big), \tag{4.13}$$

where m_B is the number of inequality constraint functions in component B. The value for $\frac{\partial g_{B_j}}{\partial x_j}$ can be obtained from the global sensitivity equations.

4.2.2.3 System-Level Coordination

The prescription of the constraint limit values for each component optimization problem is done by solving a system-level coordination problem. The main ideas behind the coordination method is as follows:

1. If an aggregated constraint is violated, component designs designs that worsen the system objective value are allowed if they reduce this violation.

2. Once an aggregated constraint is satisfied, component designs that improve the system objective value are allowed if they do not cause a violation of this constraint again.

3. The responsibility to reduce the violation of an aggregated constraint is shared equally among all components, thus a component designer need only reduce a portion of the violation of its own aggregated constraint.

The responsibility as mentioned in the third point above is represented quantitatively as a r coefficient. r_{B_A} is thus the responsibility of reducing the violation of the aggregated constraint in subspace B by component A. Similarly, r_{A_A} is the responsibility given to component A for reducing the violation in its own aggregated constraint. r is normalised such that its sum over all components equals one. This introduces p^2 responsibility coefficients into the overall problem, where p is the number of components.

To increase the flexibility of the coordination method, trade-off or t coefficients are included to allow components to reduce its burden of satisfying an aggregated constraint by making sure that another component is able to compensate. A condition is placed such that only trade-offs that improves the system objective value are allowed. Normalization of t is done such that its sum for each aggregated constraint across all components equals zero. This once again introduces p^2 coefficients into the overall problem.

Every component optimization problem is started in a 'violation-reduction' mode until its aggregated constraints are satisfied, after which an 'objective-improvement' mode is initialised. A switching coefficient, s, is used to control this transition by moving from $s = 1$ to $s = 0$. For the local aggregated constraints, a factor of $\max\{\hat{g}_A(\mathbf{x}_A^*, \mathbf{y}_A^*)\}$ is used so that constraints which have already been satisfied are no longer taken into consideration.

A system-level coordination problem in the form of a linear program is then solved to yield new values for the set of these parameters, $\mathbf{z}$,

$$\min_{\mathbf{r},\mathbf{t}} \left(f(\mathbf{x},\mathbf{y}) + \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{\partial f}{\partial r_{j_i}} \Delta r_{j_i} + \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{\partial f}{\partial t_{j_i}} \Delta t_{j_i} \right), \tag{4.14}$$

where the $\mathbf{r}$ and $\mathbf{t}$ vectors contain all the r and t coefficients in the system. Note that the s coefficients are controlled within the components which is where the constraint violations are assessed, thus Equation 4.14 has no control over them. The dimension of the coordination problem is never larger than p^2 as the mode-switching element makes

the responsibility and trade-off behaviours independent from each other. New values for the r and t coefficients are then passed to each component and the process is repeated. Applications of the original CSSO formulation can be found in Bloebaum et al. (1992), where Equation 4.14 was solved using a generalised reduced-gradient algorithm.

4.2.2.4 Limitations of the CSSO Architecture

There has been numerous research efforts to improve the competitiveness of the concurrent subspace optimization architecture relative to the other MDO architectures. Modifications to the formulations of the coordination parameters and the cumulative form of the responses were proposed to improve its convergence performance (Bloebaum et al. (1992), Shankar et al. (1993), Renaud and Gabriele (1993), Wujek et al. (1996)). Some research has also gone into extending CSSO to solve multi-objective problems (Parashar and Bloebaum (2005), Huang et al. (2007)). However, the significant cost of calculating the sensitivity equations remained as the primary factor preventing CSSO from gaining widespread popularity, even with its emphasis on minimizing rework which is highly-valued in industrial design processes.

New system approximations are required at the start of every iteration, requiring expensive recalculation of the GSEs involving multidisciplinary analysis cycles. Coupled with the need to use move limits in the coordination problem due to the linearity of the approximations, a large number of iterations were required, making the overall cost infeasibly high. Renaud and Gabriele (1993) addressed the move limit issue by storing objective and constraint function evaluations after every design iteration in a database and subsequently using this information to construct second-order approximations for the coordination problem. However, even with this modification, the GSEs still had to be recalculated frequently.

4.3 A Safe Approach for Integrating Concurrent Component Design Activities

Research into multidisciplinary design optimization architectures like concurrent subspace optimization provide insights as to how different problem formulations and treatment of inter-component couplings can affect the progress of a system design process. However, most of the existing examples on MDO applications are on the use of monolithic architectures early on in the system design lifecycle to establish configurations. There is a lack of research in trying to tailor the distributed architectures to the detail design stage where the coordination of independent teams is still a necessity. The difficulty lies in the fact that engineering design in practice is a complex, human-driven, and time-sensitive endeavour. There is a lot of value in preserving knowledge bases and

systems engineering best practices that have been built up over time. Restructuring organization-wide design goals and methods to fit an MDO architecture can be very risky.

It is thus the goal of the proposed approach to be effective at guiding the overall structural design process without demanding significant changes to existing workflows. The *coordination-based* approaches that are developed in MDO, where component objectives are modified akin to mechanism design, are thus foregone for a less-intrusive *integration-based* approach in which the responsibility for assembling the component designs into a feasible whole rests squarely on a systems integrator.

This is achieved by constructing a system-level Kriging surrogate using FE model evaluations from the component design process. This surrogate can then be used at end-of-iteration design reviews to search for a load path configuration that has improved load-sharing capabilities and can be used as the kick-off point for the component design processes in the next iteration. Using this approach, component designers can then direct their full effort and attention towards making value-added decisions based on actual performance requirements, instead of 'interface requirements' that are purely by-products of the decomposition process. There is evidence suggesting that the removal of these interface requirements can accelerate the overall design process (Bell et al. (2008)). These interface requirements are also present in most distributed MDO architectures in the form of consistency constraints.

Load path configurations represent stable states of an engine's structural behaviour that are shared with teams of other disciplines, with external suppliers, and with the airframe manufacturer throughout the design process. There is thus a requirement that the configuration that is arrived at at the end of the integration process needs to be feasible. The cost of validating the configuration is also expensive as it involves the solution of a high-fidelity global FE model. The search for a new configuration thus needs to be 'safe', that is, we seek an immediate improvement in the feasible performance using a minimal number of evaluations of the global FE model.

The system-level surrogate will be inaccurate as there are only a limited number of data points for what is a high-dimensional problem. Using a Kriging surrogate, the value of the constraints at unsampled locations are predicted based on the observations at sampled points. The Kriging predictions have uncertainty, which is represented using Gaussian distributions. The proposed integration method classifies unsampled points into two categories, 'safe' and 'unsafe'. A point is labelled as safe if it satisfies every constraint almost certainly, that is, the probability of feasibility is greater than a pre-defined confidence level. Otherwise, it is considered unsafe and should not be sampled.

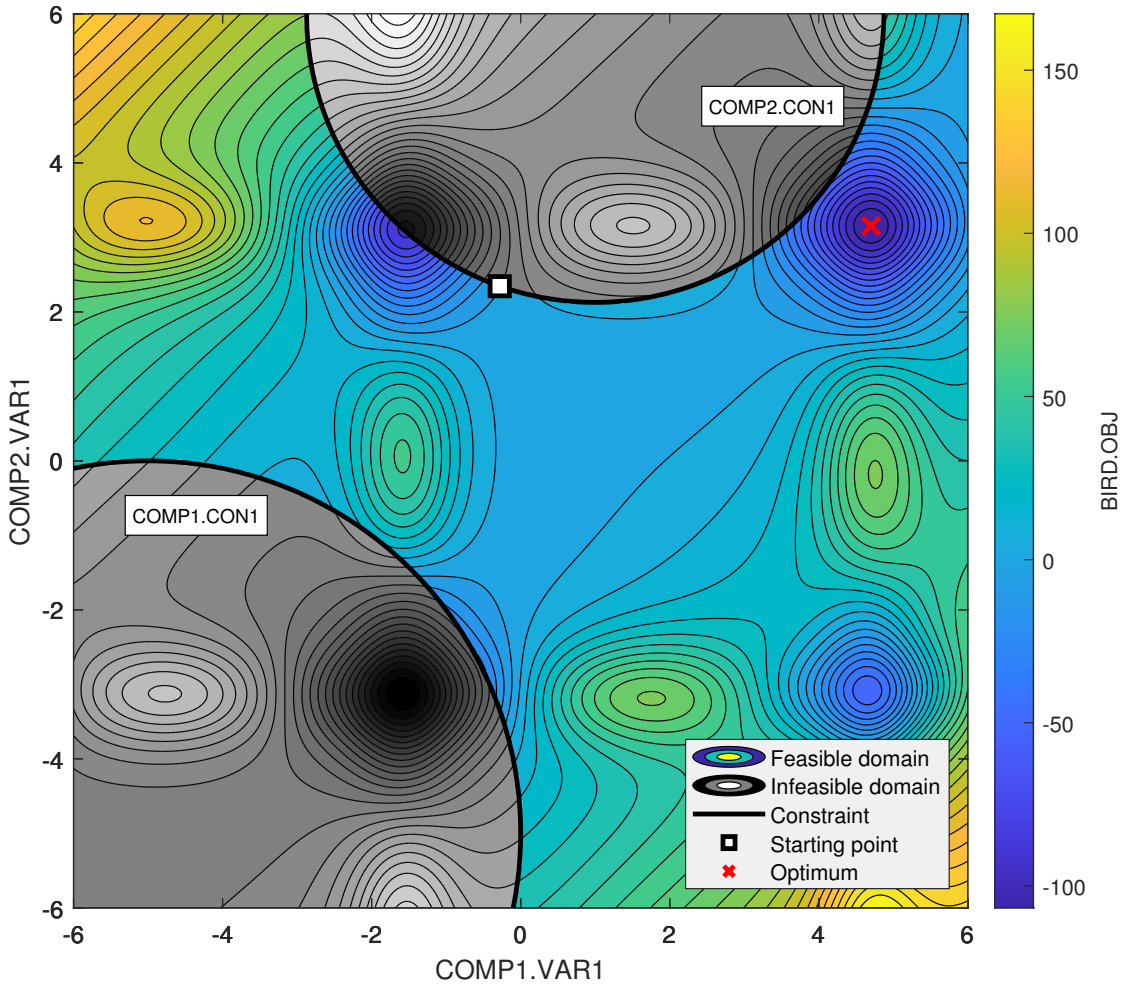


Figure 4.1: Response surface of the distributed Bird function.

4.3.1 An Illustrated Example

The proposed integration approach is illustrated here using a simple 2D optimization of the constrained Bird function. This problem has been reformulated into a distributed optimization problem with two components (Section A.1). The first component, COMP1, controls the design variable COMP1.VAR1 and evaluates its objective COMP1.OBJ and a constraint COMP1.CON1. The second component, COMP2, controls the design variable COMP2.VAR1 and evaluates its objective COMP2.OBJ and a constraint COMP2.CON1. Figure 4.1 shows the true response surface of this problem.

Figure 4.2 illustrates what would happen if the component optimum designs were simply combined through a 'direct' integration approach (as opposed to a safe integration approach) to form the system solution. Each component designer searches within their slice of the design space using a design-in-context approach. For example, designs for component COMP1 is searched for within the {COMP1.VAR1, COMP2.VAR2 = 2.3438} slice while satisfying constraints COMP1.CON1 and COMP2.CON1. The component

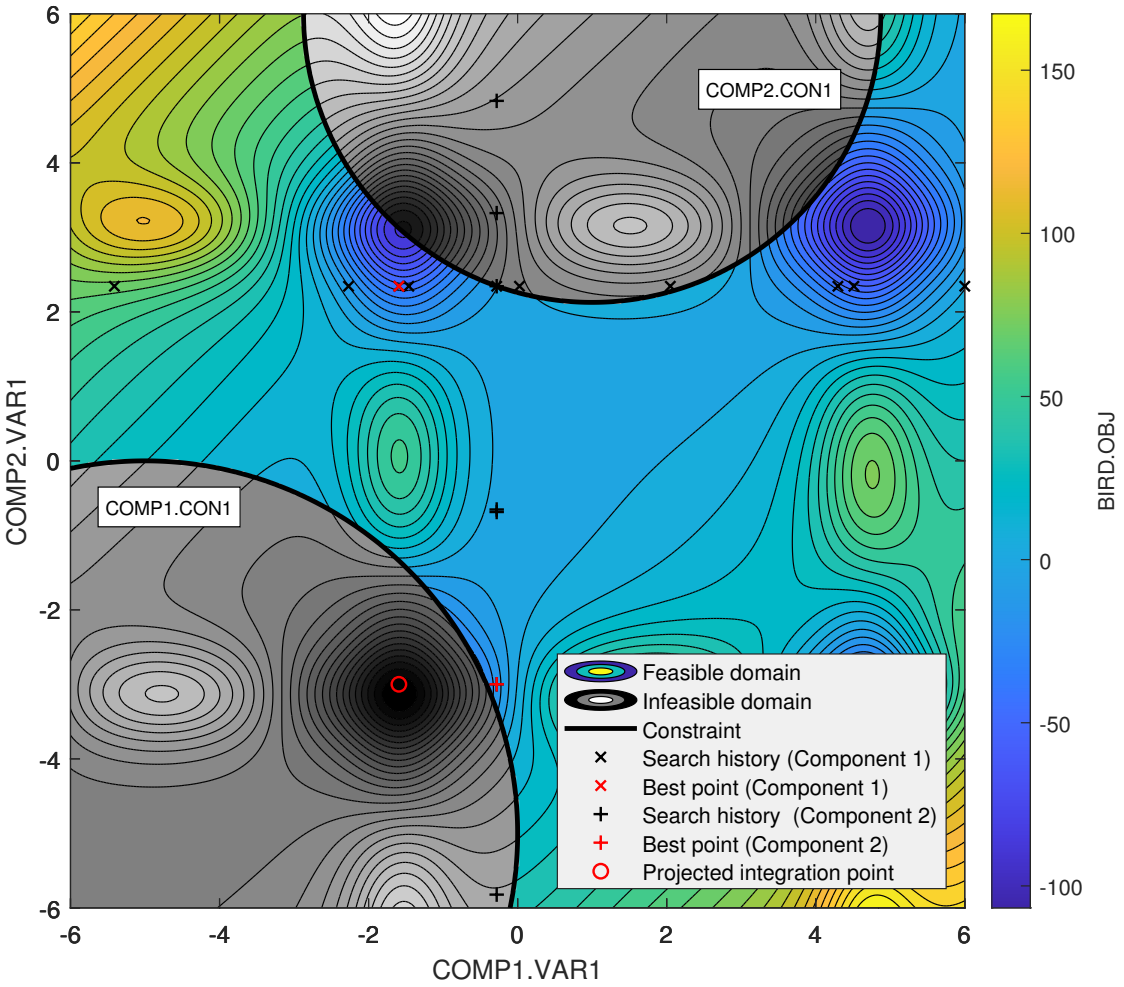
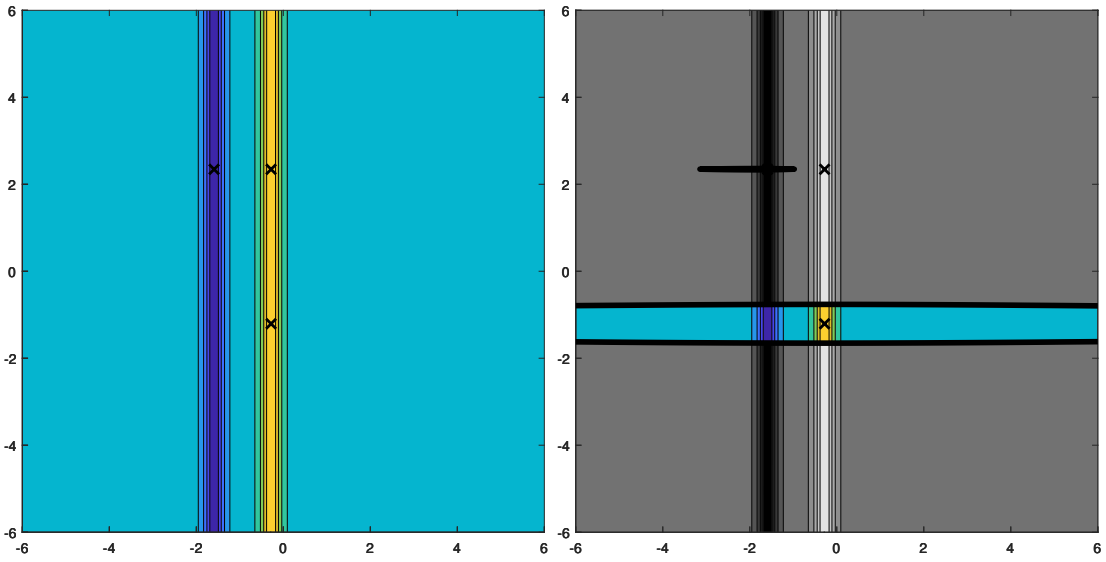


Figure 4.2: Example of an infeasible system design resulting from a direct integration of component optimum designs.

designers thus individually arrive at optimum designs that are feasible. However, because they have not accounted for the decisions made by the other party, the resulting combined solution was found to violate the constraint COMP1.CON1.

In the proposed integration approach, a system-level surrogate is constructed using the component optimum designs. Due to the low number of samples, the resulting surrogate is not expected to predict the constraints accurately. The constraint responses are thus replaced with their probability of feasibility responses, and their bounds are replaced with a single lower bound that corresponds to a confidence level, $\mathrm{PF}_{\mathrm{min}}$, that is prescribed by the systems integrator pertaining to the level of acceptable risk.

Figure 4.3 compares a feasible region computed by predictions of the constraints against another that is computed by the probabilities of their satisfaction. The confidence level for the latter approach is set to 0.90. There are only three samples in this 2D design space: the starting system design, the optimum COMP1 design, and the optimum COMP2 design. It can be observed that the prediction-based surrogate was unable to locate any constraints in the design space. On the other hand, the probability of



(a) Predicted constraints.

(b) $\mathrm{PF}_{\mathrm{min}} = 0.90$ constraints.

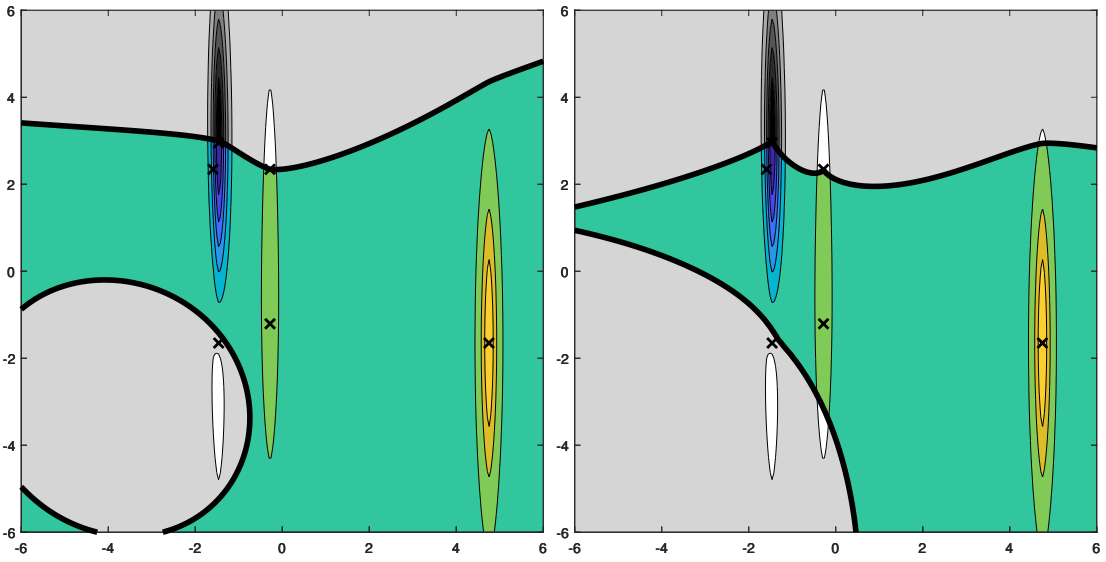
Figure 4.3: Comparison of the Bird problem's response surface between the prediction-based approach and a probability of feasibility-based approach for modelling the constraints after one system-level iteration.

feasibility approach produces a very conservative estimate of the feasible region that is concentrated around existing feasible samples. Regions that will violate the constraint COMP2.CON1 have been completely excluded.

For the constraint COMP1.CON1, a portion of the true infeasible region remains. This overconfidence can be attributed to its distance from the samples. One can simply increase the required confidence level to further shrink the feasible region, or leave it infeasible and proceed to the next system-level iteration if it is acceptable to do so, especially in the earlier iterations when the configuration has yet to be depended upon by external parties.

One can also budget more time for the integration process such that two or three update iterations are allowed as opposed to the default single-iteration implementation. A low required confidence level could be purposely set to encourage exploration of the design space, and any resulting samples that are infeasible are used to update the surrogate for a subsequent search until a feasible solution is found. The strength in the proposed approach is thus the flexibility that is provided to the systems integrator in performing the integration task subject to engineering judgement and to the requirements of specific stages in the overall design process.

Figure 4.4 shows the response surface after a further system-level iteration. In this case, the infeasible integrated solution has been left as is and is used to fix the locations of the slices for another cycle of component design searches. The required confidence level for the probability of feasibility approach has been maintained at a value of 0.90. The under-



(a) Predicted constraints. (b) $PF_{min} = 0.90$ constraints.

Figure 4.4: Comparison of the Bird problem's response surface between the prediction-based approach and a probability of feasibility-based approach for modelling the constraints after two system-level iterations.

and over-conservative predictions of the feasible region between the two constraint modelling approaches can again be observed. The feasible region concentrates appropriately around feasible samples, and the resulting predicted optimum at $[-1.4092, 2.8951]$ can be shown to be feasible.

4.3.2 Application over Multiple Iterations

The proposed safe integration approach is designed to allow a systems integrator to make decisions based on quantitative data for individual system-level iterations. The outcome of each of these integration activities is governed by several parameters at both the component and system level. This section is focused on investigating how variations in these parameters affect the performance of the proposed approach over multiple iterations. The goal of this study is to provide some guidance and experimental data to practitioners for developing their own best practices when using the safe integration approach.

Benchmark Problems

It will be extremely expensive to run the full set of experiments with actual FE models so a suite of analytical benchmark problems is used instead. A decision was made to decompose traditional benchmark problems into sub-problems and impose an artificial coupling error, instead of using problems from the MDO literature (Hulme and Bloebaum (2000); Allison et al. (2006); Yi et al. (2008)). The main reason for this is there has yet

Table 4.1: Description of the analytical benchmark problems used to test the integration approaches. D is the total number of design variables in each problem. g is the number of inequality constraints. h is the number of equality constraints. $f(\mathbf{x}*)$ is the best known feasible objective function value from the literature. c is the number of components in the distributed version of each problem.

ID	Name	D	g	h	$f(\mathbf{x}^*)$	c
BIRD	Bird function	2	2	0	-106.7645	2
ROSU	Rosen-Suzuki function	4	3	0	-44.0000	3
SPDRED	Weight minimisation of a speed reducer	7	11	0	2994.4245	5
INDFRIG	Optimal design of an industrial refrigeration system	14	15	0	-0.0322	6

to be a comprehensive set of distributed design problems. Most of the existing coupled problems have been designed for monolithic MDO architectures and are thus presented in terms of a single large system of coupled equations with little suggestion as to how they should be decomposed. Scalable formulations like those presented in Tedford and Martins (2010), Tosserams et al. (2010), and Chauhan et al. (2018) also suffer from a lack of solution results against which the proposed methods can be compared.

Table 4.1 summarizes the four benchmark problems that have been selected, with a mix of both classical and engineering optimization problems. The test suite consists of a good variety of problem complexity in terms of dimensionality and the number of responses that are included. A complete description of each problem and their decomposed forms can be found in Appendix A.

To better represent the distributed design environment, the responses in each problem are augmented with additional error terms. The first type of error is the consistency error, e_c. The consistency error simulates the error in the analysis of an isolated component as its design shifts away from the system equilibrium. This is similar to how the free-body boundary conditions given to a component FE model becomes unreliable as the component design is changed from the state at which those boundary conditions were evaluated. This error is modelled such that it increases as the magnitude of the design change increases, and is given as,

$$e_c = \alpha \cos(\mathbf{x}) \cdot \|\mathbf{x} - \mathbf{x}_0\|, \tag{4.15}$$

where α is the maximum error, $\mathbf{x}$ is the current point, and $\mathbf{x}_0$ is the equilibrium point. In the current study, e_c is only applied to constraint evaluations in isolated component design processes.

The second type of error is the resolution error, e_r. The resolution error simulates the analysis error when evaluating models of lower fidelities, such as a mid-surface mesh. This error is adapted from Wang et al. (2018) and is given as,

$$\begin{aligned} e_r &= \alpha\theta \sum_{i=1}^{d} \cos(10\pi\theta x_i + 0.5\pi\theta + \pi), \\ \theta &= 1 - 0.0001\phi, \end{aligned} \tag{4.16}$$

where α is the maximum error, $\mathbf{x}$ is the current point, and ϕ is a parameter that can be used to tune the fidelity, where a value of 0 corresponds to the lowest fidelity (not necessarily $r^2 = 0$) and a value of 10000 corresponds to the highest fidelity ($r^2 = 1$ and no errors). e_r is applied selectively to the set of constraint evaluations that are associated with a low-fidelity model, a physical example being those originating from the mid-surface portions of an embedded global FE model.

A set of 25 feasible points are generated for each benchmark problem to serve as starting system designs for repeating the optimizations.

System-Level Iteration

A system-level iteration is modelled in two parts: a component-level optimization, followed by a system-level design search using the surrogate constructed by an integration method.

An initial system design is determined for the first iteration. Each component is then optimized using the Kriging-based approach as described in Chapter 3. Each component optimization is initiated with a Latin hypercube DOE of size $5 \times d$ and is updated with a single maximum EI point per component-level iteration until a maximum EI value of $< 1\%$ the objective of the current best design is reached. Once all component-level optimizations have converged, their optimum designs are evaluated using a high-fidelity global FE model and added to the system-level training data set.

A system-level surrogate is then constructed according to the integration method of choice and subsequently searched for a maximum EI point. This optimization process is terminated when a feasible integrated solution is found. This solution then serves as the starting system design for the next system-level iteration. The system-level optimization consisting of multiple of such iterations is terminated when a maximum EI value of $<$ 1% the objective of the current best design is reached.

Measures of Performance

Three metrics are established to compare the performance of the integration approaches.

The first metric is the relative accuracy of the solutions, a, calculated with respect to the best known feasible objective function values in Table 4.1.

The second metric is the cost, C. A conversion between the costs of component-level and system-level evaluations is required. A theoretical set of cost ratios is selected for the present study. The cost for evaluating an isolated component design is taken to be 10% of the cost for evaluating a high-fidelity global FE model, while the cost for evaluating an embedded global FE model is taken to be 20% of the cost for evaluating a high-fidelity global FE model. These cost ratios are greater than those found for the CRESCENDO engine in Chapter 3 and represent a perhaps more realistic engine architecture consisting of a larger number of components.

The third metric is the percentage of infeasible system designs, v. This percentage is used as a proxy for the probability that a design rework procedure is triggered. In a practical design environment, the number of system-level iterations allowed is subject to the schedule of the design programme and may be terminated at any time. This percentage is thus an indication of how likely a feasible final design will be obtained upon the termination of the design process.

4.3.2.1 Impact of the System-Level Iteration Strategy

In this case study, system-level iteration strategies with varying component-level design and integration approaches are compared. Two options are available for the component level: a design-in-context approach with an embedded global FE model of fidelity $\phi_{\mathrm{G}} = 8000$, and a default isolated approach with a local component FE model.

Three strategies were tested. Strategy I represents a collaborative design environment with minimum interactions, where components are designed in isolation, and the optimum designs are directly assembled to give the integrated solution. This strategy can be taken as a close approximation of a realistic organization. Strategy II represents an updated version of the response surface-based implementation of the CSSO architecture as described in Sellar et al. (1996), where components are designed with respect to non-local requirements but are then also directly assembled. Some elements of the CSSO implementation have been updated to improve the parity of the current comparison. The quadratic surrogates that were built from MDA evaluations for the purpose of eliminating the need for the global sensitivity equations are replaced here with Kriging surrogates built from evaluations of the global model. The move limits which were enforced to ensure the quality of the quadratic surrogate predictions have thus also been removed. Strategy III represents the most conservative approach, with design-in-context

Table 4.2: Description of system-level iteration strategies.

Parameter	Strategy I	Strategy II	Strategy III
Component design method	Isolated	Design-in-context	Design-in-context
Global model fidelity, ϕ_G	N/A	8000	8000
Integration method	Direct	Direct	Safe
Confidence level, $\mathrm{PF}_{\mathrm{min}}$	N/A	N/A	0.90

at the component level followed by the safe integration method. In this implementation, the required confidence level over all system-level iterations is set to 0.90. Table 4.2 summarizes these strategies.

Figure 4.5 shows the results of the study. In terms of the solution accuracy a, all three strategies managed to arrive at optimum designs that are close to the best reported solution except for the industrial refrigeration system problem where a significant error is observed. As all three strategies performed similarly poorly for this problem, the difficulty in finding the solution could be attributed to either the high dimensionality of the problem, or a poor decomposition.

With regard to the cost C, a noticeable difference can be observed. The strategies that use design-in-context at the component level are, in general, cheaper than the strategy that is based on an isolated component design process. Such an observation is unexpected, as there is no discernable reason why having more infeasible points in the system-level surrogate's training data set will cause the constraint predictions to be more inaccurate, and leading to more update iterations required for finding a feasible system design. The safe integration method in Strategy III, however, does causes it to be cheaper than the direct integration method in Strategy II. This is also reflected in the differences in the percentage v between Strategy II and Strategy III.

4.3.2.2 Impact of global model fidelity

In this case study, design-in-context-based approaches are tested with varying levels of the global model fidelity, ϕ_{G}. The validation studies in Section 2.5 and the case studies in Section 3.3.3 have shown that the mid-surface mesh is accurate with respect to an equivalent solid mesh. This level of accuracy may not hold for a different set of geometries, especially geometries that have a smaller proportion of thin-walled regions. The global model fidelity is expected to influence the reliability of the component optima, and will thus affect the proportion of feasible points in the data set that is used to

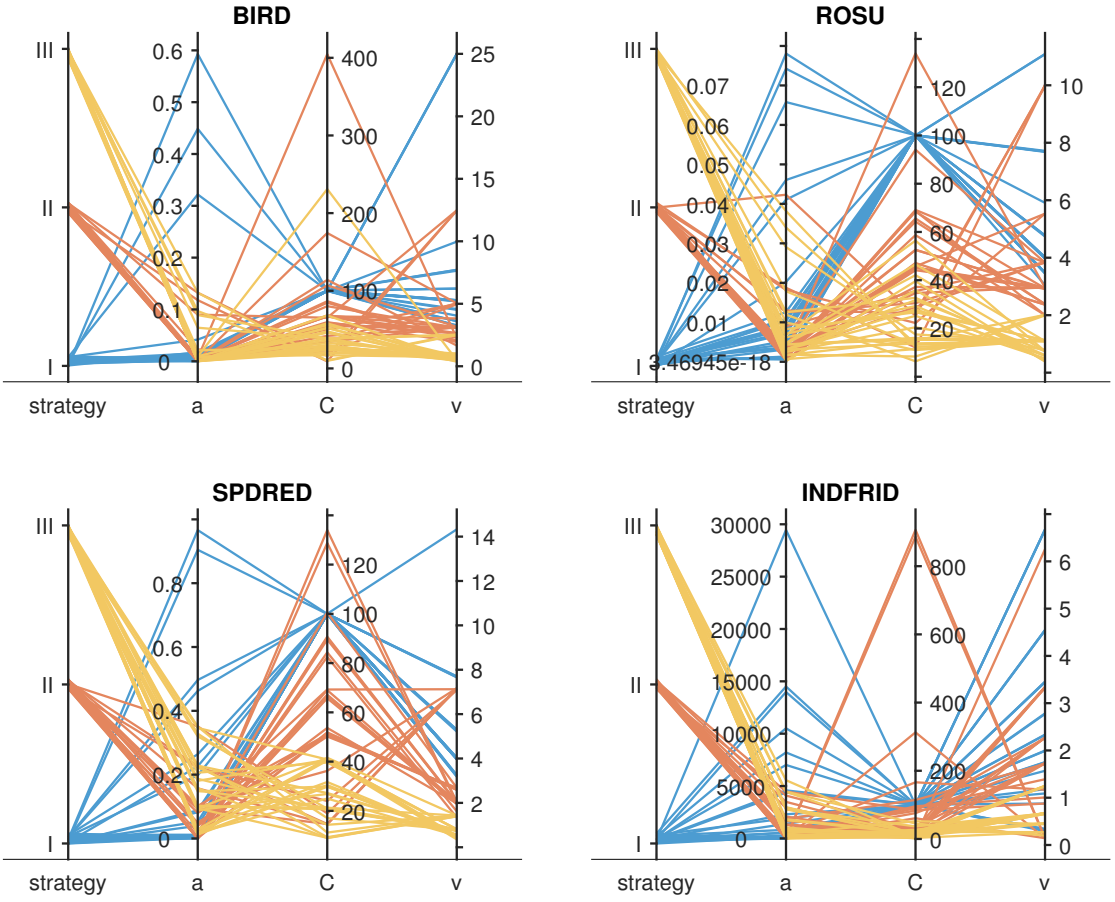


Figure 4.5: Results for the case study comparing the impact that the system-level iteration strategy has on the performance of the overall design process.

Table 4.3: Description of strategies with varying global model fidelities, ϕ_G.

Parameter	Strategy I	Strategy II	Strategy III
Component design method	Design-in-context	Design-in-context	Design-in-context
Global model fidelity, ϕ_G	0	4000	8000
Integration method	Safe	Safe	Safe
Confidence level, PF_{min}	0.90	0.90	0.90

construct the system-level surrogate. A lower-fidelity global model should cause this proportion to decrease.

The three fidelities tested correspond to different levels of correlation and errors. $\phi_G = 8000$ corresponds to a r^2 range of [0.78, 0.93] and a percentage RMSE range of = [7.31, 9.80]. $\phi_G = 4000$ corresponds to a r^2 range of [0.39, 0.68] and a percentage RMSE range of = [19.56, 27.23]. $\phi_G = 0$ corresponds to a r^2 range of [0.17, 0.27] and a percentage RMSE range of = [32.66, 48.04]. Table 4.3 summarizes these strategies.

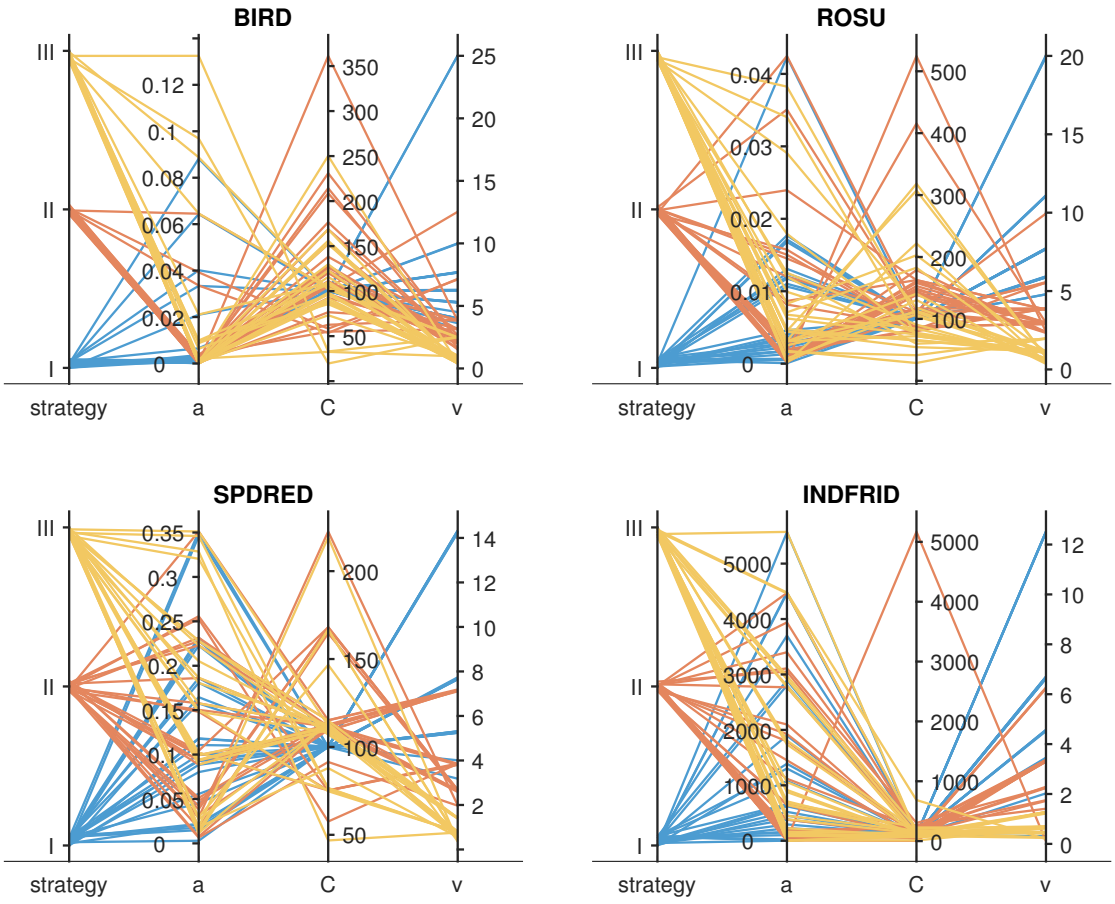


Figure 4.6: Results for the case study comparing the impact that the global model fidelity ϕ_{G} has on the performance of a component design-in-context-based safe integration process.

Figure 4.6 shows the results of the study. In terms of the solution accuracy a, all three strategies managed to arrive at optimum designs that are close to the best reported solution except for the industrial refrigeration system problem, due to reasons that have been discussed in the previous case study. With regard to the cost C, there is no noticeable difference between the three strategies. There is thus some evidence for the hypothesis that the proportion of infeasible points in the system-level surrogate's training data does not affect its ability to predict a feasible system design, in opposition to what was observed in the results of Section 4.3.2.1. The percentage v increases as ϕ_{G} increases, which is expected since component optimum designs are more likely to be feasible.

4.3.2.3 Impact of safe integration confidence level

In this case study, safe integration-based approaches are tested with varying levels of the required confidence level, $\mathrm{PF}_{\mathrm{min}}$. A higher confidence level causes the system-level surrogate to produce a more conservative estimate of the feasible region. This lowers

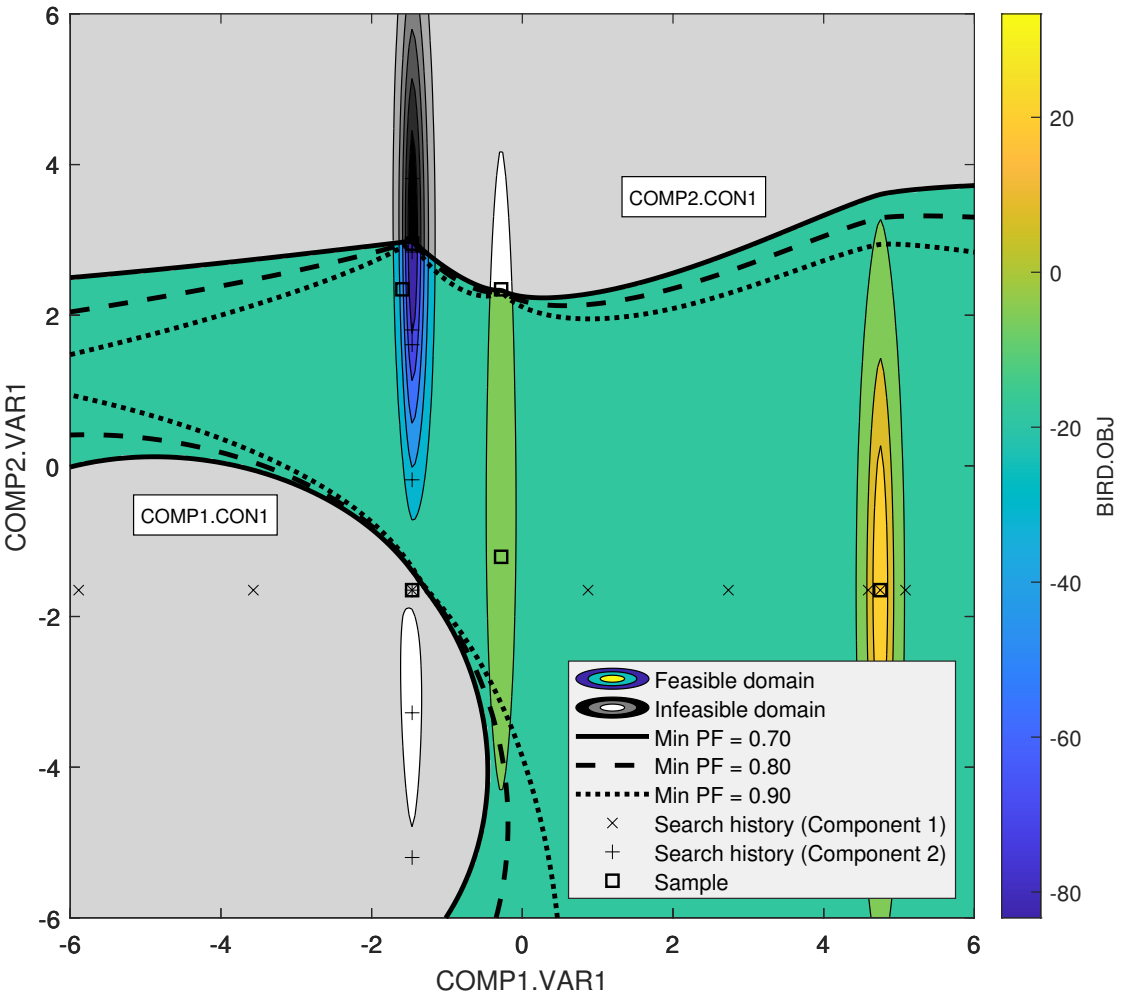


Figure 4.7: Comparison of the Bird problem's response surface with varying confidence levels. The component evaluations are also shown to indicate the component search paths, but they are not used in the construction of the system-level surrogate.

the magnitude of the design change allowed and causes the overall design process to progress more slowly through the design space. There is more assurance, however, that the process can be terminated at any one step and still return a feasible system design. Figure 4.7 illustrates the relationship between the confidence level and the size of the feasible region.

The three confidence levels tested are, from least to most conservative, 0.70, 0.80, and 0.90. Table 4.4 summarizes these strategies.

Figure 4.8 shows the results of the study. In terms of the solution accuracy a, all three strategies managed to arrive at optimum designs that are close to the best reported solution except for the industrial refrigeration system problem, due to reasons that have been discussed in the first case study. With regard to the cost C, there is a noticeable trend showing that the cost decreases as the conservativeness of the safe integration approach increases. A higher confidence level $\mathrm{PF}_{\mathrm{min}}$ causes the integration

Table 4.4: Description of strategies with varying confidence levels, PF_{min}.

Parameter	Strategy I	Strategy II	Strategy III
Component design method	Design-in-context	Design-in-context	Design-in-context
Global model fidelity, ϕ_G	8000	8000	8000
Integration method	Safe	Safe	Safe
Confidence level, PF_{min}	0.70	0.80	0.90

step to require less iterations to arrive at a feasible system design and hence decreases the overall cost. This is also reflected in how the percentage v, responds to changes in PF_{min}.

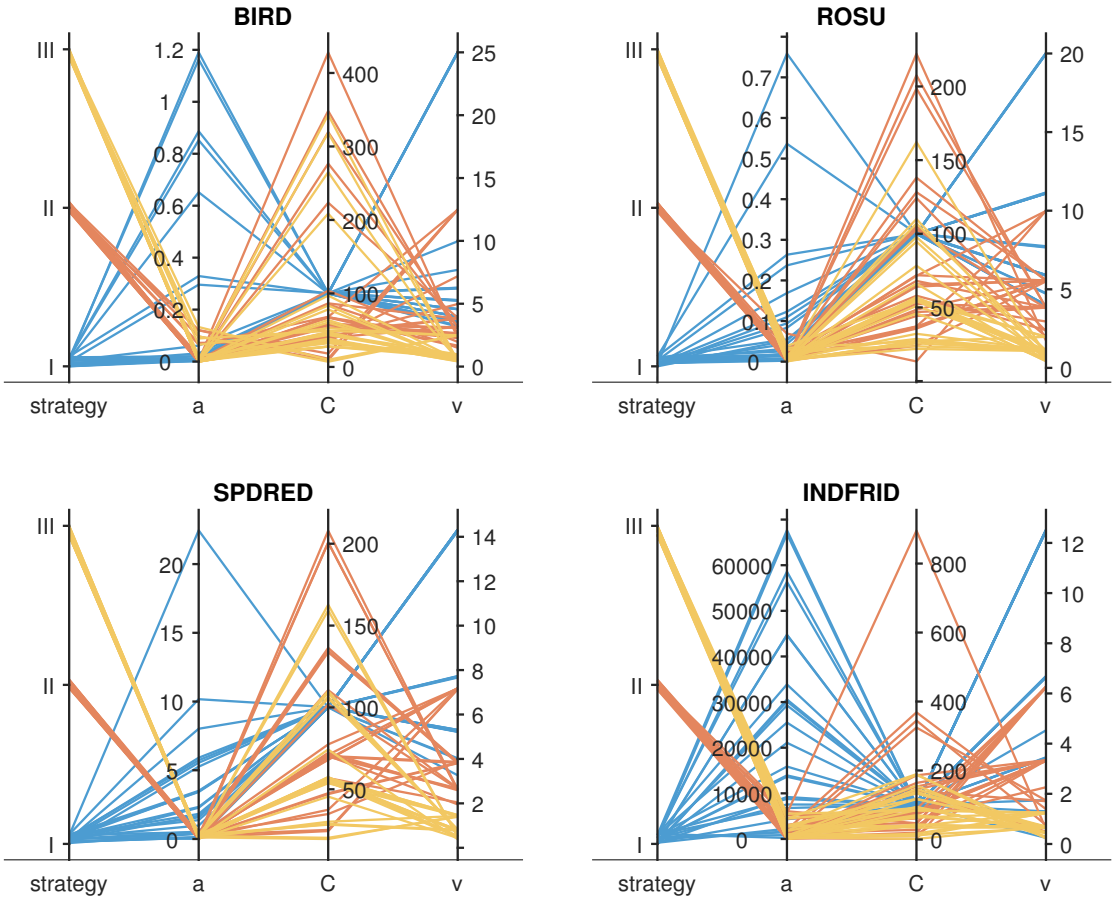


Figure 4.8: Results for the case study comparing the impact that the safe integration confidence level PF_{min} has on the performance of a component design-in-context-based safe integration process.

4.3.3 Distributed Design of a Full Engine

Having tested a variety of distributed optimization approaches on the analytical benchmark problems, a realistic design problem involving FE models is now presented. The subject of this case study is the CRESCENDO engine (Section 2.5).

In the decomposed formulation, the engine is split naturally along physical component boundaries into four sub-problems corresponding to the four components: the FC, the HPCC, the HPTC, and the LPTC. The parameters of each component sub-problem are identical to the previous case studies involving the CRESCENDO engine and can be seen in Figure 3.6 and Figure 2.13. The system-level iterations are carried out as described in Section 4.3.2. The design-in-context option for the component design method has been upgraded to its multi-fidelity version as described in Section 3.4.2. This multi-fidelity approach constructs Co-Kriging surrogates using embedded global FE model evaluations as high-fidelity data, and mid-surface engine evaluations as low-fidelity data.

A new multi-fidelity version for the safe integration method is also investigated. Currently, only the optimum design from each component design process is validated and added to the training data set for the system-level surrogate. This restriction exists because the validation process involves evaluations of the high-fidelity global model. However, if a multi-fidelity design-in-context approach is used at the component level, the low fidelity global model evaluations in each sub-problem are of the same fidelity and can be combined into a separate low fidelity data set for the system-level surrogate. The proposed multi-fidelity safe integration method thus takes advantage of this new data set to construct a Co-Kriging model.

The distributed optimization approaches are tested against an all-at-once approach, that is, the optimization proceeds with all eight variables at once and uses only high-fidelity global FE model evaluations. Table 4.3 summarizes these strategies.

Table 4.5: Description of system-level iteration strategies for the CRESCENDO engine distributed optimization problem.

Parameter	Strategy I	Strategy II	Strategy III	Strategy IV
Component design method	N/A (all-at-once)	Default isolated	Design-in-context (MF)	Design-in-context (MF)
Integration method	N/A	Safe	Safe	Safe (MF)
Confidence level, PF_{min}	N/A	N/A	0.90	0.90

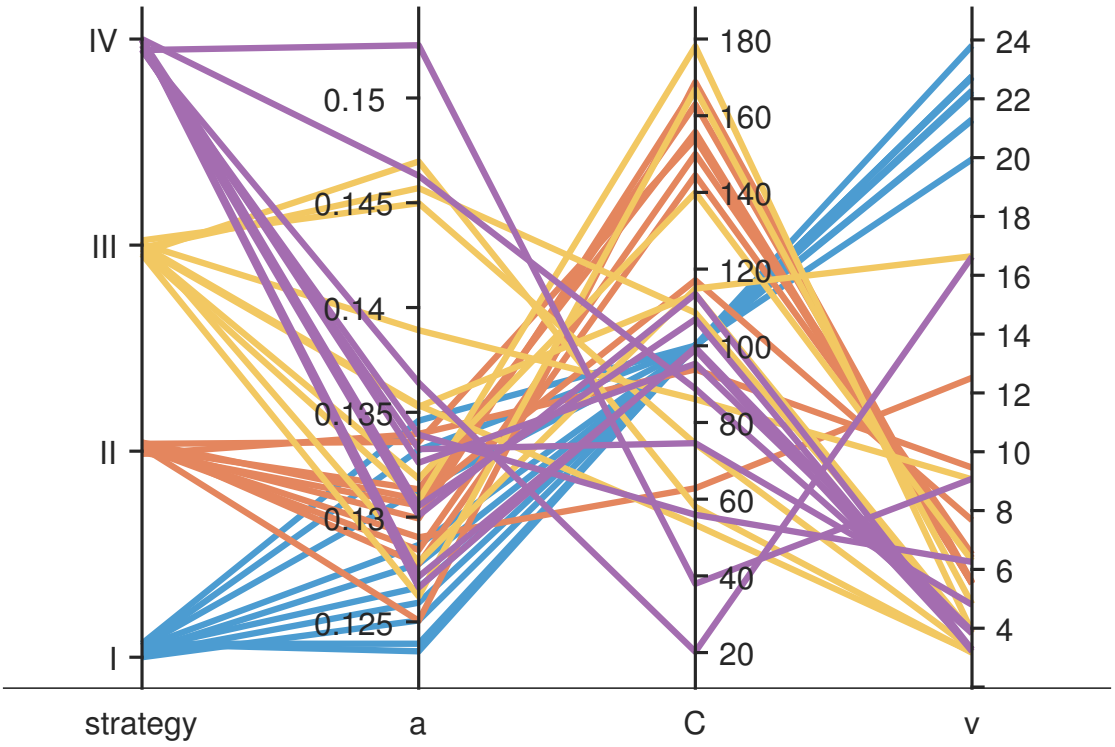


Figure 4.9: Results for the CRESCENDO engine distributed optimization problem.

Figure 4.9 shows the results of the study. In terms of the solution accuracy a, all three strategies managed to arrive at optimum designs that are close to the best reported solution in Section 3.3.3.

With regard to the cost C, Strategy II and Strategy III are noticeably more expensive than the all-at-once approach. This result is expected as the problem decomposition transforms a search through the original design space into a staggered series of searches along lower-dimensional slices. However, as the percentage v shows, it is much more risky to use the all-at-once approach if it does not converge before the optimization has to be terminated due to any time constraints.

The multi-fidelity safe integration approach in Strategy IV has managed to achieve costs that are on par with the all-at-once approach. This can be attributed to the larger steps that it can take in the integration phase without encountering infeasibilities due to the improved accuracy of the system-level surrogate. Its percentage v values also indicate that it can compete with the all-at-once approach in terms of cost while maintaining a low risk of rework.

Chapter 5

Conclusion

The major contributions of this thesis are:

- An extension of the medial object generation workflow originally established by the MANTLE library to accelerate the generation of Whole Engine Models using a geometry splitting, thickness remapping, and mesh coarsening approach.
- A comprehensive study of the properties of medial object-based mid-surface meshes and their accuracy relative to high fidelity tetrahedral meshes over a variety of design modifications and in novel configurations such as the embedded global FE model.
- A novel multi-fidelity surrogate modelling method that uses tetrahedral meshes as high fidelity data and mid-surface meshes as low fidelity data to achieve a high level of prediction accuracy at a lower cost than traditional single-fidelity methods.
- A novel design-in-context method that uses embedded global FE models to efficiently train surrogate models that can accurately predict the global landscape for component-level design processes.
- A novel integration method that constructs system-level surrogates from component-level optimum designs and conservatively estimates the feasible region such that a feasible system design can be found quickly and reliably.
- A comprehensive study of the effects that different combinations of component-level design strategies and system-level integration strategies have on the performance of the overall design process.

Overall, this thesis represents an effort to improve existing component-level processes in engineering design. By exploiting new finite element modelling technologies, resourceful use of multi-fidelity surrogate modelling methods, and taking concepts from concurrent

design methods, a set of reliable and efficient distributed design strategies have been developed. While these strategies have yet to be tested on industry-scale projects, the experimental results in the present work should give some insight into how they can be tailored to the requirements of a specific problem.

5.1 Component Design-in-Context

The proposed component design-in-context method improves upon the traditional component design process by enabling components to design directly with respect to physical constraints in the residual system using information from embedded global FE models. In the development of the design-in-context method, the following has been achieved:

- An assessment of the ability of the embedded global FE model to predict global constraints and subsequently direct component design optimizations to satisfy them.
- A demonstration of the cost versus feasibility trade-off that can affect the choice of whether the embedded global FE model should be treated as a low fidelity model or a high fidelity model.
- The identification of a need to adjust constraint bounds to accommodate the different scale of responses in a mid-surface mesh.

An important limitation of the implementation suggested in the present work is that it was only tested on a relatively simple engine geometry with only a small number of components. Therefore a direction for future work is to investigate the performance of the design-in-context approach on more complex engine geometries or even on cross-organizational design studies such as an engine-nacelle optimization (Wang (2020)). Practical considerations about sharing geometry and requirements across organizations would thus need to be addressed.

5.2 Integration of Component Designs

The proposed safe integration method is designed to favour the preferences of component-level design processes and is in direct contrast to the rigid coordination-based approaches in systems engineering and MDO. The safe integration method in combination with component design-in-context produces a flexible and reliable design process that is able to accommodate for dynamic schedules with a low risk of rework. In the development of the safe integration method, the following has been achieved:

- An assessment of how the safe integration method reduces the number of infeasible system designs and hence reduces the overall cost of the design process relative to a direct integration method.
- An assessment of how an increase in the global model fidelity in the component design-in-context optimizations reduces the proportion of infeasible system designs and hence the probability of rework.
- An assessment of how an increase in the confidence level for the safe integration method reduces the overall cost of the design process because it requires less iterations to terminate at a feasible system design.
- A demonstration of a novel multi-fidelity variant for the safe integration method that uses constructs a Co-Kriging surrogate by compiling low fidelity global model evaluations from every component design process.
- A demonstration of how the performance of the safe integration approach is maintained as the problem dimensionality increases.

The amount of information reuse in the developed strategies are still relatively low. The inner-iterations of each component design-in-context process is an untapped resource of FE model simulations. The main obstacle preventing the combination of all component-level evaluations is that they are each their own fidelity, and that there is no clear ordering between them. A mid-surface engine with an embedded solid HPCC cannot be said to be more accurate than a mid-surface engine with an embedded solid HPTC.

A useful avenue for future work is therefore to investigate how these different embedded global model evaluations should be treated appropriately in a multi-fidelity approach. A naive approach that could very well work is to assign all component-level evaluations to the same fidelity. A more promising method could be to use Gaussian processes to define links between component evaluations before projecting them to the system-level space (Bonilla et al. (2007)). The difficulty that Kriging surrogates have when dealing with high-dimensional problems and with a large data set needs to also be further investigated to determine the viability of the safe integration approach when applied to realistic design processes.

Appendix A

Benchmark Problems

A.1 Bird Problem

The formulation for the two-variable Bird problem, as described in Mishra (2006), has been augmented with constraints g_1 and g_2. The resulting formulation is,

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \sin(x_1) \cdot \exp((1-\cos(x_2))^2) + \\ & \qquad \cos(x_1) \cdot \exp((1-\sin(x_2))^2) + (x_2 - x_1)^2, \\ \text{s.t.} \quad & g_1(\mathbf{x}) = 25 - (x_1+5)^2 - (x_2+5)^2 \leq 0, \\ & g_2(\mathbf{x}) = 15 - (x_1-1)^2 - (x_2-6)^2 \leq 0, \\ & \text{where} \quad -6 \leq x_1, x_2 \leq 6. \end{aligned} \tag{A.1}$$

This formulation is decomposed into the following two sub-problems,

$$\begin{aligned} \min_{x_1} \quad & f_1(x_1, \hat{\mathbf{x}}) = \cos(x_1) \cdot \exp((1-\sin(\hat{x}_2))^2), \\ \text{s.t.} \quad & g_1(x_1, \hat{\mathbf{x}}) = 25 - (x_1+5)^2 - (\hat{x}_2+5)^2 \leq 0, \\ & \text{where} \quad -6 \leq x_1 \leq 6. \end{aligned} \tag{A.2}$$

$$\begin{aligned} \min_{x_2} \quad & f_2(x_2, \hat{\mathbf{x}}) = \sin(\hat{x}_1) \cdot \exp((1-\cos(x_2))^2) + (x_2 - \hat{x}_1)^2, \\ \text{s.t.} \quad & g_2(x_2, \hat{\mathbf{x}}) = 15 - (\hat{x}_1-1)^2 - (x_2-6)^2 \leq 0, \\ & \text{where} \quad -6 \leq x_2 \leq 6. \end{aligned} \tag{A.3}$$

A.2 Rosen-Suzuki Problem

The formulation for the four-variable Rosen-Suzuki problem, as described in Hock and Schittkowski (1981), is,

$$\begin{aligned}
\min_{\mathbf{x}} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4, \\
\text{s.t.} \quad & g_1(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0, \\
& g_2(\mathbf{x}) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 + x_1 - x_4 - 10 \leq 0, \\
& g_3(\mathbf{x}) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \leq 0, \\
& \text{where} \quad -3 \leq x_1, x_2, x_3, x_4 \leq 3.
\end{aligned} \tag{A.4}$$

This formulation is decomposed into the following three sub-problems,

$$\begin{aligned}
\min_{x_1} \quad & f_1(x_1) = x_1^2 - 5x_1, \\
\text{s.t.} \quad & g_3(x_1, \hat{\mathbf{x}}) = 2x_1^2 + \hat{x}_2^2 + \hat{x}_3^2 + 2x_1 - \hat{x}_2 - \hat{x}_4 - 5 \leq 0, \\
& \text{where} \quad -3 \leq x_1 \leq 3.
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
\min_{x_2, x_4} \quad & f(x_2, x_4) = x_2^2 + x_4^2 - 5x_2 + 7x_4, \\
\text{s.t.} \quad & g_2(x_2, x_4, \hat{\mathbf{x}}) = \hat{x}_1^2 + 2x_2^2 + \hat{x}_3^2 + 2x_4^2 + \hat{x}_1 - x_4 - 10 \leq 0, \\
& \text{where} \quad -3 \leq x_2, x_4 \leq 3.
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
\min_{x_3} \quad & f_1(x_3) = 2x_3^2 - 21x_3, \\
\text{s.t.} \quad & g_1(x_3, \hat{\mathbf{x}}) = \hat{x}_1^2 + \hat{x}_2^2 + x_3^2 + \hat{x}_4^2 + \hat{x}_1 - \hat{x}_2 + x_3 - \hat{x}_4 - 8 \leq 0, \\
& \text{where} \quad -3 \leq x_3 \leq 3.
\end{aligned} \tag{A.7}$$

A.3 Weight Minimisation of a Speed Reducer

The formulation for the seven-variable speed reducer optimisation problem, as described in Kumar et al. (2020), is,

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & f(\mathbf{x}) = 0.7854x_2^2x_1(14.9334x_3 - 43.0934 + 3.3333x_3^2) + \\
& \qquad 0.7854(x_5x_7^2 + x_4x_6^2) - 1.508x_1(x_7^2 + x_6^2) + 7.477(x_7^3 + x_6^3), \\
\text{s.t.} \quad & g_1(\mathbf{x}) = -x_1x_2^2x_3 + 27 \leq 0, \\
& g_2(\mathbf{x}) = -x_1x_2^2x_3^2 + 397.5 \leq 0, \\
& g_3(\mathbf{x}) = -x_2x_6^4x_3x_4^{-3} + 1.93 \leq 0, \\
& g_4(\mathbf{x}) = -x_2x_7^4x_3x_5^{-3} + 1.93 \leq 0, \\
& g_5(\mathbf{x}) = 10x_6^{-3}\sqrt{16.91 \times 10^6 + (745x_4x_2^{-1}x_3^{-1})^2} - 1100 \leq 0, \\
& g_6(\mathbf{x}) = 10x_7^{-3}\sqrt{157.5 \times 10^6 + (745x_5x_2^{-1}x_3^{-1})^2} - 850 \leq 0, \\
& g_7(\mathbf{x}) = x_2x_3 - 40 \leq 0, \\
& g_8(\mathbf{x}) = -x_1x_2^{-1} + 5 \leq 0, \\
& g_9(\mathbf{x}) = x_1x_2^{-1} - 12 \leq 0, \\
& g_{10}(\mathbf{x}) = 1.5x_6 - x_4 + 1.9 \leq 0, \\
& g_{11}(\mathbf{x}) = 1.1x_7 - x_5 + 1.9 \leq 0, \\
\text{where} \quad & 2.6 \leq x_1 \leq 3.6, \quad 0.7 \leq x_2 \leq 0.8, \quad 17 \leq x_3 \leq 28 \\
& 7.3 \leq x_4, x_5 \leq 8.3, \quad 2.9 \leq x_6 \leq 3.9, \quad 5.0 \leq x_7 \leq 5.5.
\end{aligned}
\tag{A.8}
$$

This formulation is decomposed into the following five sub-problems,

$$
\begin{aligned}
\min_{x_1} \quad & f_1(x_1, \hat{\mathbf{x}}) = 0.7854 \cdot 14.9334\hat{x}_2^2x_1\hat{x}_3, \\
\text{s.t.} \quad & g_8(x_1, \hat{\mathbf{x}}) = -x_1\hat{x}_2^{-1} + 5 \leq 0, \\
& g_9(x_1, \hat{\mathbf{x}}) = x_1\hat{x}_2^{-1} - 12 \leq 0, \\
& \text{where} \quad 2.6 \leq x_1 \leq 3.6.
\end{aligned}
\tag{A.9}
$$

$$
\begin{aligned}
\min_{x_2} \quad & f_2(x_2, \hat{\mathbf{x}}) = -0.7854 \cdot 43.0934x_2\hat{x}_1, \\
\text{s.t.} \quad & g_1(x_2, \hat{\mathbf{x}}) = -\hat{x}_1x_2^2\hat{x}_3 + 27 \leq 0, \\
& g_2(x_2, \hat{\mathbf{x}}) = -\hat{x}_1x_2^2\hat{x}_3^2 + 397.5 \leq 0, \\
& \text{where} \quad 0.7 \leq x_2 \leq 0.8.
\end{aligned}
\tag{A.10}
$$

$$\begin{aligned} \min_{x_3} \quad & f_3(x_3, \hat{\mathbf{x}}) = 0.7854 \cdot 3.3333 \hat{x}_2^2 \hat{x}_1 x_3^2, \\ \text{s.t.} \quad & g_7(x_3, \hat{\mathbf{x}}) = \hat{x}_2 x_3 - 40 \leq 0, \\ & \text{where} \quad 17 \leq x_3 \leq 28. \end{aligned} \tag{A.11}$$

$$\begin{aligned} \min_{x_4, x_5} \quad & f_4(x_4, x_5, \hat{\mathbf{x}}) = 0.7854 x_5 \hat{x}_7^2 + 0.7854 x_4 \hat{x}_6^2, \\ \text{s.t.} \quad & g_3(x_4, \hat{\mathbf{x}}) = -\hat{x}_2 \hat{x}_6^4 \hat{x}_3 x_4^{-3} + 1.93 \leq 0, \\ & g_4(x_5, \hat{\mathbf{x}}) = -\hat{x}_2 \hat{x}_7^4 \hat{x}_3 x_5^{-3} + 1.93 \leq 0, \\ & g_{10}(x_4, \hat{\mathbf{x}}) = 1.5 \hat{x}_6 - x_4 + 1.9 \leq 0, \\ & g_{11}(x_5, \hat{\mathbf{x}}) = 1.5 \hat{x}_7 - x_5 + 1.9 \leq 0, \\ & \text{where} \quad 7.3 \leq x_4, x_5 \leq 8.3. \end{aligned} \tag{A.12}$$

$$\begin{aligned} \min_{x_6, x_7} \quad & f_4(x_6, x_7, \hat{\mathbf{x}}) = -1.508 \hat{x}_1 (x_7^2 + x_6^2) + 7.477 (x_7^3 + x_6^3), \\ \text{s.t.} \quad & g_5(x_6, \hat{\mathbf{x}}) = 10 x_6^{-3} \sqrt{16.91 \times 10^6 + (745 \hat{x}_4 \hat{x}_2^{-1} \hat{x}_3^{-1})^2} - 1100 \leq 0, \\ & g_6(x_7, \hat{\mathbf{x}}) = 10 x_7^{-3} \sqrt{157.5 \times 10^6 + (745 \hat{x}_5 \hat{x}_2^{-1} \hat{x}_3^{-1})^2} - 850 \leq 0, \\ & \text{where} \quad 2.9 \leq x_6 \leq 3.9, \quad 5.0 \leq x_7 \leq 5.5. \end{aligned} \tag{A.13}$$

A.4 Optimal Design of an Industrial Refrigeration System

The formulation for the fourteen-variable industrial refrigeration system optimisation problem, as described in Kumar et al. (2020), is,

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & f(\mathbf{x}) = 63098.88x_2x_4x_{12} + 5441.5x_2^2x_{12} + 115055.5x_2^{1.664}x_6 + \\
& \quad 6172.27x_2^2x_6 + 63098.88x_1x_3x_{11} + 5441.5x_1^2x_{11} + \\
& \quad 115055.5x_1^{1.664}x_5 + 6172.27x_1^2x_5 + 140.53x_1x_{11} + \\
& \quad 281.29x_3x_{11} + 70.26x_1^2 + 281.29x_1x_3 + 281.29x_3^2 + \\
& \quad 14437x_8^{1.8812}x_{12}^{0.3424}x_{10}x_{14}^{-1}x_1^2x_7x_9^{-1} + 20470.2x_7^{2.893}x_{11}^{0.316}x_1^2, \\
\text{s.t.} \quad & g_1(\mathbf{x}) = 1.524x_7{-}1 - 1 \leq 0, \\
& g_2(\mathbf{x}) = 1.524x_8{-}1 - 1 \leq 0, \\
& g_3(\mathbf{x}) = 0.07789x_1 - 2x_7^{-1}x_9 - 1 \leq 0, \\
& g_4(\mathbf{x}) = 7.0530x_9^{-1}x_1^2x_{10}x_8^{-1}x_2^{-1}x_{14}^{-1} - 1 \leq 0, \\
& g_5(\mathbf{x}) = 0.0833x_{13}^{-1}x_{14} - 1 \leq 0, \\
& g_6(\mathbf{x}) = 47.136x_2^{0.333}x_{10}x_{12} - 1.333x_8x_{13}^{2.1195} + 62.08x_{13}^{2.1195}x_1 2^{-1}x_8^{0.2}x_{10}^{-1} \leq 0, \\
& g_7(\mathbf{x}) = 0.04771x_{10}x_8^{1.8812}x_{12}^{0.3424} - 1 \leq 0, \\
& g_8(\mathbf{x}) = 0.0488x_9x_7^{1.893}x_{11}^{0.316} - 1 \leq 0, \\
& g_9(\mathbf{x}) = 0.0099x_1x_3^{-1} - 1 \leq 0, \\
& g_{10}(\mathbf{x}) = 0.0193x_2x_4{-}1 - 1 \leq 0, \\
& g_{11}(\mathbf{x}) = 0.0298x_1x_5^{-1} - 1 \leq 0, \\
& g_{12}(\mathbf{x}) = 0.056x_2x_6^{-1} - 1 \leq 0, \\
& g_{13}(\mathbf{x}) = 2x_9^{-1} - 1 \leq 0, \\
& g_{14}(\mathbf{x}) = 2x_{10}^{-1} - 1 \leq 0, \\
& g_{15}(\mathbf{x}) = x_{12}x_{11}^{-1} - 1 \leq 0, \\
& \text{where} \quad 0.001 \leq x_1, \ldots, x_{14} \leq 5.0.
\end{aligned}
\tag{A.14}
$$

This formulation is decomposed into the following six sub-problems,

$$
\begin{aligned}
\min_{x_1} \quad & f_1(x_1, \hat{\mathbf{x}}) = 5441.5x_1^2\hat{x}_{11} + 115055.5x_1^{1.664}\hat{x}_5 + 6172.27x_1^2\hat{x}_5 + 70.26x_1^2, \\
\text{s.t.} \quad & g_3(x_1, \hat{\mathbf{x}}) = 0.07789x_1 - 2\hat{x}_7^{-1}\hat{x}_9 - 1 \leq 0, \\
& g_4(x_1, \hat{\mathbf{x}}) = 7.0530\hat{x}_9^{-1}x_1^2\hat{x}_{10}\hat{x}_8^{-1}\hat{x}_2^{-1}\hat{x}_{14}^{-1} - 1 \leq 0, \\
& \quad \text{where} \quad 0.001 \leq x_1 \leq 5.0.
\end{aligned}
\tag{A.15}
$$

$$\begin{aligned}
\min_{x_2,x_{13}} \quad & f_2(x_2,\hat{\mathbf{x}}) = 5441.5x_2^2\hat{x}_{12} + 115055.5x_2^{1.664}\hat{x}_6 + 6172.27x_2^2\hat{x}_6, \\
\text{s.t.} \quad & g_6(x_2,x_{13},\hat{\mathbf{x}}) = 47.136x_2^{0.333}\hat{x}_{10}\hat{x}_{12} - 1.333\hat{x}_8x_{13}^{2.1195} + 62.08x_{13}^{2.1195}\hat{x}_1 2^{-1}\hat{x}_8^{0.2}\hat{x}_{10}^{-1} \leq 0, \\
& \text{where} \quad 0.001 \leq x_2, x_{13} \leq 5.0.
\end{aligned} \tag{A.16}$$

$$\begin{aligned}
\min_{x_3,x_4,x_5,x_6} \quad & f_3(x_3,\hat{\mathbf{x}}) = 63098.88\hat{x}_1x_3\hat{x}_{11} + 281.29\hat{x}_1x_3 + 281.29x_3^2, \\
\text{s.t.} \quad & g_9(x_3,\hat{\mathbf{x}}) = 0.0099\hat{x}_1x_3^{-1} - 1 \leq 0, \\
& g_{10}(x_4,\hat{\mathbf{x}}) = 0.0193\hat{x}_2x_4{-}1 - 1 \leq 0, \\
& g_{11}(x_5,\hat{\mathbf{x}}) = 0.0298\hat{x}_1x_5^{-1} - 1 \leq 0, \\
& g_{12}(x_6,\hat{\mathbf{x}}) = 0.056\hat{x}_2x_6^{-1} - 1 \leq 0, \\
& \text{where} \quad 0.001 \leq x_3, x_4, x_5, x_6 \leq 5.0.
\end{aligned} \tag{A.17}$$

$$\begin{aligned}
\min_{x_7,x_8} \quad & f_4(x_7,\hat{\mathbf{x}}) = 20470.2x_7^{2.893}\hat{x}_{11}^{0.316}\hat{x}_1^2, \\
\text{s.t.} \quad & g_1(x_7,\hat{\mathbf{x}}) = 1.524x_7{-}1 - 1 \leq 0, \\
& g_2(x_8,\hat{\mathbf{x}}) = 1.524x_8{-}1 - 1 \leq 0, \\
& \text{where} \quad 0.001 \leq x_7, x_8 \leq 5.0.
\end{aligned} \tag{A.18}$$

$$\begin{aligned}
\min_{x_9,x_{10},x_{14}} \quad & f_5(x_9,x_{10},x_{14},\hat{\mathbf{x}}) = 14437\hat{x}_8^{1.8812}\hat{x}_{12}^{0.3424}x_{10}x_{14}^{-1}\hat{x}_1^2\hat{x}_7x_9^{-1}, \\
\text{s.t.} \quad & g_5(x_{14},\hat{\mathbf{x}}) = 0.0833\hat{x}_{13}^{-1}x_{14} - 1 \leq 0, \\
& g_{13}(x_9,\hat{\mathbf{x}}) = 2x_9^{-1} - 1 \leq 0, \\
& g_{14}(x_{10},\hat{\mathbf{x}}) = 2x_{10}^{-1} - 1 \leq 0, \\
& \text{where} \quad 0.001 \leq x_9, x_{10}, x_{14} \leq 5.0.
\end{aligned} \tag{A.19}$$

$$\begin{aligned}
\min_{x_{11},x_{12}} \quad & f_6(x_{11},x_{12},\hat{\mathbf{x}}) = 63098.88\hat{x}_2\hat{x}_4x_{12} + 140.53\hat{x}_1x_{11} + 281.29\hat{x}_3x_{11}, \\
\text{s.t.} \quad & g_7(x_{12},\hat{\mathbf{x}}) = 0.04771\hat{x}_{10}\hat{x}_8^{1.8812}x_{12}^{0.3424} - 1 \leq 0, \\
& g_8(x_{11},\hat{\mathbf{x}}) = 0.0488\hat{x}_9\hat{x}_7^{1.893}x_{11}^{0.316} - 1 \leq 0, \\
& g_{15}(x_{11},x_{12},\hat{\mathbf{x}}) = x_{12}x_{11}^{-1} - 1 \leq 0, \\
& \text{where} \quad 0.001 \leq x_{11}, x_{12} \leq 5.0.
\end{aligned} \tag{A.20}$$

Appendix B

Optimization Results

Table B.1: Optimization results for the CRESCENDO HPCC case study in Section 3.3.2. This table of results is the complete version of Table 3.4. The objective values have been normalized against the nominal design. The constraint values have been normalized against the constraint upper bounds, so a value ≥ 1 denotes a violation. μ is the average value over 10 repeated runs. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

Parameter	Strategy I		Strategy II		Strategy III	
	μ	95% CI	μ	95% CI	μ	95% CI
Objective						
HPCC.OBJ	0.9442	0.9440 - 0.9447	0.9439	0.9437 - 0.9441	0.9445	0.9442 - 0.9449
Constraints						
HPCC.CON1	0.9954	0.9942 - 0.9963	0.9954	0.9931 - 0.9970	0.9944	0.9928 - 0.9958
HPCC.CON2	0.9947	0.9937 - 0.9955	0.9948	0.9929 - 0.9962	0.9939	0.9925 - 0.9951
HPCC.CON3	0.9975	0.9964 - 0.9983	0.9977	0.9958 - 0.9991	0.9966	0.9952 - 0.9979
HPCC.CON4	0.9963	0.9955 - 0.9968	0.9966	0.9954 - 0.9974	0.9957	0.9948 - 0.9966
HPCC.CON5	0.9985	0.9979 - 0.9990	0.9990	0.9984 - 0.9999	0.9986	0.9979 - 0.9991
HPCC.CON6	0.9995	0.9987 - 0.9999	1.0000	0.9996 - 1.0003	0.9991	0.9984 - 0.9996
HPCC.CON7	0.9993	0.9987 - 0.9997	0.9998	0.9994 - 1.0003	0.9992	0.9984 - 0.9996

Table B.2: Optimization results for the CRESCENDO engine case study in Section 3.3.3. This table of results is the complete version of Table 3.8. The objective values have been normalized against the nominal design. The constraint values have been normalized against the constraint upper bounds, so a value ≥ 1 denotes a violation. μ is the average value over 10 repeated runs. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

Parameter	Strategy I		Strategy II		Strategy III	
	μ	95% CI	μ	95% CI	μ	95% CI
Objective						
ENGINE.OBJ	0.9816	0.9813 - 0.9819	0.9802	0.9800 - 0.9804	0.9820	0.9816 - 0.9823
Constraints						
FC.CON1	0.9991	0.9980 - 0.9997	1.0021	1.0018 - 1.0023	0.9973	0.9956 - 0.9987
FC.CON2	0.9972	0.9969 - 0.9975	0.9996	0.9993 - 1.0000	0.9963	0.9954 - 0.9971
FC.CON3	0.9981	0.9973 - 0.9989	1.0011	1.0007 - 1.0015	0.9984	0.9977 - 0.9990
HPCC.CON1	0.9895	0.9874 - 0.9922	0.9945	0.9916 - 0.9972	0.9923	0.9898 - 0.9947
HPCC.CON2	0.9889	0.9871 - 0.9914	0.9936	0.9912 - 0.9959	0.9915	0.9892 - 0.9935
HPCC.CON3	0.9898	0.9873 - 0.9930	0.9960	0.9937 - 0.9983	0.9933	0.9905 - 0.9954
HPCC.CON4	0.9890	0.9868 - 0.9917	0.9939	0.9925 - 0.9954	0.9924	0.9904 - 0.9940
HPCC.CON5	0.9937	0.9885 - 0.9968	0.9984	0.9961 - 1.0002	0.9978	0.9960 - 0.9989
HPCC.CON6	0.9881	0.9841 - 0.9916	0.9960	0.9947 - 0.9970	0.9930	0.9903 - 0.9952
HPCC.CON7	0.9880	0.9831 - 0.9916	0.9959	0.9939 - 0.9977	0.9927	0.9900 - 0.9950
HPTC.CON1	0.9684	0.9644 - 0.9738	0.9774	0.9687 - 0.9896	0.9605	0.9411 - 0.9687
HPTC.CON2	0.9745	0.9626 - 0.9833	0.9782	0.9657 - 0.9904	0.9557	0.9162 - 0.9750
LPTC.CON1	0.9753	0.9455 - 0.9866	0.9702	0.9424 - 0.9803	0.9517	0.9107 - 0.9787
LPTC.CON2	0.9489	0.9441 - 0.9538	0.9557	0.9496 - 0.9647	0.9402	0.9196 - 0.9559
LPTC.CON3	0.9598	0.9566 - 0.9628	0.9684	0.9635 - 0.9754	0.9590	0.9505 - 0.9687

Table B.3: Optimization results for the CRESCENDO HPCC design-in-context case study in Section 3.4.1. This table of results is the complete version of Table 3.11. The objective values have been normalized against the nominal design. The constraint values have been normalized against the constraint upper bounds, so a value ≥ 1 denotes a violation. μ is the average value over the 10 repeated runs. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

Parameter	Strategy I		Strategy II		Strategy III	
	μ	95% CI	μ	95% CI	μ	95% CI
Objective						
HPCC.OBJ	0.9466	0.9463 - 0.9469	0.9442	0.9440 - 0.9447	0.9465	0.9461 - 0.9479
Constraints						
FC.CON1	0.9920	0.9918 - 0.9921	0.9919	0.9917 - 0.9920	0.9919	0.9912 - 0.9921
FC.CON2	0.9921	0.9919 - 0.9922	0.9918	0.9915 - 0.9919	0.9920	0.9912 - 0.9922
FC.CON3	0.9928	0.9925 - 0.9930	0.9927	0.9923 - 0.9929	0.9927	0.9916 - 0.9930
HPCC.CON1	0.9988	0.9977 - 0.9993	0.9954	0.9943 - 0.9963	0.9982	0.9945 - 0.9993
HPCC.CON2	0.9974	0.9964 - 0.9979	0.9947	0.9938 - 0.9955	0.9969	0.9937 - 0.9978
HPCC.CON3	0.9991	0.9981 - 0.9997	0.9975	0.9964 - 0.9983	0.9987	0.9950 - 0.9997
HPCC.CON4	0.9967	0.9960 - 0.9971	0.9964	0.9955 - 0.9968	0.9964	0.9938 - 0.9972
HPCC.CON5	0.9931	0.9927 - 0.9933	0.9985	0.9980 - 0.9990	0.9934	0.9928 - 0.9937
HPCC.CON6	0.9965	0.9960 - 0.9969	0.9995	0.9987 - 0.9999	0.9965	0.9945 - 0.9972
HPCC.CON7	0.9942	0.9938 - 0.9944	0.9993	0.9987 - 0.9997	0.9943	0.9933 - 0.9948
HPTC.CON1	0.9866	0.9856 - 0.9875	1.0032	1.0008 - 1.0050	0.9878	0.9865 - 0.9898
HPTC.CON2	0.9990	0.9969 - 0.9998	1.0262	1.0232 - 1.0286	1.0005	0.9983 - 1.0018
LPTC.CON1	0.9860	0.9851 - 0.9873	1.0007	0.9984 - 1.0025	0.9872	0.9859 - 0.9902
LPTC.CON2	0.9856	0.9852 - 0.9861	0.9918	0.9908 - 0.9925	0.9861	0.9856 - 0.9873
LPTC.CON3	0.9879	0.9878 - 0.9882	0.9910	0.9905 - 0.9913	0.9882	0.9879 - 0.9888

Table B.4: Optimization results for the multi-fidelity CRESCENDO HPCC design-in-context case study in Section 3.4.2. This table of results is the complete version of Table 3.13. The objective values have been normalized against the nominal design. The constraint values have been normalized against the constraint upper bounds, so a value ≥ 1 denotes a violation. μ is the average value over the 10 repeated runs. The confidence intervals were calculated through resampling with 50,000 bootstrap samples.

Parameter	Strategy I		Strategy II		Strategy III	
	μ	95% CI	μ	95% CI	μ	95% CI
Objective						
HPCC.OBJ	0.9466	0.9463 - 0.9469	0.9520	0.9463 - 0.9741	0.9578	0.9473 - 0.9786
Constraints						
FC.CON1	0.9920	0.9918 - 0.9921	0.9913	0.9884 - 0.9921	0.9905	0.9885 - 0.9917
FC.CON2	0.9921	0.9919 - 0.9922	0.9914	0.9885 - 0.9921	0.9906	0.9885 - 0.9917
FC.CON3	0.9928	0.9925 - 0.9930	0.9917	0.9871 - 0.9929	0.9905	0.9871 - 0.9923
HPCC.CON1	0.9988	0.9977 - 0.9993	0.9960	0.9848 - 0.9991	0.9926	0.9848 - 0.9973
HPCC.CON2	0.9974	0.9964 - 0.9979	0.9948	0.9845 - 0.9976	0.9918	0.9844 - 0.9960
HPCC.CON3	0.9991	0.9981 - 0.9997	0.9958	0.9818 - 0.9995	0.9919	0.9819 - 0.9975
HPCC.CON4	0.9967	0.9960 - 0.9971	0.9940	0.9826 - 0.9970	0.9909	0.9827 - 0.9955
HPCC.CON5	0.9931	0.9927 - 0.9933	0.9905	0.9793 - 0.9935	0.9882	0.9808 - 0.9929
HPCC.CON6	0.9965	0.9960 - 0.9969	0.9930	0.9779 - 0.9969	0.9894	0.9776 - 0.9955
HPCC.CON7	0.9942	0.9938 - 0.9944	0.9909	0.9771 - 0.9946	0.9879	0.9765 - 0.9936
HPTC.CON1	0.9866	0.9856 - 0.9875	0.9834	0.9718 - 0.9874	0.9818	0.9714 - 0.9881
HPTC.CON2	0.9990	0.9969 - 0.9998	0.9878	0.9397 - 1.0006	0.9784	0.9378 - 0.9988
LPTC.CON1	0.9860	0.9851 - 0.9873	0.9847	0.9792 - 0.9868	0.9847	0.9799 - 0.9892
LPTC.CON2	0.9856	0.9852 - 0.9861	0.9850	0.9824 - 0.9859	0.9849	0.9825 - 0.9868
LPTC.CON3	0.9879	0.9878 - 0.9882	0.9876	0.9864 - 0.9881	0.9876	0.9864 - 0.9885

Appendix C

Response Surface Plots

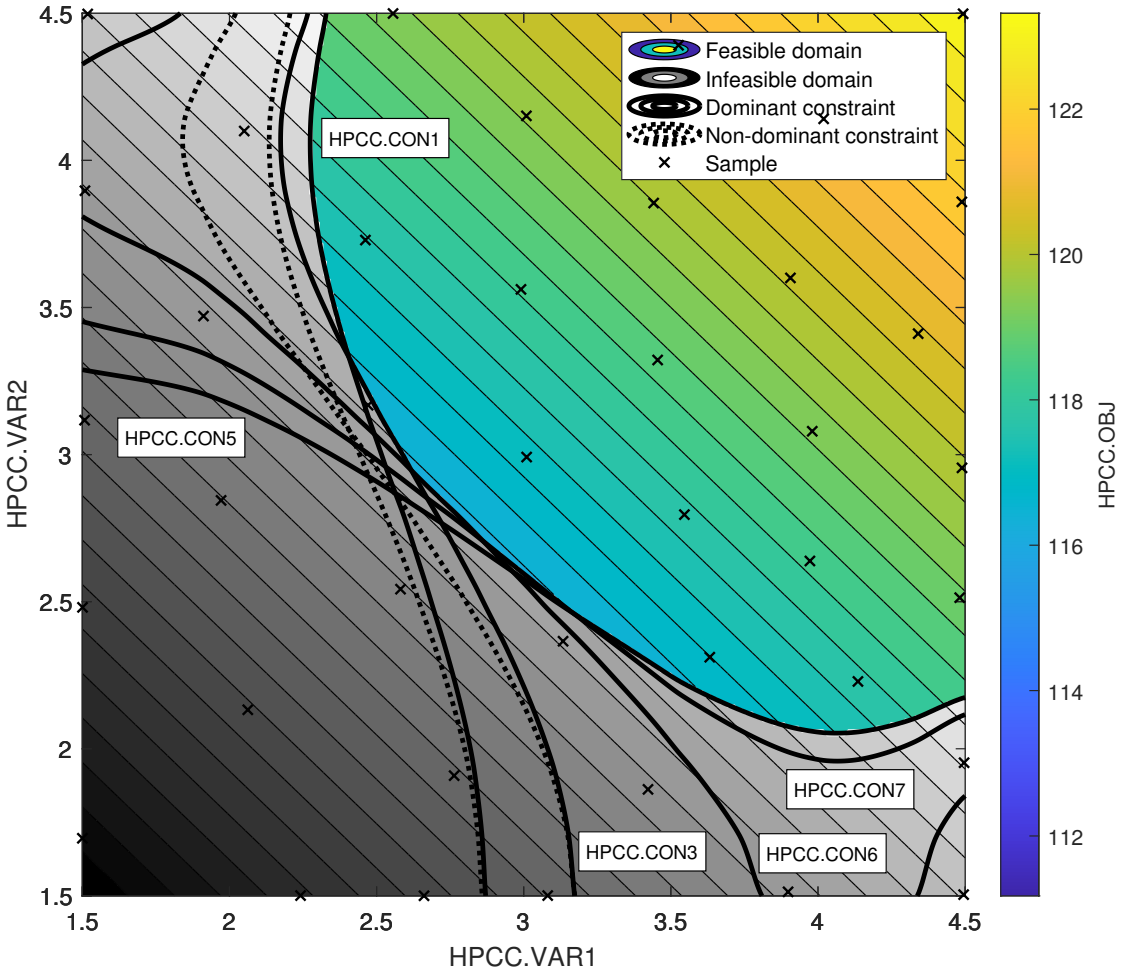


Figure C.1: Response surface of the CRESCENDO HPCC space-filling DOE in Section 3.3.2. This plot is the detailed version of Figure 3.3a. The dominant constraints have been labelled.

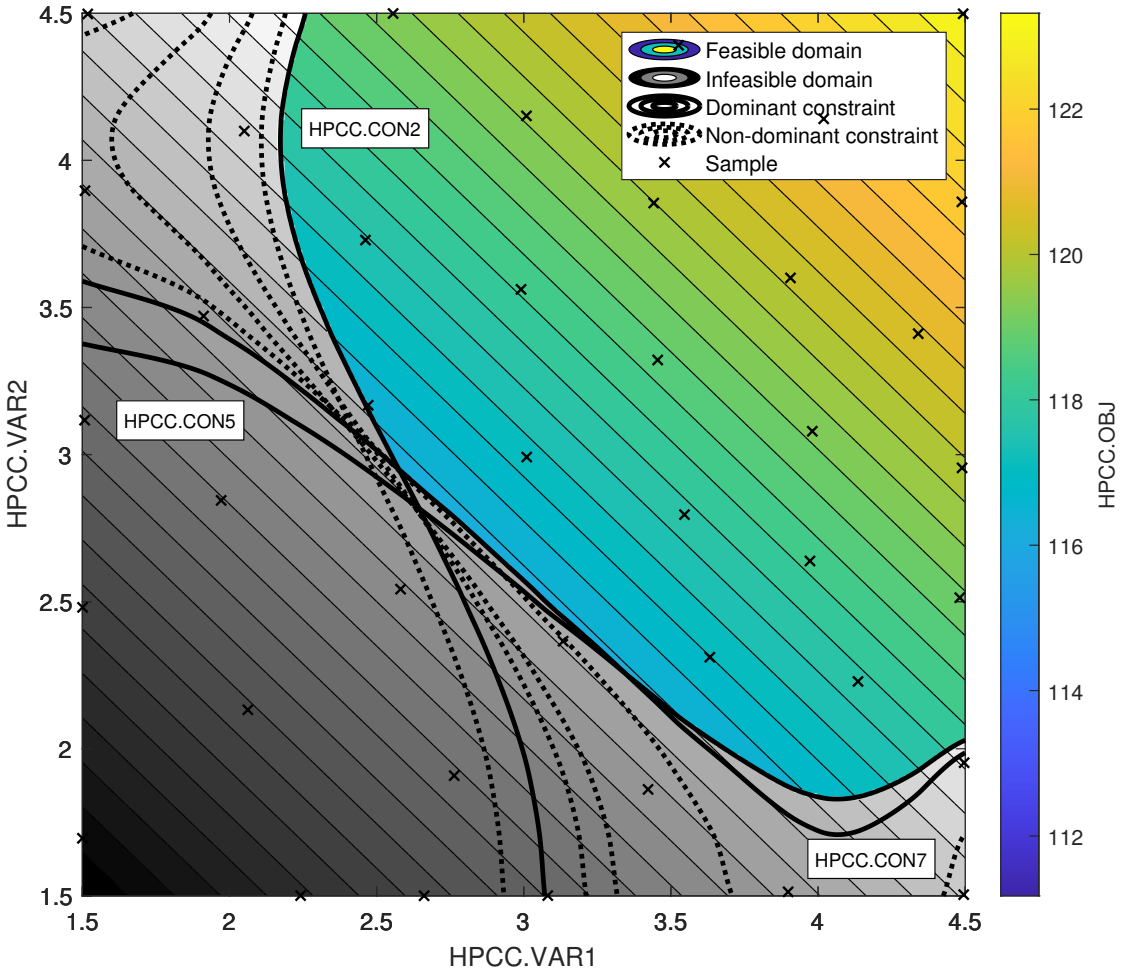


Figure C.2: Response surface of the CRESCENDO HP compressor casing optimization case study in Section 3.3.2. This plot is the detailed version of Figure 3.3b. The dominant constraints have been labelled.

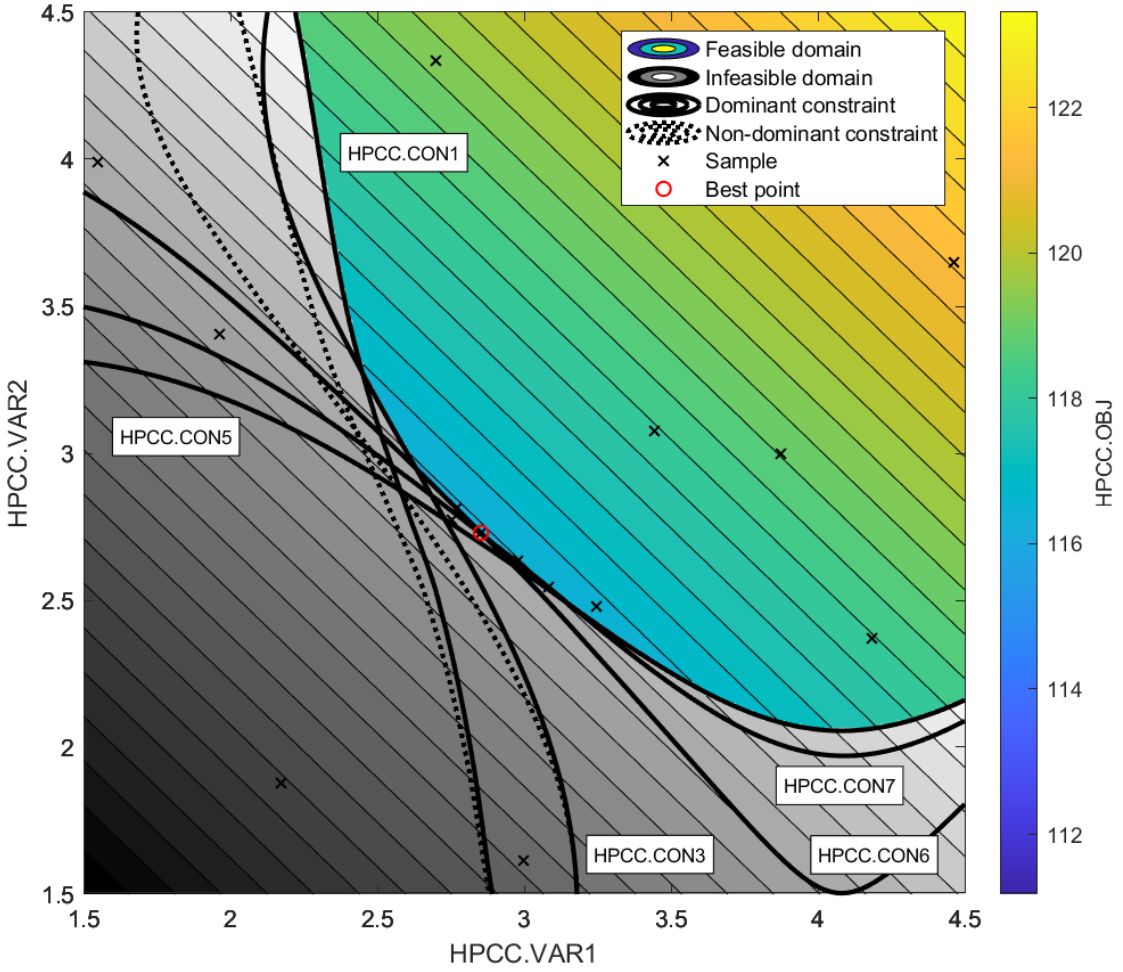


Figure C.3: Response surface of the CRESCENDO HPCC optimization problem in Section 3.3.2. This plot is the detailed version of Figure 3.4a. The dominant constraints have been labelled.

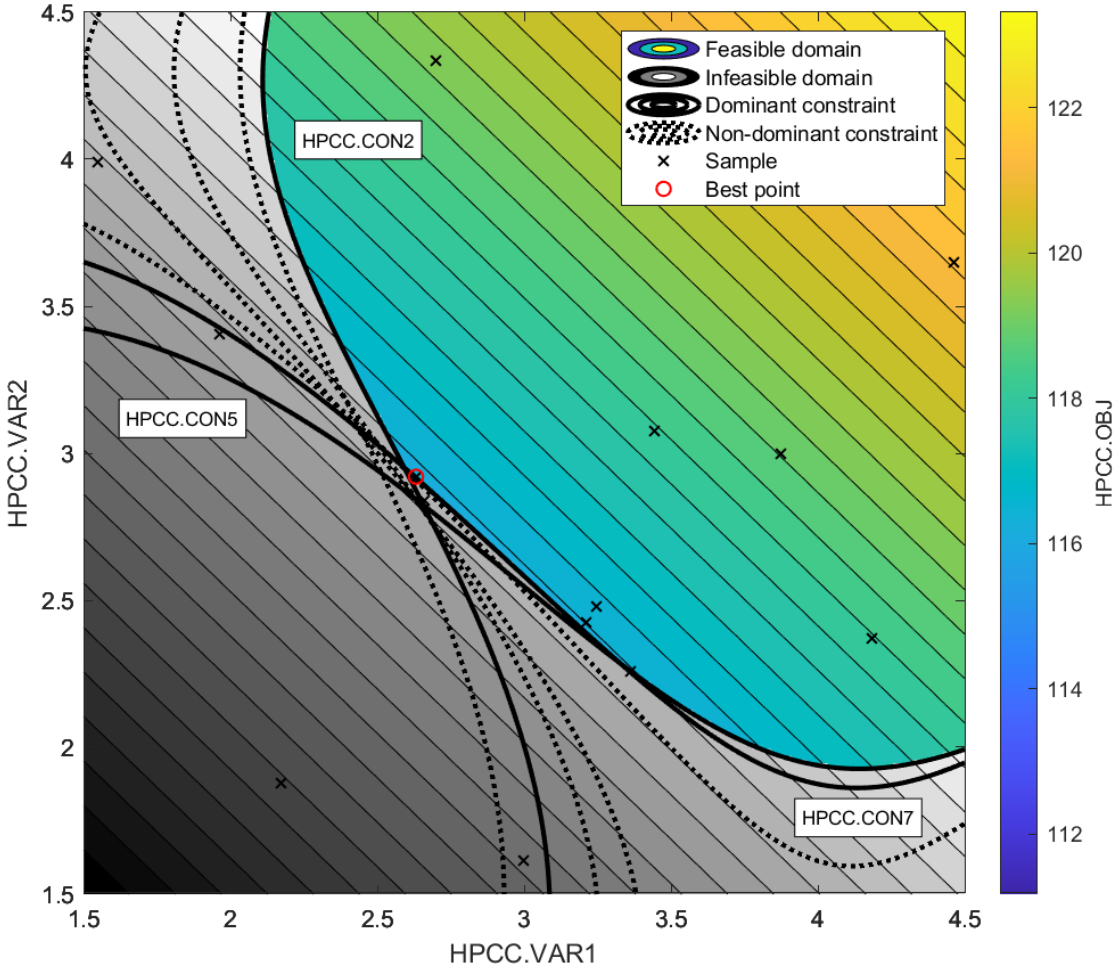


Figure C.4: Response surface of the CRESCENDO HPCC optimization problem in Section 3.3.2. This plot is the detailed version of Figure 3.4b. The dominant constraints have been labelled.

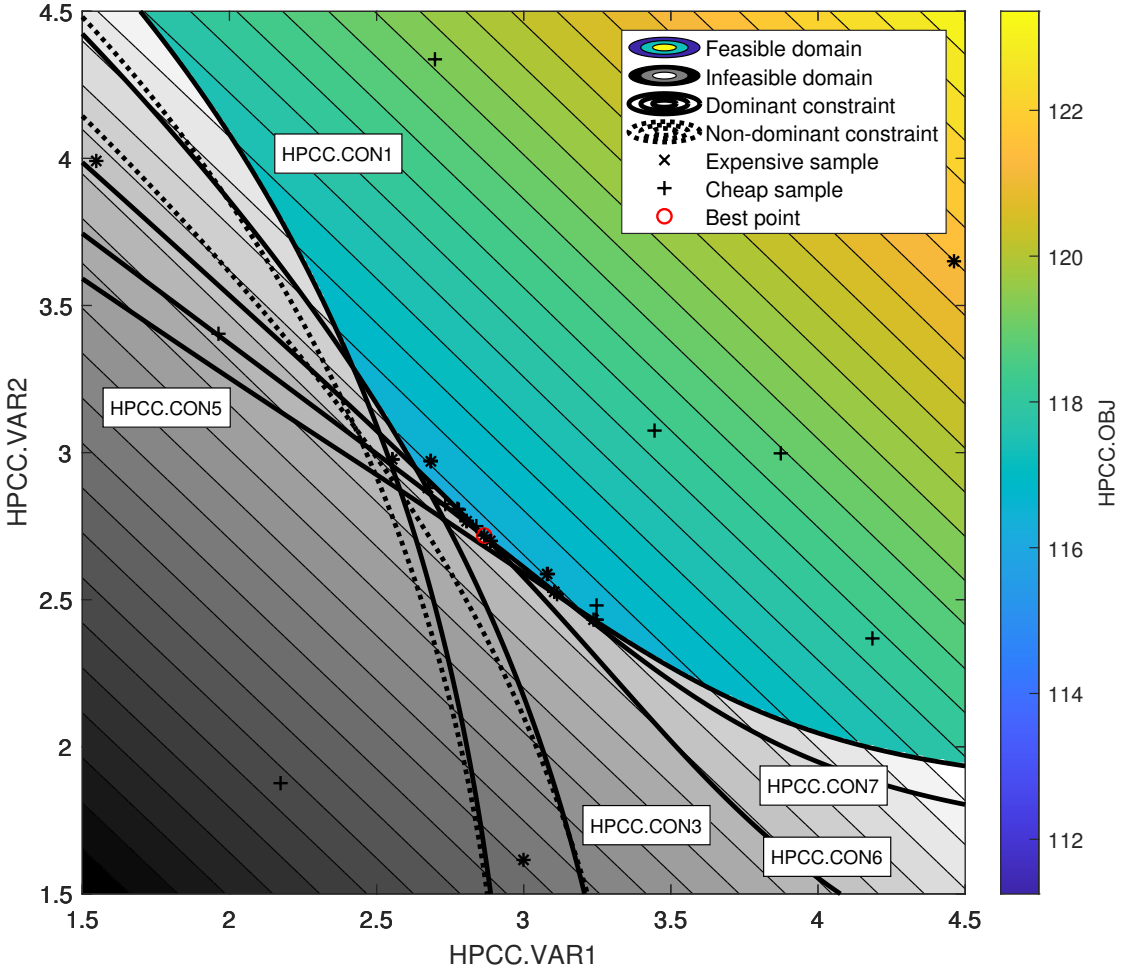


Figure C.5: Response surface of the CRESCENDO HPCC optimization problem in Section 3.3.2. This plot is the detailed version of Figure 3.4c. The dominant constraints have been labelled.

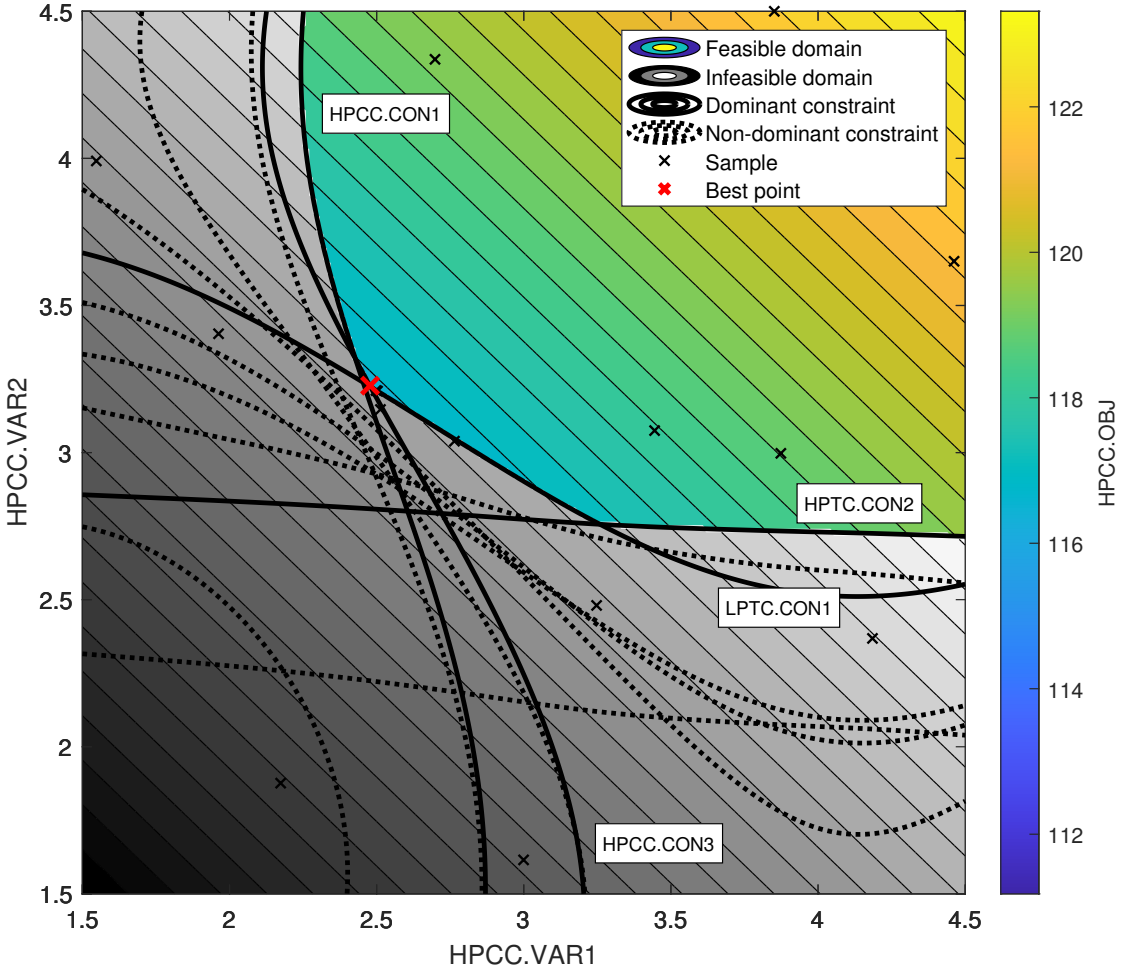


Figure C.6: Response surface of the CRESCENDO HPCC design-in-context optimization problem in Section 3.4.1. This plot is the detailed version of Figure 3.8a. The dominant constraints have been labelled.

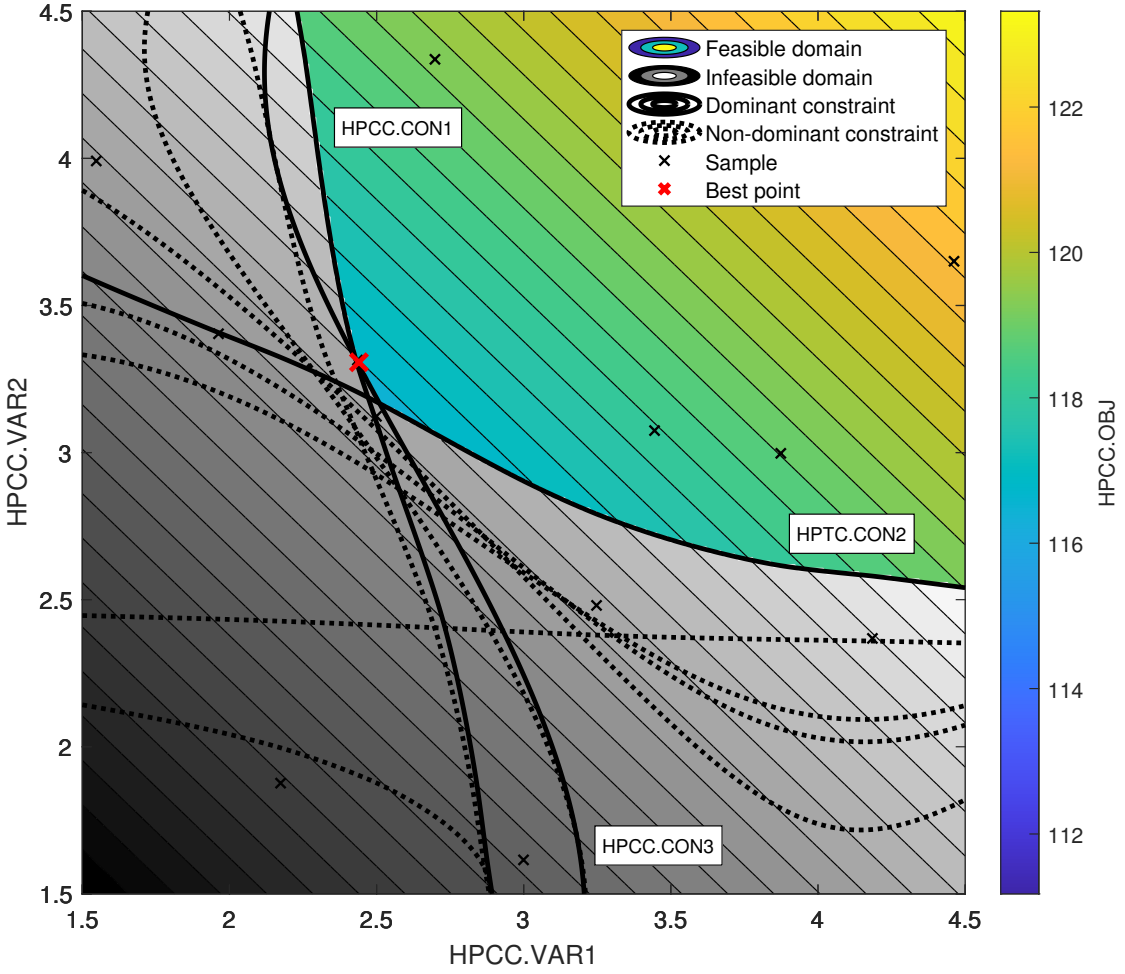


Figure C.7: Response surface of the CRESCENDO HPCC design-in-context optimization problem in Section 3.4.1. This plot is the detailed version of Figure 3.8b. The dominant constraints have been labelled.

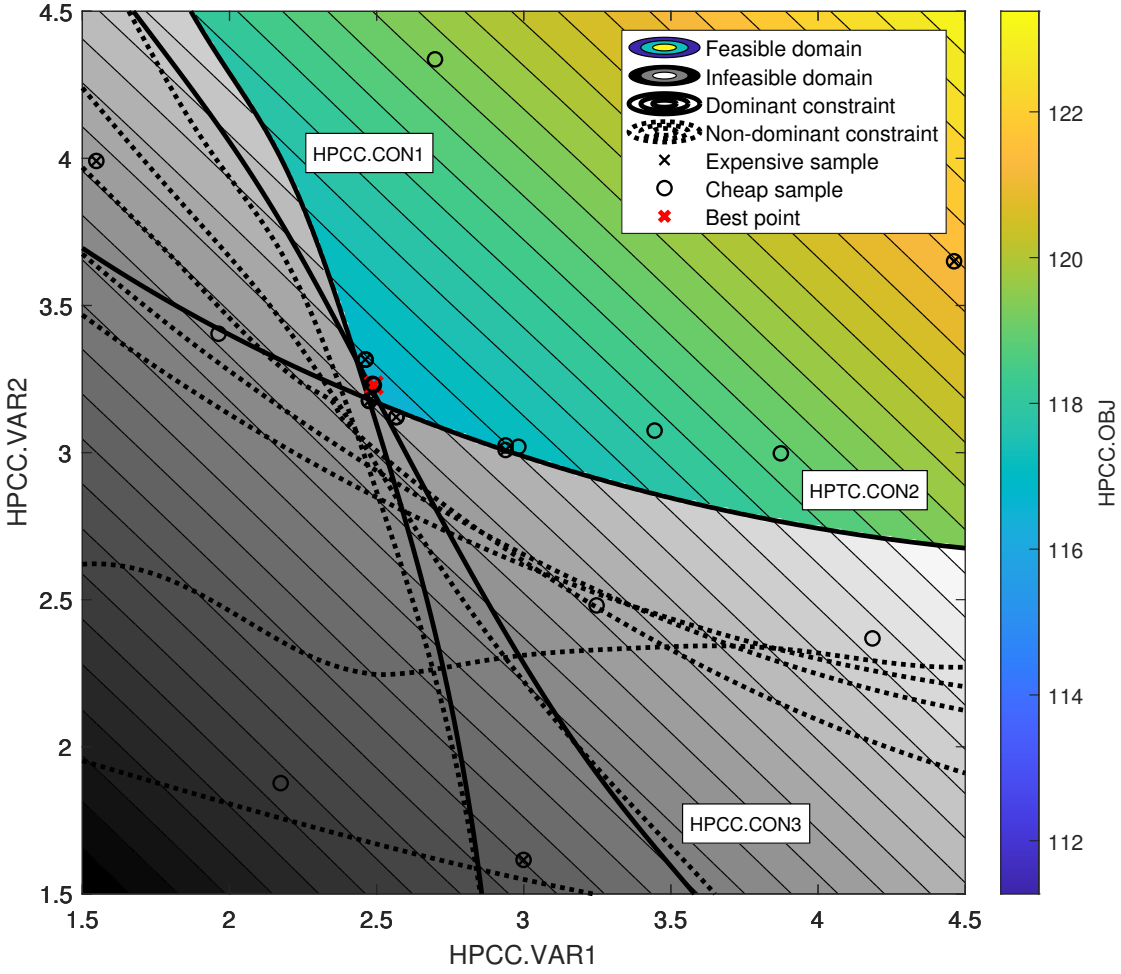


Figure C.8: Response surface of the multi-fidelity CRESCENDO HPCC design-in-context optimization problem in Section 3.4.2. This plot is the detailed version of Figure 3.10a. The dominant constraints have been labelled.

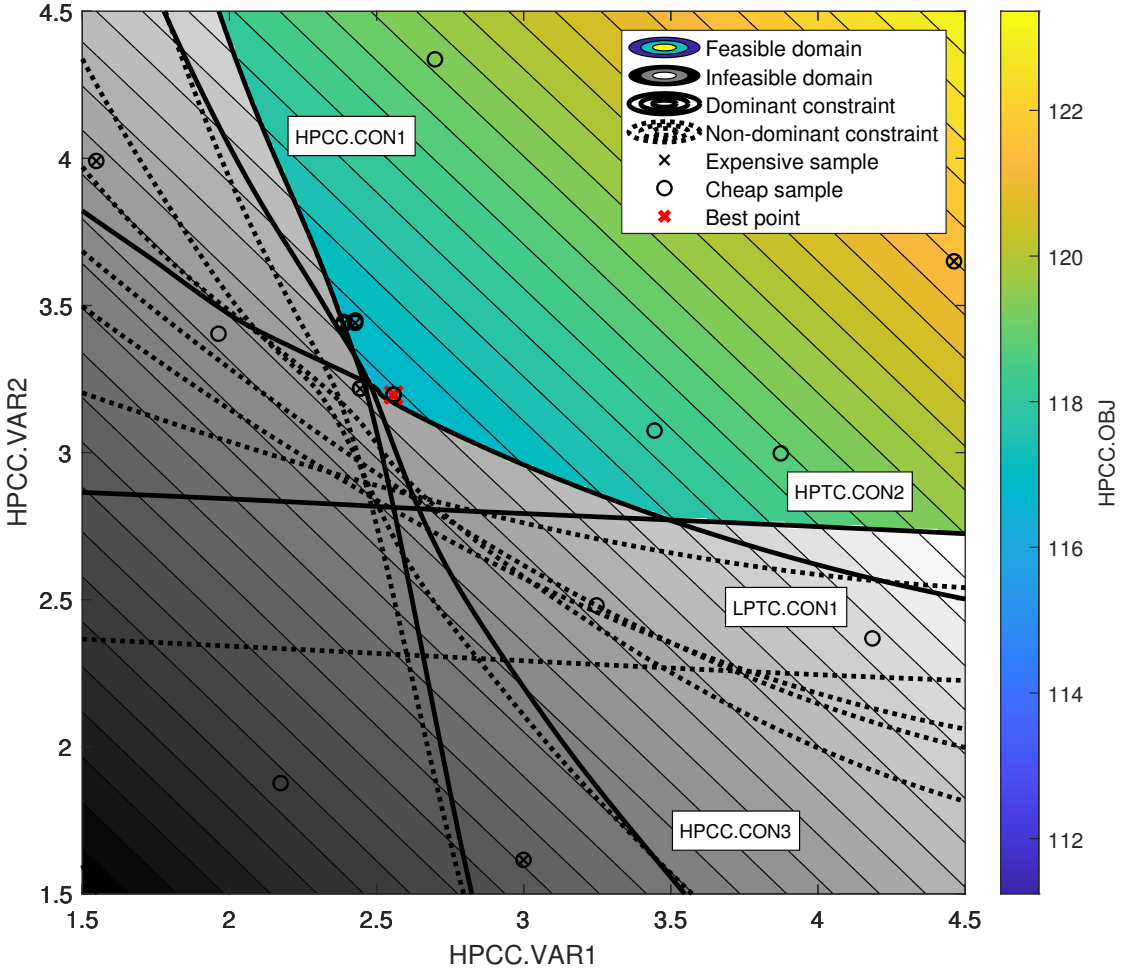


Figure C.9: Response surface of the multi-fidelity CRESCENDO HPCC design-in-context optimization problem in Section 3.4.2. This plot is the detailed version of Figure 3.10b. The dominant constraints have been labelled.

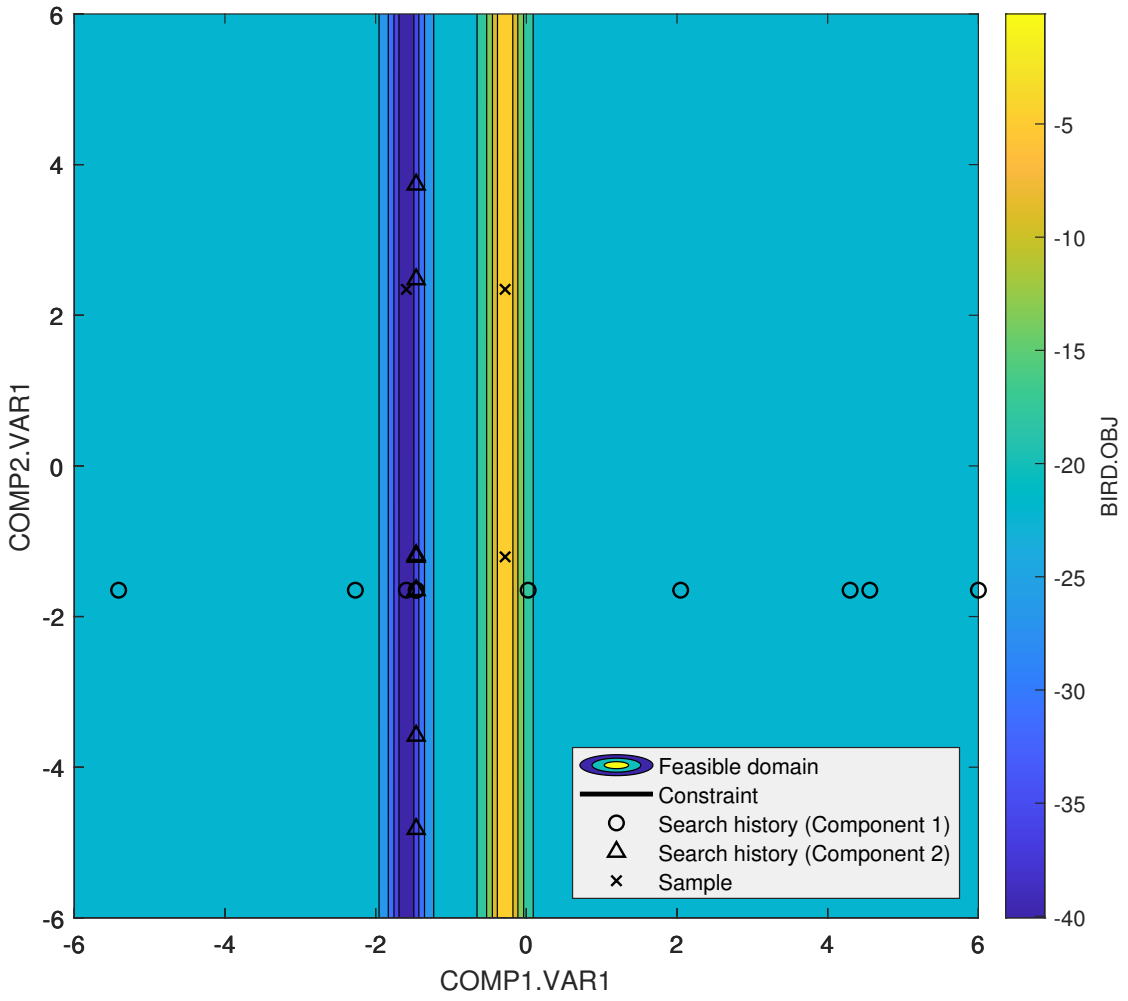


Figure C.10: Response surface of the Bird problem example in Section 4.3.1. This plot is the detailed version of Figure 4.3a.

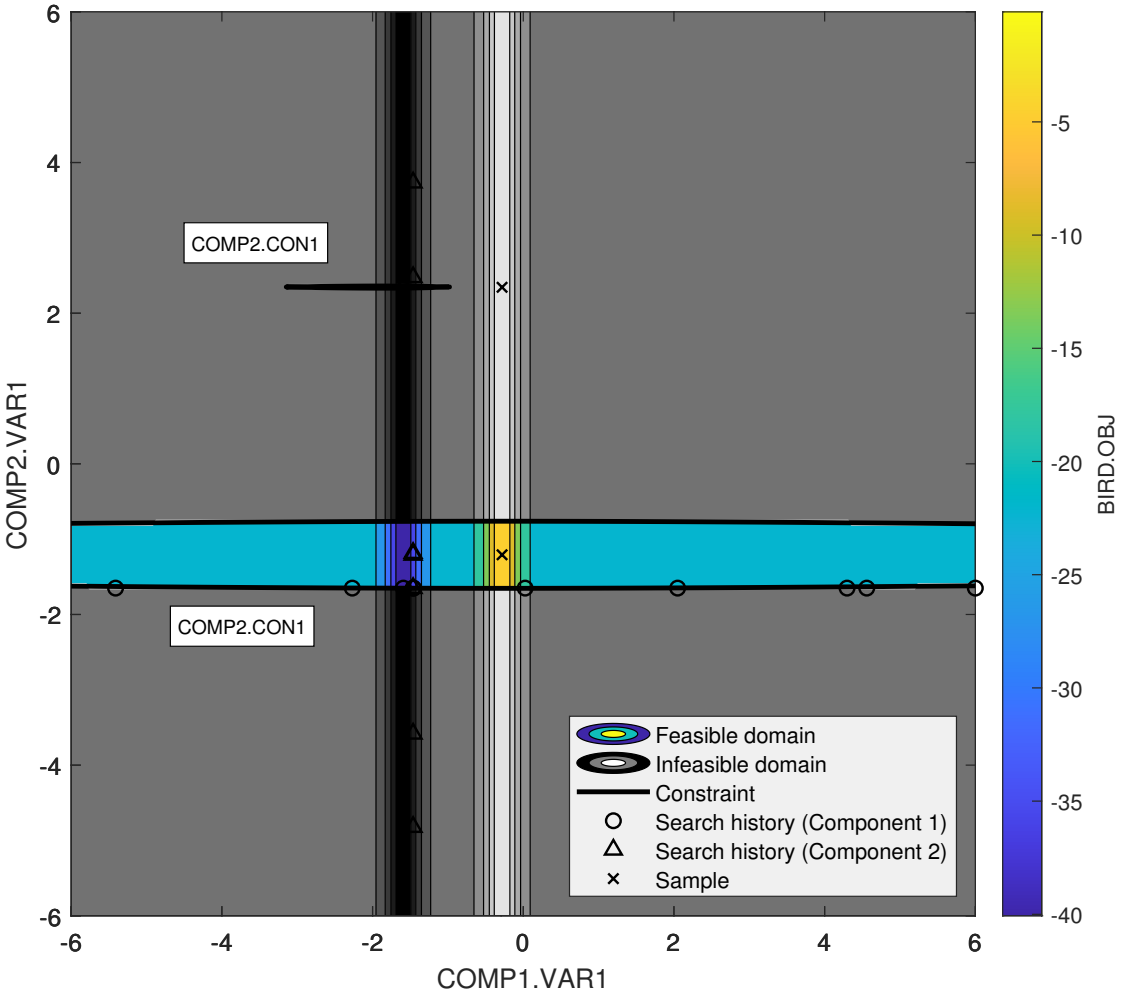


Figure C.11: Response surface of the Bird problem example in Section 4.3.1. This plot is the detailed version of Figure 4.3b. The dominant constraints have been labelled.

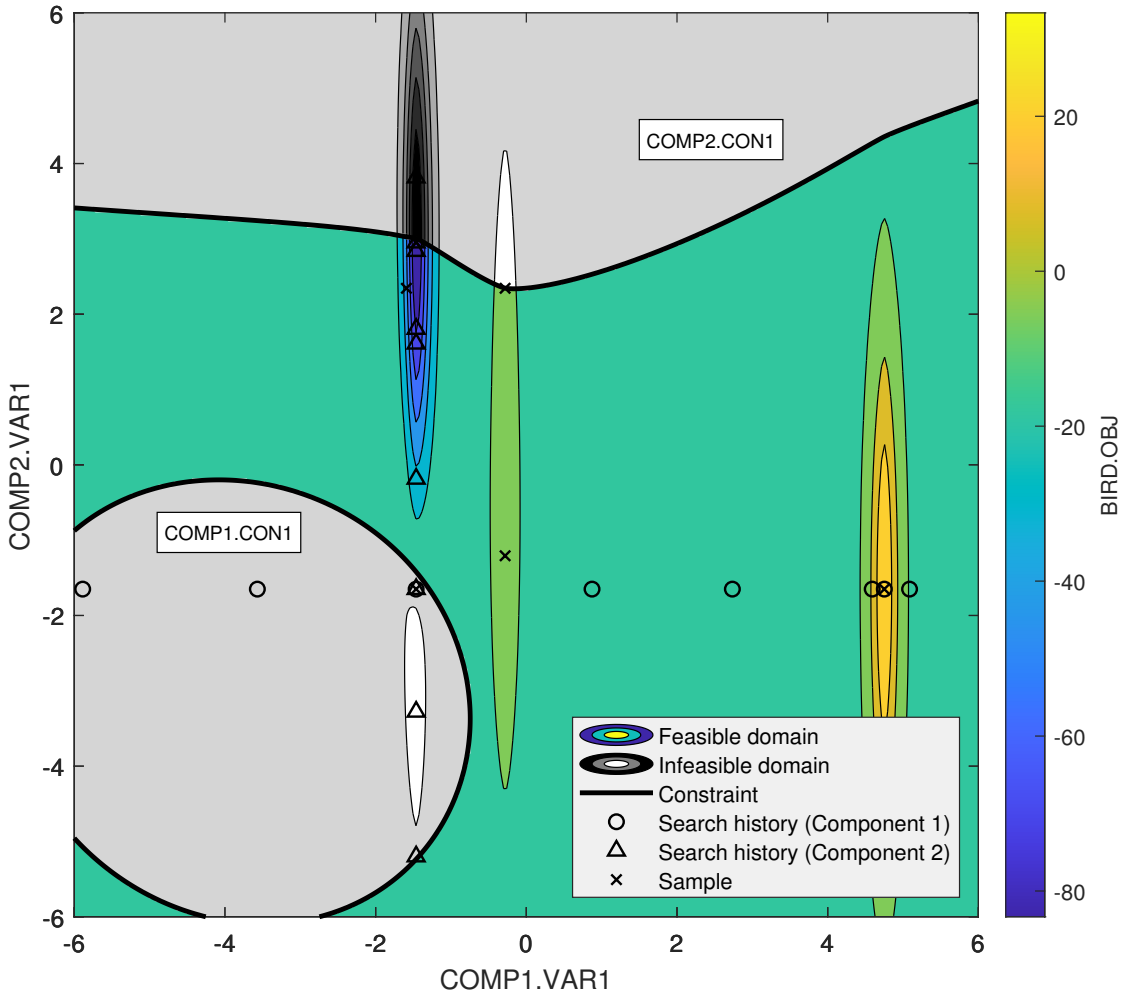


Figure C.12: Response surface of the Bird problem example in Section 4.3.1. This plot is the detailed version of Figure 4.4a. The dominant constraints have been labelled.

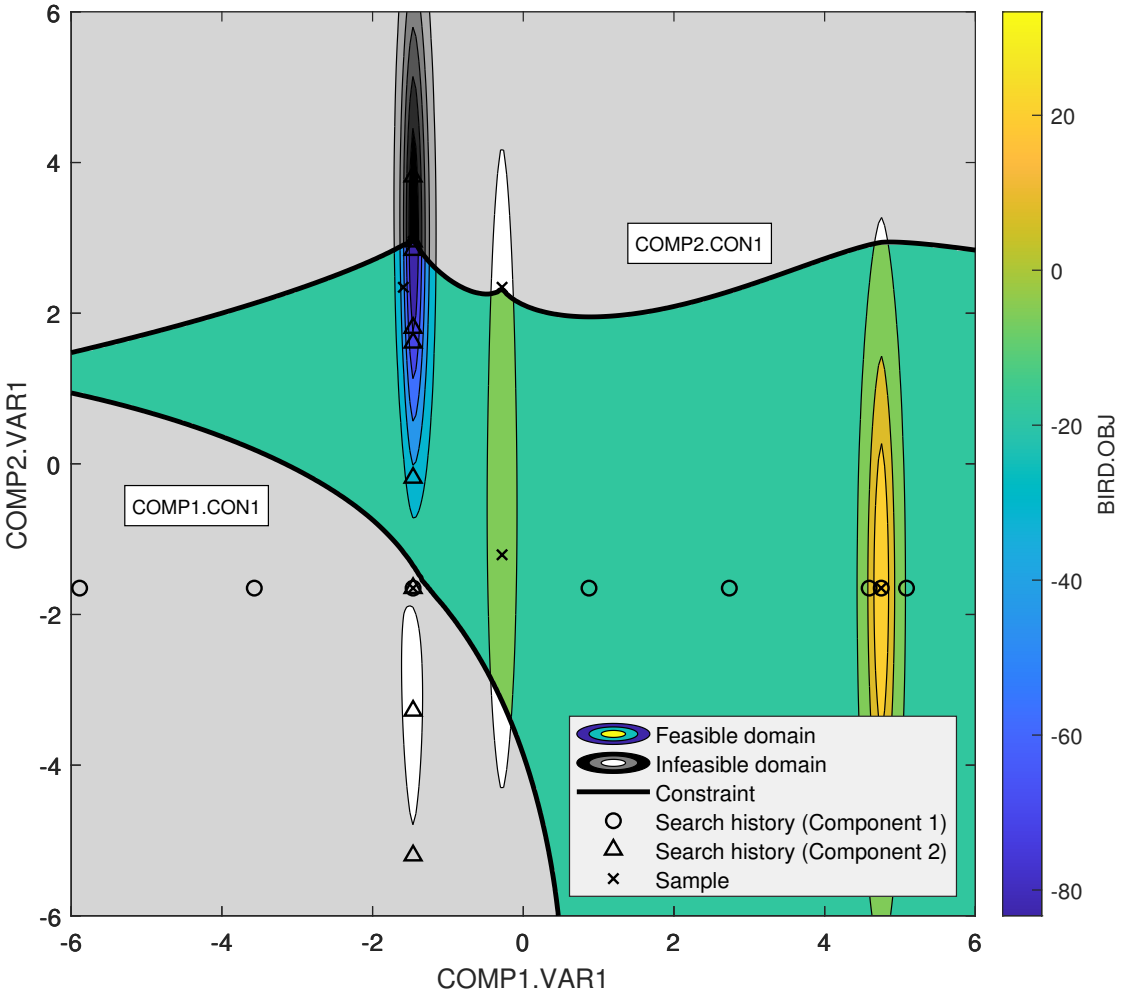


Figure C.13: Response surface of the Bird problem example in Section 4.3.1. This plot is the detailed version of Figure 4.4b. The dominant constraints have been labelled.

References

Allemang, R. J. (2002). The modal assurance criterion - 20 years of use and abuse. In *Proceedings of the 20th International Modal Analysis Conference*, Los Angeles, CA, USA.

Allison, J. T., Kokkolaras, M., and Papalambros, P. Y. (2006). On selecting single-level formulations for complex system design optimization. *Journal of Mechanical Design*, 129(9):898–906.

Arkhipov, A. N., Karaban, V. V., Putchkov, I. V., Filkorn, G., and Kieninger, A. (2009). The whole-engine model for clearance evaluation. In *Volume 6: Structures and Dynamics, Parts A and B*, pages 9–17, Orlando, Florida, USA. ASME.

Belaziz, M., Bouras, A., and Brun, J. M. (2000). Morphological analysis for product design. *Computer-Aided Design*, 32(5):377–388.

Bell, T. A., Jarrett, J. P., and Clarkson, P. J. (2008). Exploring the effects of removing process-intrinsic constraints on gas turbine design. *Journal of Propulsion and Power*, 24(4):751–762.

Berg, M. d., Cheong, O., Kreveld, M. v., and Overmars, M. (2008). *Computational geometry: Algorithms and applications*. Springer-Verlag, Berlin Heidelberg, 3 edition.

Bloebaum, C. L., Hajela, P., and Sobieszczanski-Sobieski, J. (1992). Non-hierarchic system decomposition in structural optimization. *Engineering Optimization*, 19(3):171–186.

Blum, H. (1967). A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge.

Blum, H. (1973). Biological shape and visual science (part I). *Journal of Theoretical Biology*, 38(2):205–287.

Boeller, S., Feuillard, B., Filkorn, G., Olmes, S., Prou, F., Robson, C., and Santos, R. (2018). An introduction into the clearance management of Ansaldo GT36 from development to validation. In *Proceedings of the ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition*, volume 3, Oslo, Norway. American Society of Mechanical Engineers.

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I. (2007). Multi-task Gaussian Process prediction. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 153–160, Red Hook, NY, USA. Curran Associates Inc.

Bordo, L., Bruzzone, S., Perrone, A., and Traversone, L. (2013). Prediction of clearance in industrial gas turbine validated by field operation data. In *Proceedings of the ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*, volume 7, pages 675–683, Copenhagen, Denmark. American Society of Mechanical Engineers.

Cao, W., Chen, X., and Gao, S. (2012). An approach to automated conversion from design feature model to analysis feature model. pages 655–665. American Society of Mechanical Engineers (ASME).

Chauhan, S. S., Hwang, J. T., and Martins, J. R. R. A. (2018). Benchmarking approaches for the multidisciplinary analysis of complex systems using a taylor series-based scalable problem. In Schumacher, A., Vietor, T., Fiebig, S., Bletzinger, K.-U., and Maute, K., editors, *Advances in structural and multidisciplinary optimization*, pages 98–116, Braunschweig, Germany. Springer International Publishing.

Chen, G. (2001). *FE model validation for structural dynamics*. PhD thesis, University of London, London.

Chong, C. S., Senthil Kumar, A., and Lee, K. H. (2004). Automatic solid decomposition and reduction for non-manifold geometric model generation. *Computer-Aided Design*, 36(13):1357–1369.

Dharmasaroja, A., Armstrong, C. G., Murphy, A., Robinson, T. T., McGuinness, S. H. M., Iorga, N. L., and Barron, J. R. (2017). Load case characterization for the aircraft structural design process. *AIAA Journal*, 55(8):2783–2792.

Donaghy, R. J., Armstrong, C. G., and Price, M. A. (2000). Dimensional reduction of surface models for analysis. *Engineering with Computers*, 16(1):24–35.

Forrester, A. I. J., Sóbester, A., and Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269.

Forrester, A. I. J., Sobester, A., and Keane, A. J. (2008). *Engineering design via surrogate modelling: A practical guide*. Wiley & Sons, Ltd.

Gürsoy, H. N. and Patrikalakis, N. M. (1991). Automated interrogation and adaptive subdivision of shape using medial axis transform. *Advances in Engineering Software and Workstations*, 13(5):287–302.

Heidari, M., Carlson, D., Sinha, S., Sadeghi, R., Heydari, C., Bayoumi, H., and Son, J. (2008). An efficient multi-disciplinary simulation of engine fan-blade out event

using md nastran. Schaumburg, IL, USA. American Institute of Aeronautics and Astronautics.

Hock, W. and Schittkowski, K. (1981). *Test examples for nonlinear programming codes.* Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin Heidelberg.

Huang, C.-H., Galuski, J., and Bloebaum, C. L. (2007). Multi-objective pareto concurrent subspace optimization for multidisciplinary design. *AIAA Journal*, 45(8):1894–1906.

Hulme, K. and Bloebaum, C. (2000). A simulation-based comparison of multidisciplinary design optimization solution strategies using CASCADE. *Structural and Multidisciplinary Optimization*, 19(1):17–35.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning.* Springer International Publishing.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Keane, A. J. and Nair, P. (2005). *Computational approaches for aerospace design: The pursuit of excellence.* Wiley & Sons, Ltd.

Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

Kimia, B. B., Tannenbaum, A. R., and Zucker, S. W. (1995). Shapes, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15(3):189–224.

Klein, C., Reitenbach, S., Schoenweitz, D., and Wolters, F. (2017). A fully coupled approach for the integration of 3D-CFD component simulation in overall engine performance analysis.

Klein, C., Wolters, F., Reitenbach, S., and Schönweitz, D. (2018). Integration of 3D-CFD component simulation into overall engine performance analysis for engine condition monitoring purposes.

Kulkarni, Y. H., Sahasrabudhe, A., and Kale, M. (2017). Leveraging feature generalization and decomposition to compute a well-connected midsurface. *Engineering with Computers*, 33(1):159–170.

Kumar, A., Wu, G., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., and Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 56.

Kumar, D., Juethner, K., and Fournier, Y. (2017). Efficient rotordynamic analysis using the superelement approach for an aircraft engine. page V07BT33A009.

Lee, S. H. (2005). A CAD–CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques. *Computer-Aided Design*, 37(9):941–955.

Leymarie, F. F. and Kimia, B. B. (2008). From the infinitely large to the infinitely small. In Siddiqi, K. and Pizer, S. M., editors, *Medial Representations: Mathematics, Algorithms and Applications*, Computational Imaging and Vision, pages 327–351. Springer Netherlands, Dordrecht.

Lockett, H. and Guenov, M. (2008). Similarity measures for mid-surface quality evaluation. *Computer-Aided Design*, 40(3):368–380.

Martins, J. R. R. A. and Lambe, A. B. (2013). Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075.

McCune, R. W., Armstrong, C. G., and Robinson, D. J. (2000). Mixed-dimensional coupling in finite element models. *International Journal for Numerical Methods in Engineering*, 49(6):725–750.

Minnicino, M. A., Ii, H., and David A. (2004). Overview of reduction methods and their implementation into finite-element local-to-global techniques:. Technical report, Defense Technical Information Center, Fort Belvoir, VA.

Mishra, S. (2006). Some new test functions for global optimization and performance of repulsive particle swarm method.

Monaghan, D. J. (2000). *Automatically coupling elements of dissimilar dimension in finite element analysis.* Ph.D., Queen's University of Belfast.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402.

Nolan, D. C., Tierney, C. M., Armstrong, C. G., Robinson, T. T., and Makem, J. E. (2014). Automatic dimensional reduction and meshing of stiffened thin-wall structures. *Engineering with Computers*, 30(4):689–701.

Parashar, S. and Bloebaum, C. (2005). Decision support tool for multidisciplinary design optimization (MDO) using multi-domain decomposition. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics.

Park, C., Haftka, R. T., and Kim, N. H. (2017). Remarks on multi-fidelity surrogates. *Structural and Multidisciplinary Optimization*, 55(3):1029–1050.

Pastor, M., Binda, M., and Harčarik, T. (2012). Modal assurance criterion. *Procedia Engineering*, 48:543–548.

Pollard, C. R., Bucklow, H., Butlin, G., and Johnson, B. (2017). Towards fully automatic mid-surface meshing. In *Proceedings of the 2017 NAFEMS World Congress*, page 36, Stockholm, Sweden. National Agency for Finite Element Methods and Standards (NAFEMS).

Price, M. A., Armstrong, C. G., and Sabin, M. A. (1995). Hexahedral mesh generation by medial surface subdivision: Part I. Solids with convex edges. *International Journal for Numerical Methods in Engineering*, 38(19):3335–3359.

Renaud, J. E. and Gabriele, G. A. (1993). Improved coordination in nonhierarchic system optimization. *AIAA Journal*, 31(12):2367–2373.

Rezayat, M. (1996). Midsurface abstraction from 3D solid models: general theory and applications. *Computer-Aided Design*, 28(11):905–915.

Robinson, T. T., Armstrong, C. G., and Fairey, R. (2011). Automated mixed dimensional modelling from 2D and 3D CAD models. *Finite Elements in Analysis and Design*, 47(2):151–165.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423.

Sóbester, A., Forrester, A. I. J., Toal, D. J. J., Tresidder, E., and Tucker, S. (2014). Engineering design applications of surrogate-assisted optimization techniques. *Optimization and Engineering*, 15(1):243–265.

Sellar, R. S., Batill, S. M., and Renaud, J. E. (1996). Response surface based, concurrent subspace optimization for multidisciplinary system design. Aerospace Sciences Meetings, Reno, NV, USA. American Institute of Aeronautics and Astronautics.

Sengoz, K., Kan, S., and Eskandarian, A. (2015). Development of a generic gas turbine engine fan blade-out full-fan rig model. Technical Report DOT/FAA/TC-14/43, George Washington University, Ashburn, VA, USA.

Shankar, J., Haftka, R. T., and Watson, L. T. (1993). Computational study of a nonhierarchical decomposition algorithm. *Computational Optimization and Applications*, 2(3):273–293.

Shim, K. W., Monaghan, D. J., and Armstrong, C. G. (2002). Mixed dimensional coupling in finite element stress analysis. *Engineering with Computers*, 18(3):241–252.

Sobester, A., Leary, S. J., and Keane, A. J. (2004). A parallel updating scheme for approximating and optimizing high fidelity computer simulations. *Structural and Multidisciplinary Optimization*, 27(5).

Sobieszczanski-Sobieski, J. (1990). Sensitivity of complex, internally coupled systems. *AIAA Journal*, 28(1):153–160.

Sracic, M. W. and Elke, W. J. (2019). Effect of boundary conditions on finite element submodeling. In Kerschen, G., editor, *Nonlinear Dynamics, Volume 1*, Conference Proceedings of the Society for Experimental Mechanics Series, pages 163–170, Cham. Springer International Publishing.

Stanley, F. (2010). *Dimensional reduction and design optimization of gas turbine engine casings for tip clearance studies*. PhD thesis, University of Southampton, Southampton.

Sun, L., Tierney, C. M., Armstrong, C. G., and Robinson, T. T. (2018a). Decomposing complex thin-walled CAD models for hexahedral-dominant meshing. *Computer-Aided Design*, 103:118–131.

Sun, L., Tierney, C. M., Armstrong, C. G., and Robinson, T. T. (2018b). An enhanced approach to automatic decomposition of thin-walled components for hexahedral-dominant meshing. *Engineering with Computers*, 34(3):431–447.

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., and Telea, A. (2016). 3D skeletons: A state-of-the-art report. *Computer Graphics Forum*, 35(2):573–597.

Taylor, R. M. (2006). The role of optimization in component structural design: application to the F-35 Joint Strike Fighter. In *Proceedings of the 25th International Congress of Aeronautical Sciences*, page 11, Hamburg, Germany. International Council of Aeronautical Sciences (ICAS).

Tedford, N. P. and Martins, J. R. R. A. (2010). Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11(1):159–183.

Thakur, A., Banerjee, A. G., and Gupta, S. K. (2009). A survey of CAD model simplification techniques for physics-based simulation applications. *Computer-Aided Design*, 41(2):65–80.

Toal, D. J. J. (2015). Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. *Structural and Multidisciplinary Optimization*, 51(6):1223–1245.

Toal, D. J. J., Bressloff, N. W., and Keane, A. J. (2008). Kriging hyperparameter tuning strategies. *AIAA Journal*, 46(5):1240–1252.

Toal, D. J. J., Bressloff, N. W., Keane, A. J., and Holden, C. M. E. (2011). The development of a hybridized particle swarm for kriging hyperparameter tuning. *Engineering Optimization*, 43(6):675–699.

Toal, D. J. J., Forrester, A. I. J., Bressloff, N. W., Keane, A. J., and Holden, C. (2009). An adjoint for likelihood maximization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2111):3267–3287.

Toal, D. J. J., Keane, A. J., Benito, D., Dixon, J. A., Yang, J., Price, M., Robinson, T., Remouchamps, A., and Kill, N. (2014). Multifidelity multidisciplinary whole-engine thermomechanical design optimization. *Journal of Propulsion and Power*, 30(6):1654–1666.

Tosserams, S., Etman, L. F. P., and Rooda, J. E. (2010). A micro-accelerometer MDO benchmark problem. *Structural and Multidisciplinary Optimization*, 41(2):255–275.

Towner, R. and Band, J. (2012). An analysis technique/automated tool for comparing and tracking analysis modes of different finite element models. In *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Honolulu, HI, USA. American Institute of Aeronautics and Astronautics (AIAA).

Wang, H., Jin, Y., and Doherty, J. (2018). A generic test suite for evolutionary multifidelity optimization. *IEEE Transactions on Evolutionary Computation*, 22(6):836–850.

Wang, L. (2020). Structural topology optimisation of a gas turbine engine and nacelle integration components employing a medial surface representation. In *Proceedings of the 2020 Aerospace Europe Conference*, Bordeaux, France. Association Aéronautique et Astronautique de France (3AF).

Wang, L., Toal, D. J. J., Keane, A. J., and Stanley, F. (2014). An accelerated medial object transformation for whole engine optimisation. In *Volume 2B: Turbomachinery*, page V02BT45A011, Düsseldorf, Germany. ASME.

Wang, L., Toal, D. J. J., Keane, A. J., and Stanley, F. (2017). A whole engine optimization based on medial object transformations. Manchester.

Woo, Y. (2014). Abstraction of mid-surfaces from solid models of thin-walled parts: A divide-and-conquer approach. *Computer-Aided Design*, 47:1–11.

Wujek, B. A., Renaud, J. E., Batill, S. M., and Brockman, J. B. (1996). Concurrent subspace optimization using design variable sharing in a distributed computing environment. *Concurrent Engineering*, 4(4):361–377.

Yi, S. I., Shin, J. K., and Park, G. J. (2008). Comparison of MDO methods with mathematical examples. *Structural and Multidisciplinary Optimization*, 35(5):391–402.