Global-Local Multidisciplinary Optimisation of Aircraft

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, November 2022.

Massimo Sferza

Supervised by

Walid Tizani
Kris van der Zee

and

Jelena Ninic (external)
Dimitrios Chronopoulos (external)

Signature Massimo Sferza

Date 30/11/2022

Abstract

In the aerospace industry, Multidisciplinary Design Optimisation (MDO) is increasingly being used in the early phase design of aircraft structures.

However, the computational complexity associated to this tool limits its application to coarse FE-models, which are not sufficiently accurate to capture the internal deformation of components, such as manholes, cut-outs, bulkheads, and stringer run-outs.

Because of this, aircraft designers are currently unable to evaluate the influence of these components on the sizing of primary structures. This causes the structural designs obtained through MDO to be suboptimal at best and possibly unfeasible, which limits the benefits and thwarts a broader application of this methodology.

In this thesis a novel methodology for the preliminary sizing of aircraft structures is introduced. The proposed global-local MDO procedure relies on the use of a coarse FE-model combined with multiple finer models for the accurate evaluation of components with a complex geometry. Thanks to the introduction of an ad-hoc sensitivity analysis, to consider the coupling between different models, the novel methodology ensures the optimality and feasibility of the computed design.

The impact on computational cost of adopting the proposed global-local strategy is limited, provided that the total number of constraints and design variables is not greatly increased.

Where the reference procedure would fail to find a locally feasible design, the proposed global-local approach successfully finds a multidisciplinary optimal design, which does not violate local constraints. Thus, the methodology enables designers to account for the effect of components with a complex geometry earlier in the design process and reduces the risk of major delays in the product development cycle.

Acknowledgements

I owe a debt to the many people I met and worked with during the *OptiMACS* project, without whom this work would not have been possible.

First and foremost, I'd like to thank my supervisor Jelena Ninic. She dedicated to me a great portion of her time in the past two and a half years and has been a constant source of encouragement during my research. I greatly appreciated her advice and valued her expertise in computational mechanics.

I am deeply thankful to Florian Glock, always nice and patient, he has been my reference point at Airbus throughout the project. He was always available to listen to my ideas, answer my questions and help in any way he could.

I am grateful to Christoph Hofer for the invaluable help he provided with the implementation of the procedure. I am also thankful for the care he took in analysing some of the most theoretical aspects during our discussions.

I would also like to thank all other supervisors, who advised me during different phases of my studies: Dimitrios Chronopoulos, who guided and supported me from the start, Kris van der Zee, for his support and his lessons on error estimation, Arthur Jones, who guided me during the first months and Walid Tizani, who took over during the last ones.

I am also thankful to many others in Airbus: Fernass Daoud, who welcomed me in the Stress Methods & Optimisation Team, Ögmundur Petersson, for his willingness in sharing his vast expertise and all other members of the team, who were always keen on helping me when needed.

I also thank, for both their help and the fun moments we shared, my fellow ESRs and collegues: Alessandro, Giorgos, Giuseppe, Neoklis and Weijie.

Lastly, I must thank my parents and my girlfriend Anna, for I could always rely on their love and support.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 764650.

Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

AAO All-At-Once.

ATC Analytical Target Cascading.

BLISS Bilevel Integrated System Synthesis.

CA Contributing Analysis.

CLT Classical Laminate Theory.

CO Collaborative Optimisation.

CSSO Concurrent Subspace Optimisation.

DFEM Detailed Finite Element Model.

DOF Degree Of Freedom.

ECO Extended Collaborative Optimisation.

FE Finite Element.

GFEM Global Finite Element Model.

GSE Global Sensitivity Equation.

IDF Individual Discipline Feasible.

MDA Multidisciplinary Analysis.

MDF Multidisciplinary Feasible.

MDO Multidisciplinary Design Optimisation.

PSO Particle Swarm Optimisation.

QSD Quasi-Separable Decomposition.

QSP Quasi-Separable Problems.

RF Reserve factor.

SAND Simultaneous Analysis and Design.

SBD Specified Boundary Displacements.

SBSF Specified Boundary Stiffness/Force.

List of Symbols

Static analysis

K	Stiffness matrix
p	Load vector
u	Displacement vector

Subblocks of the displacement vector

z	Global internal displacements
i	Global interface displacements
a	Local internal displacements
o	Local internal displacements

Subblocks of the stiffness matrix

K_{zz}	Subblock of the global stiffness matrix
K_{zi}	Subblock of the global stiffness matrix
K_{iz}	Subblock of the global stiffness matrix
K_{ii}	Subblock of the global stiffness matrix
K_{aa}	Subblock of the local stiffness matrix
K_{ao}	Subblock of the local stiffness matrix
K_{oa}	Subblock of the local stiffness matrix
K_{oo}	Subblock of the local stiffness matrix

Subblocks of the load vector

p_z	Subblock of the global load vector
p_i	Subblock of the global load vector
p_a	Subblock of the local load vector
p_o	Subblock of the local load vector

Static condensation

$K^{\dagger}_{aa}$	Condensed local stiffness matrix
$p^{\dagger}_{a}$	Condensed local load vector

Aeroelastic analysis

f^{A}_{rigid}	Rigid component of the aeroelastic load vector
C	Aeroelastic stiffness matrix

Optimisation

f	Objective function
g	Vector of constraints
$\boldsymbol{g}$	Inequality constraints
$\boldsymbol{h}$	Equality constraints
$\mathbf{x}$	Design variables
$\mathbf{x}_0$	Shared design variables
$\mathbf{x}_i$	Design variables influencing the i^{th} CA
$\overline{\mathbf{y}}$ ($\overline{\mathbf{y}_i}$)	Solution of the MDA (of the i^{th} CA)
R (R_i)	Residual of the MDA (of the i^{th} CA)
F (F_i)	Function defining the explicit form of the MDA (of the i^{th} CA)
$\mathbf{y}_i$	Coupling variable (output of the i^{th} CA)
$\hat{\mathbf{y}}_i$	Copy of $\mathbf{y}_i$
c	Consistency constraints
b_i	Budget
μ_i	Margin

Decorators

$\cdot^{G}$	Quantity related to the global model
$\cdot^{L}$	Quantity related to the local model

Computational cost

n_z	Number of z DOFs
n_i	Number of i DOFs
n_a	Number of a DOFs
n_o	Number of o DOFs
$\bar{n}_{\mathrm{IT}}^{\mathrm{aero}}$	Average number of iterations required to solve an aeroelastic analysis
n_{sc}^{stat}	Number of static subcases
n_{sc}^{aero}	Number of aeroelastic subcases
n_{sc}^{total}	Number of subcases
n^{LOCALS}	Number of local models
n_G	Number of global design variables
n_L	Number of local design variables

Chapter 1

Introduction

This chapter introduces the context of aircraft design, in which Multidisciplinary Design Optimisation (MDO) is employed as a tool to address the design paradox. The reliance of this procedure on coarse Finite Element (FE) models and the lack of an accurate description of some detailed parts, known as “non-regular areas”, is presented as an hindrance to the successful application of MDO. The challenge posed by “non-regular areas” and the limitations of the current strategies employed to contain the problem is described in detail.

1.1 The Design Process Paradox

1.1.1 Causes of the Paradox

The design of any complex product, not only in the aircraft industry, starts with a list of requirements or desired characteristics. At the beginning, the designers are free to make any design decision, but as soon as the process starts, each new decision is influenced by the previous ones. For example in aircraft design the wing position must be defined before its profile and aspect ratio can be designed.

Although similar products developed in the past may serve as a reference, at the beginning, the designers have a limited knowledge of how the new concept will match the target requirements, simply because a design, which has not yet been defined in detail, cannot be fully evaluated.

If during the design process the solution fails to satisfy the design requirements due to constraints imposed in previous steps, the designers must seek alternative solutions. This means retracing their steps and repeating part of the work, as represented in Figure

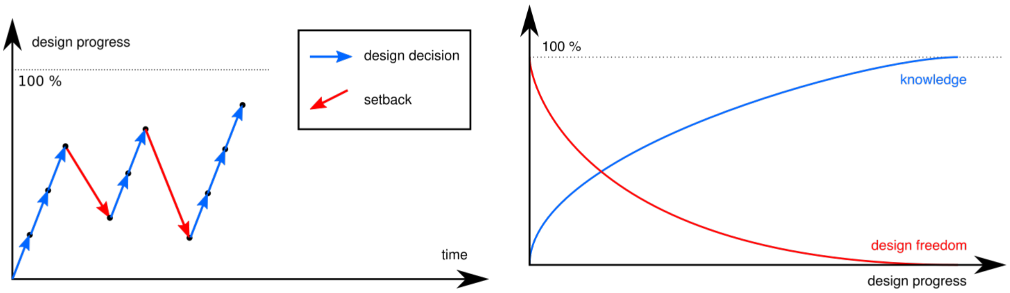
1.1a. In the earlier mentioned example, the designers would fail to find a satisfactory wing profile and aspect ratio and be forced to change the wing position.

Since looking for alternative solutions requires going back to a previous state and advancing once again from there, setbacks result in additional work and time delays and consequently an increased design cost. While some degree of iteration is inevitable, major setbacks discovered in the late development stages must be avoided at all costs [1, 2, 3]

As design advances, the increasing cost of going back progressively limits the design freedom. If problems are discovered late, it may be preferable to stick with a sub-optimal solution, because of the committed money cost.

1.1.2 The Paradox

This leads to a problem known as the *design process paradox* [4]: at the beginning, when the design freedom is maximal, little information is available to guide the decision-making, while instead towards the end, as more information is acquired, the design is set and the initial freedom is lost. The paradox is depicted in Figure 1.1b.



(a) Product development is characterised by steps forward and setbacks due to the violation of design requirements.

(b) The design paradox: as designers gain knowledge on how to design the product, they lose the freedom to modify the design.

Figure 1.1: Setbacks are normal in product development, but their opportunity cost increases as the design progresses, which leads to the design paradox.

1.1.3 Aircraft Industry Strategies to Address the Paradox

Two obvious consequences of the paradox are that knowledge acquired earlier is more valuable and that decisions, taken at the beginning of the design process, are of critical importance. Because of these, the aircraft industry relies on two main strategies to address the problem:

- Working in parallel on multiple models during early design,
- Using a multidisciplinary approach to design.

In the earliest phase of aircraft design [5], when the committed costs are small, multiple models are studied in parallel. This leads to an increased knowledge, which is vital to avoid major setbacks. Multiple possibilities are explored, and the least promising designs are quickly discarded without an impact on cost. This is the first strategy commonly adopted by the industry.

As each decision influences all decisions downstream, the first design choices are critical. Multidisciplinary Design Optimisation (MDO) is a powerful and well-established approach for designing an aircraft while concurrently considering several requirements. By means of MDO, it is possible to maximise the performance of an aircraft configuration. Bringing forward the use of MDO to the initial design phases and taking advantage of this tool to make better design decisions, while advancing the design of each concept in parallel, is the second strategy adopted to prevent major setbacks.

1.2 Use of MDO for Aircraft Design

1.2.1 Phases of the design process

The design of a new aircraft usually starts with a list of requirements, either directly requested by the customer or identified as a customer need by the aircraft manufacturer. The process that follows is divided into three different phases [6]. The first one is conceptual design. Many configurations, called conceptual designs, are generated, each with its strengths and weaknesses. During this phase, the different variants are further developed, with the aim of identifying the most promising one. The trade-off between different performance measures, such as weight, range, fuel efficiency or payload, is evaluated, and the least promising designs are identified and discarded. As the set has been reduced to one or, at most, two similar variants, the preliminary design begins. During this phase, the design is further optimised and analysed, and more details are defined. At the end of the preliminary design phase, the FE-model still lacks many details, but most of the aspects defining the design are fixed. For example, the position of cut-outs or the number and spacing of ribs and stringers has been determined. The mass estimation is more precise, and the general loadpaths of the structure have been determined. The chosen configuration then enters the detailed design phase. Each component is defined in detail. For those components that are subject to complex stress states, Detailed Finite Element Models (DFEMs) are created and sized to sustain the predetermined loads. At last, the design can be certified, and the manufacturing process starts.

1.2.2 Application of MDO in the Design Process

MDO is not applied during the early stages of conceptual design. This is because, at the beginning, not even a CAD model is available, and time is needed for the preparation of the FE-models and the definition of design variables and constraints required by MDO. Because of this extra time, other approaches are preferred to MDO during the early stages of conceptual design, as they can be used to discard several configurations more quickly.

There are two scenarios in which global structural MDO can be used. The first one is in the transition from conceptual to preliminary design. As soon as the number of configurations is sufficiently reduced, MDO can be applied to improve the performance of each variant. At this stage, proper FE-models are not available, so a CAD model is automatically generated from a set of parameters and meshed. The aim during this phase is not only to optimise each design, but also to assess its performance. In particular, two aspects are considered, with the ultimate goal of comparing different designs and selecting the most promising one: whether the configuration can meet the requirements and how sensitive the performance is to a change in design. It is therefore possible to maximise the performance and to identify so-called “show-stoppers” as early as possible.

MDO can also be used in the context of preliminary design. At this stage proper FE-models become available from other departments and the concept is defined, but there is still enough freedom to modify the main structural elements. An example would be the material or shape of the stringers. Furthermore, the location of cut-outs has been approximately defined, but the engineers still have the freedom, for example, to move a manhole to the next rib-bay. Figure 1.2 illustrates the stages where MDO is used.

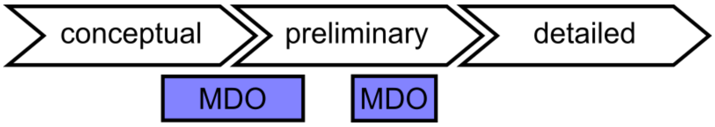


Figure 1.2: The design phases and the use of MDO.

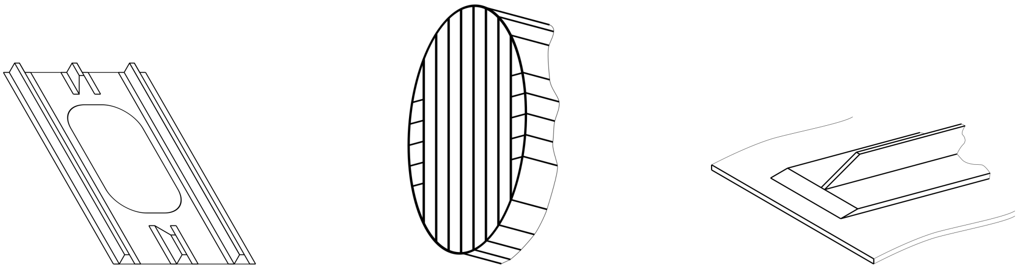
1.2.3 MDO Relies on Coarse Models

MDO, as described above, is a tool used to globally optimise the design of an entire aircraft or major components of its structure as a whole wing. In this context, the FE-models used are necessarily simple and neglect many details of the structure. Using fine models would increase the computational cost of a procedure which is already demanding [7], as it must consider multiple disciplines, loadcases and constraints, therefore coarse FE-model are used instead [8]. This goes well together with the fact that most details have not been defined yet and that the primary goal at this stage is to determine the loadpaths and the internal stresses of the structure.

1.3 Non-Regular Areas in Aircraft Design

1.3.1 Presence of Non-Regular Areas

Some components, which because of their geometry would undergo a complex deformation, are excluded from the optimisation. The level of detail required to capture the irregularities of their displacement field would result in additional modelling effort and a prohibitive computational cost of the MDO procedure. These components will be called "non-regular areas" within this work. Examples of these are cut-outs, bulkheads, stringer run-outs and points of local load introduction, such as engine pylons (Figure 1.3).



(a) Cut-out (b) Bulkhead (c) Stringer run-out

Figure 1.3: Examples of non-regular areas.

1.3.2 Location of Non-Regular Areas

It is already known that some areas will host some particular features, such as a manhole or a stringer run-out. Nevertheless, the exact position of non-regular areas in the model depends on the design phase in which MDO is being applied. When using MDO in the late stages of conceptual design, the FE-model will be created starting from a CAD model.

As the position of main structural components is still being defined, the location of non-regular areas is determined using heuristic techniques, relying on engineering experience. If instead MDO is being used in the context of preliminary design, an FE-model is already available.

In this case, the position of non-regular areas is practically prescribed in the sense that only minor adjustments are allowed.

1.3.3 FE-Modelling of Non-Regular Areas

Since local features of the structure have not been designed when MDO is used, in order to account for their mechanical properties, non-regular areas are modelled by modifying the stiffness of one or more bidimensional elements. In particular, using approximation formulae that modify the Young's modulus of the element, the in-plane and the bending stiffnesses are estimated. It is sometimes possible to capture the stiffness of the feature with sufficient precision. In other cases, this procedure results in an under- or over-estimation of the stiffness. In a similar way, the weight estimation is also compromised by the lack of a detailed model.

1.3.4 Handling of Non-Regular Areas in the Optimisation

Non-regular areas are not updated during the optimisation. If they were, one would need to update the computed equivalent stiffness properties. Instead, they are kept fixed while the MDO procedures operate on the elements around them, for which design variables are defined.

Furthermore, these areas are not constrained since the displacement field computed by the analyses, despite being capable of capturing the overall deformation of the structure, is not sufficiently accurate to describe the internal deformation of the non-regular areas. It is also not possible to capture the effect of the interaction between "non-regular areas" and the global model on the load carrying behaviour of the entire structure. A classical example in which this interaction is particularly strong is the case of a sequence of lower wing skin panels, placed spanwise one next to the other, each with a manhole or an inspection hole. In this case each hole creates an opportunity for the load paths to split, which leads to non-linear changes in the structural behaviour, which cannot be captured, if non-regular areas are ignored during the global optimisation. Ultimately, it is not possible to prevent a constraint violation within a non-regular area. One can only constrain the surrounding elements, which inject and extract the load from the part and design their thicknesses to divert part of the load away.

1.4 Consequences of neglecting non-regular areas

The approach, presented in the previous section, is limited by the lack of detailed models of non-regular areas, such as cut-outs, bulkheads, stringer run-outs and engine pylons. This causes the following issues: (i) an inaccurate weight estimation; (ii) an inaccurate stiffness estimation; (iii) the inability to apply optimisation constraints to the non-regular areas; and iv) the inability to optimise these complex parts. These consequences may result in turn in four major problems: (a) wrong assessment of the configuration performance, (b) inaccurate representation of the global optimisation problem, (c) undetected local constraint violations, (d) sub-optimal overall design.

1.4.1 Wrong Assessment of the Configuration Performance

The inaccurate weight estimation of the local part is itself a problem. It influences the weight of the final design and in turn many measures of performance. Therefore, it affects the comparison with other configurations during conceptual design, as summarised in Figure 1.4. Since the non-regular areas often represent heavily reinforced parts, the effect of these errors is not negligible.

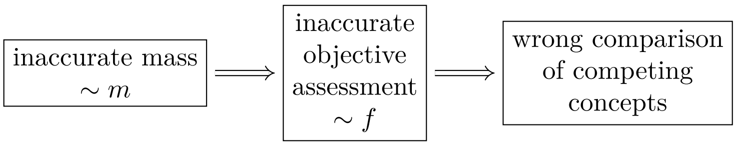
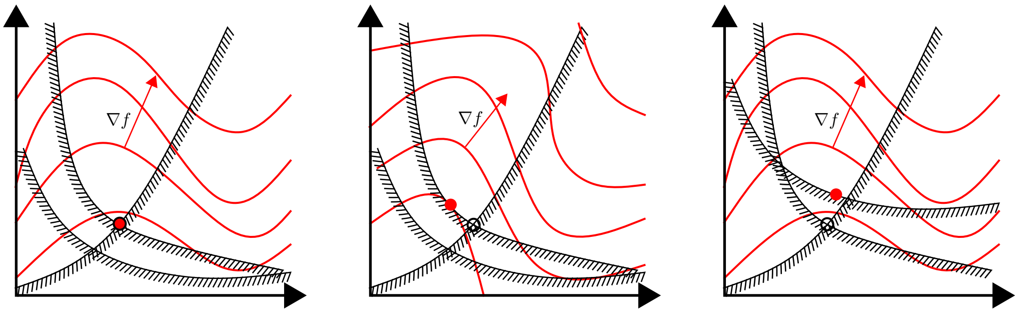

inaccurate mass
$\sim m$
inaccurate objective assessment
$\sim f$
wrong comparison of competing concepts

Figure 1.4: Wrong assessment of the configuration performance.

1.4.2 Inaccurate Representation of the Global Optimisation Problem

The second problem is that the result of the MDO procedure is, possibly, a sub-optimal or invalid design because the optimisation problem is not accurately defined. In other words, the design is feasible and optimal given the available information (Figure 1.5a), but as soon as the information on the non-regular areas is more accurately gauged, the design is either infeasible (Figure 1.5b) or too conservative (Figure 1.5c).



(a) Optimisation problem solved by the MDO procedure

(b) Suboptimal solution: when accurately represented, the objective is different

(c) Invalid solution: when accurately represented, the constraints are different

Figure 1.5: Consequences of an inaccurate representation of the optimisation problem.

Inaccurate stiffness estimation of the non-regular areas alter the analyses solutions and cause constraint violation to be over- or under-estimated. If, in particular, the stiffness was overestimated, a greater portion of the load will be funnelled through the non-regular area. As a consequence, the structures acting in series along the main loading direction, injecting and extracting load from the part, will be subject to more stress. Instead, structures acting in parallel will be subject to less stress. Therefore, the MDO procedure will design structures in series to be thicker than they need to be and structures in parallel to be thinner. An analogous effect with opposite results would be the consequence of an underestimation of the stiffness. In this case, the phenomenon is relevant, even if the non-regular areas are not particularly heavy.

The same might happen due to wrong weight estimations since these also alter the analyses solutions. Inaccurate weight estimations result in wrong inertia loads. Wrong inertia loads affect the analyses solutions. As a result, if the constraint violations are over-estimated, some parts of the structure are sized thicker than they need to be; thus, the final design will be heavier (sub-optimal design). If instead the constraint violation is under-estimated, the structure will be thinner than it needs to be and possibly unable to sustain the real loads (invalid design). This effect is non-negligible if the non-regular areas are relatively heavy compared to the whole structure.

The effect of both, the inaccurate estimation of stiffness and the inaccurate estimation of mass, have been summarised in Figure 1.6.

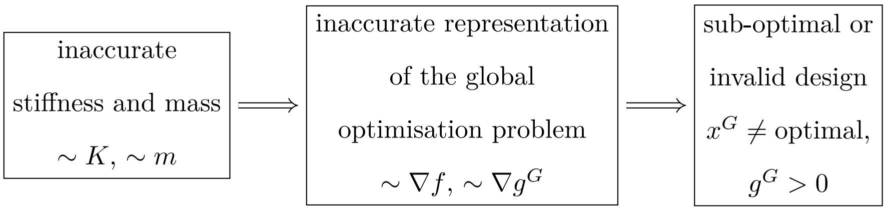


Figure 1.6: Inaccurate representation of the global optimisation problem.

1.4.3 Undetected Local Constraint Violations

The third problem is that the design could violate constraints within the non-regular areas, as summarised in Figure 1.7. The MDO will be able to converge to a feasible solution, which satisfies all the constraints considered. Nevertheless, additional requirements for the displacement field over non-regular areas exist, although they could not be applied due to the lack of a sufficiently detailed model. If the unconstrained local areas become infeasible, the design cannot be accepted. The risk of this happening can be reduced by controlling the geometry and the constraints of the structures surrounding the non-regular area, but it is impossible to prevent it. This problem is demonstrated with an example presented in Section 8.1.

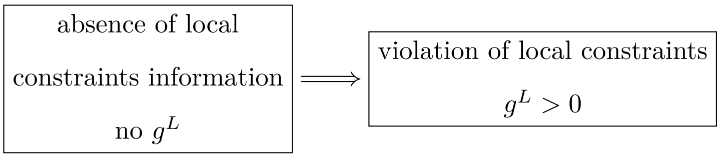


Figure 1.7: Undetected local constraint violations.

1.4.4 Sub-Optimal Overall Design

Lastly, optimising the structure without changing the design of the non-regular areas and subsequently optimising these in a subsequent step may prevent the optimiser from reaching a better optimum, as part of the design variables are alternatively fixed. This has been summarised in Figure 1.8.

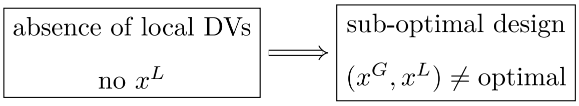


Figure 1.8: Sub-optimal overall design.

1.5 Summary

From the early phases of aircraft design, practitioners are confronted with the design paradox. In the beginning it is hard to take advantage of the available design freedom, because the ultimate effect of design choices cannot be fully understood. As the development goes on and it becomes easier to make informed decisions, part of the design choices have been made and the design freedom has been reduced. MDO has proved to be an effective tool in addressing the paradox, but the associated computational complexity has prevented its use with accurate FE-models. The use of inaccurate models for the treatment of parts with a complex geometry, commonly referred to as “non-regular areas”, can cause delays in the product development cycle, despite some of the strategies put forth by the industry. The need to address this problem is the motivation of this research project.

Chapter 2

Aims and research objectives

This chapter clarifies the motivation of this work and identifies the research objectives. Additionally, section 2.3 presents the structure of the thesis.

2.1 Motivation

The optimization of aircraft structures is a challenging task: many disciplines have to be taken into account, several constraints must be respected and multiple loadcases must be considered. The problem is complicated even further by the great design freedom provided by the use of composites.

As explained in Chapter 1, MDO is a tool successfully employed by aircraft designers to tackle this problem, but there is a tension between the already high computational cost and the need for additional accuracy to ensure the feasibility of “non-regular areas” and precisely evaluate the performance of a design.

To satisfactorily resolve this conflict, local information must be made available during the optimisation, while containing the computational cost. Despite the fact that the design of “non-regular areas” is strongly coupled with the remaining part of the structure, some room for compromise exists, as some features are inherently local.

For example, design variables can be divided in local or global, based on whether they affect the design of “non-regular areas” or the remaining structure. Similarly strength and buckling constraints can also be divided in local and global. Although several types of disciplines must be considered during the optimisation, static analysis is deemed sufficient for the evaluation of local constraints. Lastly, not all loadcases and disciplines considered at global level require the evaluation of local constraints.

2.2 Research objective

The aim of this research project is to develop a multidisciplinary optimisation procedure for the design of composite aircraft structures, capable of evaluating strength and buckling constraints over "non-regular areas" at an acceptable computational cost.

The take advantage of the locality of part of the problem, the existing multidisciplinary optimisation procedure is combined with a global-local analysis strategy. By analysing "non-regular areas" separately, the efficient solution of the global problem is coupled with the ability to capture the mechanical behaviour of local parts only when needed.

To accomplish this goal, the following research objectives have been identified:

- Review the literature on the available MDO architectures, their applications to the optimisation of aircraft structures, in conjunction with models with different levels of accuracy, and identify the limitations of current procedures.
- Review the literature on coupling techniques and global-local analysis strategies and select a suitable strategy to be applied within a MDO procedure.
- Develop a novel global-local MDO methodology to extend the existing MDO procedure.
- Implement the procedure within *Lagrange*, an Multidisciplinary Design Optimisation software used and developed by Airbus Defense and Space [9], to extend and compare against the currently employed optimisation methodology.
- Investigate and identify the main drivers of the computational cost of the procedure.
- Verify the correct implementation of the procedure through a series of tests and evaluate the modelling accuracy.
- Demonstrate the limitations of the design strategy, currently employed in dealing with non-regular areas.
- Demonstrate the capacity of the global-local MDO procedure to overcome this limitations.
- Show the applicability of the procedure to industrial case studies.

2.3 Structure of the thesis

The work conducted within this research project is described in 9 chapters. A brief description of each chapter is presented next.

Chapter 1 describes the general context of Multidisciplinary Design Optimisation of aircraft, the inaccurate representation of local “non-regular areas”and the consequent problems. Despite the mitigation strategies employed by practitioners, “non-regular areas” can result in an inaccurate comparison of multiple conceptual designs or lead to an infeasible or suboptimal design. Hence in this chapter the need for an alternative way to treat “non-regular areas” is motivated.

Chapter 2 outlines the motivation and the research objectives for this work.

Chapter 3 presents the literature review on MDO architectures and their application as well as the literature review on global-local analysis techniques. Despite the variety of global-local analysis methods, these are generally ignored when MDO is applied and a decomposition of the optimisation problem and a distributed architecture are used instead. For the specific problem of “non-regular areas” a monolithic architecture used with a global-local analysis technique is proposed to alleviate this problem.

Chapter 4 presents the novel methodology that has been developed for global-local MDO. The structure is partitioned in a global and one or more local models. The analyses are solved by condensing the local models, solving the global one by using the condensed local information and solving the local models by using the global solution as a boundary condition. An ad-hoc sensitivity analysis formulation is derived, based on the global-local coupling defined in the analysis step. Lastly, the problem is solved, adopting a monolithic architecture and executing the global-local procedure within each optimisation iteration.

Chapter 5 describes in detail the implementation of the proposed global-local MDO within *Lagrange*, a software for Multidisciplinary Design Optimisation, used and developed by Airbus Defense and Space. The procedure has been successfully implemented working mostly through the *Lagrange*-Python interface. Similar results could be obtained combining another software for optimisation or MDO and a FE-solver.

Chapter 6 analyses the computational cost of the procedure. The procedure is more computationally efficient than a mesh refinement of the reference model and the overall impact on computational cost is limited, provided that the total number of constraints and design variables is not greatly increased.

Chapter 7 verifies its correct implementation. The correct computation of analyses solutions and sensitivities is verified by comparing the solution of global-local problems and the solution computed by *Lagrange* of their single-model equivalents. The methodology delivers results in good agreement with those obtained by *Lagrange*, with minor deviations mostly attributable to round-off errors.

Chapter 8 presents a series of case studies, which illustrate the shortcomings of the current strategies in dealing with “non-regular areas”, proves the successful application of the new methodology introduced and proves its applicability to industry size problems. At last, Chapter 9 contains the conclusions of this research work, including the limitations of the proposed methodology and future research directions.

2.4 Publications

The following articles and papers where produced, while working on this thesis:

- *Multidisciplinary Optimisation of Aircraft Structures with Critical Non-Regular Areas: Current Practice and Challenges*, Aerospace, 2021, Massimo Sferza, Jelena Ninic, Dimitrios Chronopoulos, Florian Glock, Fernass Daoud;
- *Global-local multidisciplinary optimisation of Aircraft* UKACM Conference, 2021, Massimo Sferza, Jelena Ninic, Dimitrios Chronopoulos, Florian Glock, Fernass Daoud, awarded *Best Post-graduate Research Student Award UKACM 2021*
- *Recent progress, challenges and outlook for multidisciplinary structural optimization of aircraft and aerial vehicles* Progress in Aerospace Science, 2022, Giuseppe Corrado, Georgios Ntourmas, Massimo Sferza, Neoklis Traiforos, Albertino Arteiro, Louise Brown, Dimitrios Chronopoulos, Fernass Daoud, Florian Glock, Jelena Ninic, Ender Ozcan, Jose Reinoso, Gerd Schuhmacher, Thomas Turner;
- *Global-local multidisciplinary optimisation with ad-hoc sensitivity analysis for the preliminary design of aircraft*, Engineering with Computers, 2022, Massimo Sferza, Jelena Ninic, Florian Glock, Christoph Hofer, Fernass Daoud, Dimitrios Chronopoulos, Kristoffer Van Der Zee.

2.5 Summary

The development of a novel global-local methodology is identified as the strategy to address the problem of “non-regular areas” in the context of MDO. The ultimate goal is to improve the reliability of MDO as a design tool, by bringing knowledge of “non-regular areas” to the early phases of design, where changes are cheap and design decisions are crucial.

The following part of the thesis comprises a thorough literature review, the theoretical formulation of the methodology, its implementation, considerations on the associated computational cost, verification and proof of its applicability to industrial size problems.

Chapter 3

Literature Review

In chapter 1 Multidisciplinary Design Optimisation (MDO) was identified as suitable strategy to address the design paradox. Since the design of an aircraft requires taking into account several disciplines and the interaction of several subsystems, the idea to use MDO to address the paradox can already be found found in [10] by Sobieszczanski-Sobieski et al. This chapter contains a general introduction to MDO and a review the literature on the available MDO architectures. Their applications to the optimisation of aircraft structures and the possibility to use them for the analysis and design of "non-regular areas" is reviewed. Furthermore, coupling techniques and global-local analysis strategies commonly employed outside the context of MDO have been reviewed.

3.1 General Aspects of Multidisciplinary Design Optimisation

3.1.1 MDO Optimisation Problem

A MDO problem is a classical optimisation problem of the form:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimise}} \quad & f(\mathbf{x}, \overline{\mathbf{y}}) \\ \text{subject to} \quad & \boldsymbol{g}(\mathbf{x}, \overline{\mathbf{y}}) \leq 0 \\ & \boldsymbol{h}(\mathbf{x}, \overline{\mathbf{y}}) = 0 \end{aligned} \tag{3.1}$$

where f is the objective function, $\boldsymbol{g}$ and $\boldsymbol{h}$ are the inequality and equality constraints, respectively, $\mathbf{x}$ are the design variables and $\overline{\mathbf{y}}$ is the solution of the Multidisciplinary

Analysis (MDA):

$$\begin{array}{cc} \text{explicit form} & \text{residual form} \\ \overline{\mathbf{y}} = \mathrm{F}(\mathbf{x}) & \mathrm{R}((\mathbf{x}), \overline{\mathbf{y}}) = 0 \end{array} \tag{3.2}$$

The solution of the optimisation problem is bound to the solution of the MDA, since f, $\boldsymbol{g}$, $\boldsymbol{h}$ and their derivatives depend on $\overline{\mathbf{y}}$, as represented in Figure 3.1.

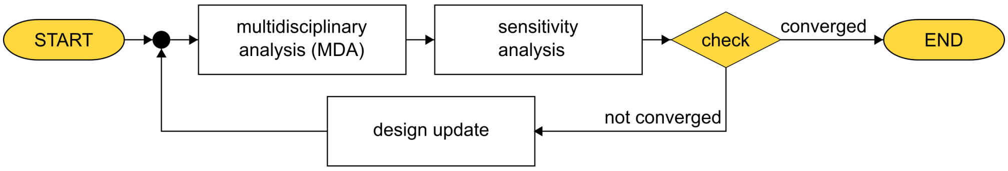


Figure 3.1: Structure of an MDO problem.

3.1.2 Internal Structure of the Multidisciplinary Analysis

The MDA naturally breaks down in multiple Contributing Analyses (CAs), each providing part of the solution $\overline{\mathbf{y}} = [\overline{\mathbf{y}_1}, \ldots, \overline{\mathbf{y}_i}, \ldots, \overline{\mathbf{y}_N}]$.

Let the solution variables of the i^{th} contributing analysis, $\overline{\mathbf{y}_i}$, be called *state variables*. These depend on the solution variables of other CAs, $\overline{\mathbf{y}_j}$, as well as on the design variables $\mathbf{x}$. The i^{th} contributing analysis can be formulated as:

$$\begin{array}{cc} \text{explicit form} & \text{residual form} \\ \overline{\mathbf{y}_i} = \mathrm{F}_i(\mathbf{x}, \overline{\mathbf{y}_j}) & \mathrm{R}_i((\mathbf{x}, \overline{\mathbf{y}_j}), \overline{\mathbf{y}_i}) = 0 \\ \forall j \in \mathcal{C}_i & \forall j \in \mathcal{C}_i \end{array} \tag{3.3}$$

where $\mathcal{C}_i$ indicates the set of CAs coupled with the i^{th} CA, thus $j \in \mathcal{C}_i \implies j \neq i$. The i^{th} CA can therefore be modelled as a system, as in Figure 3.2, receiving $\mathbf{x}$, $\overline{\mathbf{y}_j}$ as an input and generating $\overline{\mathbf{y}_i}$ as an output.

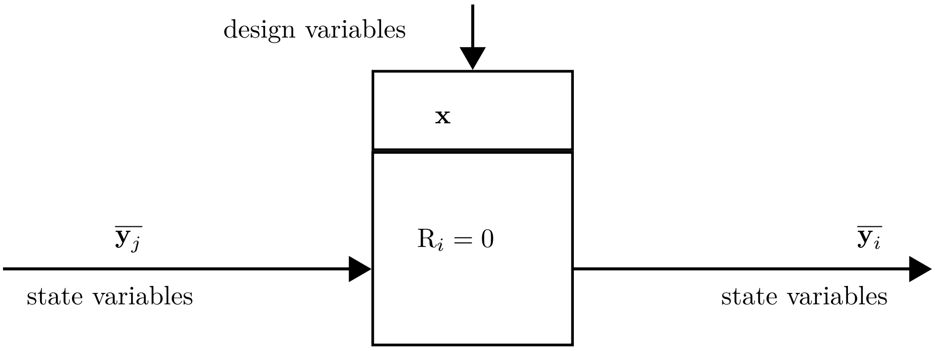


Figure 3.2: A contributing analysis (i) modelled as a system, receiving input from another CA (j).

The MDA is a complex but structured problem, which can be seen as the combination of smaller sub-problems, coupled as in Figure 3.3.

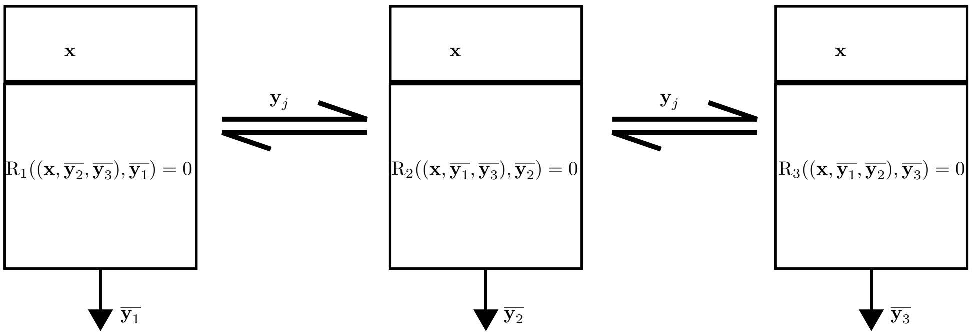

x
$R_1((\mathbf{x}, \overline{\mathbf{y}_2}, \overline{\mathbf{y}_3}), \overline{\mathbf{y}_1}) = 0$
$\overline{\mathbf{y}_1}$
$\mathbf{y}_j$
x
$R_2((\mathbf{x}, \overline{\mathbf{y}_1}, \overline{\mathbf{y}_3}), \overline{\mathbf{y}_2}) = 0$
$\overline{\mathbf{y}_2}$
$\mathbf{y}_j$
x
$R_3((\mathbf{x}, \overline{\mathbf{y}_1}, \overline{\mathbf{y}_2}), \overline{\mathbf{y}_3}) = 0$
$\overline{\mathbf{y}_3}$

Figure 3.3: An MDA with 3 interacting CAs.

3.1.3 Difference between State and Coupling Variables

As a matter of fact, each CA does not depend on the *state variables* of other CAs directly. It depends on them through a smaller subset of variables, called *coupling variables* and denoted with $\mathbf{y}_j$. The i^{th} CA can be modelled more precisely as in Figure 3.4, and its formulation can be written as:

$$\begin{array}{cc} \text{explicit form} & \text{residual form} \\ \overline{\mathbf{y}_i} = F_i(\mathbf{x}, \mathbf{y}_j) & R_i((\mathbf{x}, \mathbf{y}_j), \overline{\mathbf{y}_i}) = 0 \\ \forall j \in \mathcal{C}_i & \forall j \in \mathcal{C}_i \end{array} \tag{3.4}$$

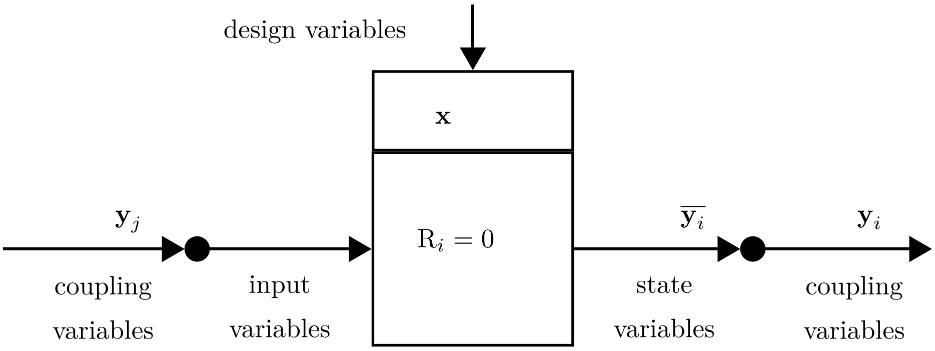


Figure 3.4: A contributing analysis (i) modelled in detail, receiving input from another CA (j).

3.1.4 Contributing Analyses Result from Disciplines or Substructures

Since the coupling variables are fewer than the state variables, the CAs are coupled together but also partially independent. It is because of this that representing the MDA as a structured problem makes sense. In other words, the decomposition of the MDA in multiple CAs is not arbitrary but such that, for each CA, the number of coupling variables is much smaller than the number of state variables.

This often happens along interdisciplinary lines and physical boundaries, so it is common to consider different disciplines and substructures as part of separate contributing analyses.

3.1.5 MDO Architecture

The key to efficiently solve an MDO problem is to exploit the internal structure of the problem and decompose it appropriately. Depending on the internal couplings and their strength, different ways of nesting optimisation and analysis solutions might prove more effective in reducing computational cost. In the following, the structure of the algorithm applied to solve the MDO problem is called MDO architecture, as in [11].

3.2 Classification of MDO Architectures

Following the example of Cramer et al. in [12, 13], MDO architectures can be classified as Multidisciplinary Feasible (MDF), Individual Discipline Feasible (IDF) and All-At-Once (AAO). All of these are based on the same optimisation procedure, depicted in Figure 3.1, but the difference lies in a different treatment of the analysis block, stemming from radically different definitions of the underlying optimisation problems.

Another aspect in the classification of MDO architectures is the distinction between monolithic and distributed, which is introduced at the end of this subsection and further described in Section 3.3.

3.2.1 Multidisciplinary Feasible

MDF architectures represent the traditional approach and are the most intuitive to understand. As represented in Figure 3.5, an MDF architecture is essentially an optimisation procedure, which solves an MDA at each iteration.

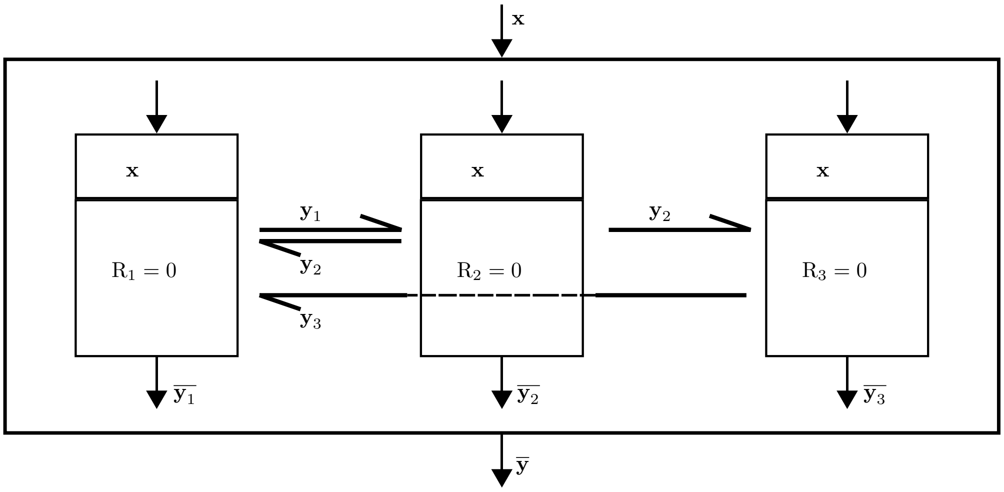


Figure 3.5: Flowchart of multidisciplinary feasible architecture: detail of the analysis block from Figure 3.1.

It solves the MDO problem exactly as formulated in Eq. 3.1.

3.2.2 Individual Discipline Feasible

Another possibility is to break the coupling between disciplines, as represented in Figure 3.6. Instead of directly providing each analysis i with the required coupling variable $\mathbf{y}_j$, computed as the response of the analysis j, a value $\hat{\mathbf{y}}_j$ is provided by the optimiser instead. In this way, each analysis can be solved independently, and the optimiser becomes responsible for enforcing the feasibility among disciplines through the consistency constraint $c = \mathbf{y}_j - \hat{\mathbf{y}}_j = 0$.

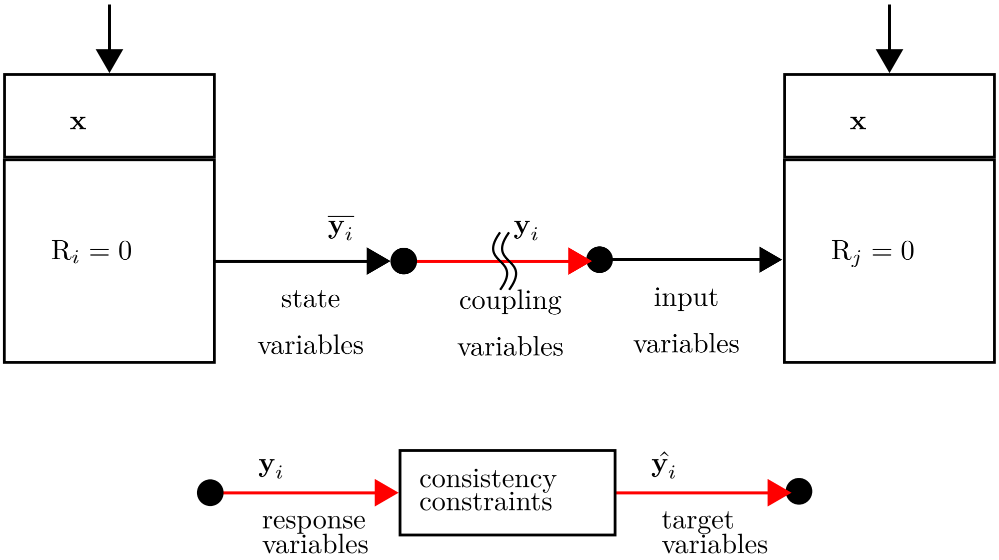


Figure 3.6: A coupling variable is substituted by a response–target and consistency constraint.

This strategy is known as IDF. As in MDF, each CA satisfies the condition for individual discipline feasibility:

$$\mathrm{R}_i((\mathbf{x}_0, \mathbf{x}_i, \hat{\mathbf{y}}_j), \mathbf{y}_i) = 0 \tag{3.5}$$

but, in contrast to MDF, it does not satisfy the condition for interdisciplinary feasibility:

$$\mathbf{y}_j - \hat{\mathbf{y}}_j = 0 \tag{3.6}$$

In other words, in each optimisation iteration, each discipline is solved with the given input, but it is not guaranteed that the input of each discipline matches the output of the coupled CAs, as represented in Figure 3.7. The optimisation problem can be written as:

$$\begin{aligned} \underset{\mathbf{x},\hat{\mathbf{y}}}{\text{minimise}} \quad & f(\mathbf{x}, \overline{\mathbf{y}}(\mathbf{x}, \hat{\mathbf{y}})) \\ \text{subject to} \quad & \boldsymbol{g}(\mathbf{x}, \overline{\mathbf{y}}(\mathbf{x}, \hat{\mathbf{y}})) \leq 0 \\ & \boldsymbol{h}(\mathbf{x}, \overline{\mathbf{y}}(\mathbf{x}, \hat{\mathbf{y}})) = 0 \\ & c = \mathbf{y}(\mathbf{x}, \hat{\mathbf{y}}) - \hat{\mathbf{y}} = 0 \end{aligned}$$

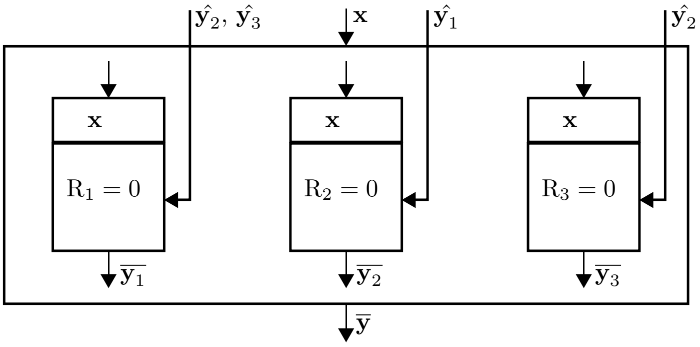


Figure 3.7: Flowchart of the individual discipline feasibility architecture: detail of the analysis block from Figure 3.1.

3.2.3 All-at-Once

The third option is AAO. In this case, the optimiser directly operates on the state variables $\overline{\mathbf{y}}$ and the copies of the coupling variables $\hat{\mathbf{y}}_i$. It is not only responsible for coupling the CAs but also for satisfying each residual equation, as represented in Figure 3.8. Therefore, during the optimisation, the design may be infeasible with respect to the single discipline, as well as with respect to the interdisciplinary coupling. Only at convergence is the design guaranteed to respect both: individual and interdisciplinary feasibility.

$$\begin{aligned}
\underset{\mathbf{x},\overline{\mathbf{y}},\hat{\mathbf{y}}}{\text{minimise}} \quad & f(\mathbf{x},\overline{\mathbf{y}}) \\
\text{subject to} \quad & \boldsymbol{g}(\mathbf{x},\overline{\mathbf{y}}) \leq 0 \\
& \boldsymbol{h}(\mathbf{x},\overline{\mathbf{y}}) = 0 \\
& \mathrm{R}((\mathbf{x}),\overline{\mathbf{y}}) = 0 \\
& c = \mathbf{y}(\mathbf{x},\hat{\mathbf{y}}) - \hat{\mathbf{y}} = 0
\end{aligned}$$

The first example of AAO was Simultaneous Analysis and Design (SAND), proposed by Haftka in [14]. Instead of iteratively solving the analysis within each optimisation iteration, SAND treats the response variables as design variables and adds an equality constraint to ensure individual discipline feasibility.

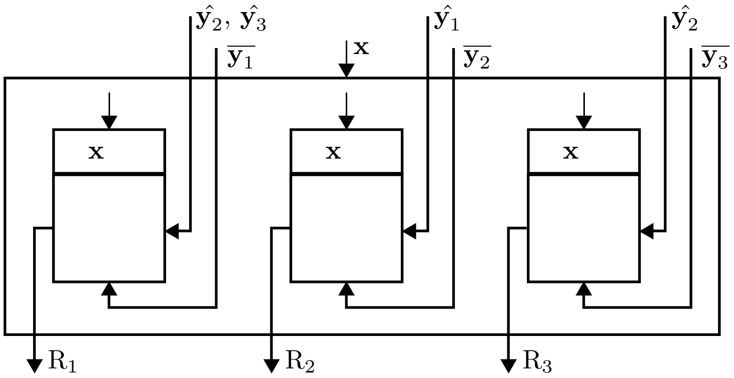


Figure 3.8: Flowchart of the all-at-once architecture: details of the analysis block from Figure 3.1.

As an example, let the underlying analysis be a linear static analysis $Ku = p$, where K is the stiffness matrix, p is the load vector and u is the displacement field. Instead of solving the problem:

$$\begin{aligned}
\underset{\mathbf{x}}{\text{minimise}} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & \boldsymbol{g}(\mathbf{x},u) \leq 0
\end{aligned} \tag{3.7}$$

by repeatedly solving $Ku = p$ for each $\mathbf{x}$, SAND would reformulate the optimisation problem as:

$$\begin{aligned}
\underset{\mathbf{x},u}{\text{minimise}} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & \boldsymbol{g}(\mathbf{x},u) \leq 0 \\
& R(\mathbf{x},u) = Ku - p = 0
\end{aligned} \tag{3.8}$$

The same idea can be applied if the underlying analysis is an MDA.

3.2.4 Monolithic and Distributed Architectures

Thus far, the entire procedure was assumed to be based on one unique optimisation loop. When this is the case, the architecture is called monolithic. The alternative is a procedure that includes multiple optimisation procedures. When this is the case, the architecture is called distributed. Figure 3.9 illustrates the difference between the two. In order to use a distributed architecture, the MDO problem must be decomposed in multiple optimisation problems. Since there are several ways of decomposing the problem, the topic of distributed architectures is rather complex. The next subsection is therefore entirely dedicated to distributed architectures.

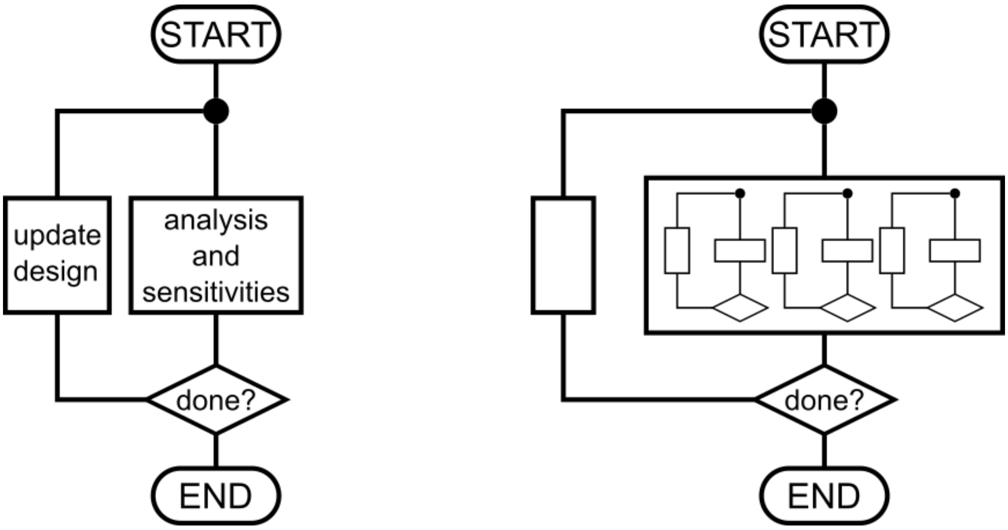


Figure 3.9: Comparison between a monolithic (**left**) and a distributed architecture (**right**).

3.3 Distributed Architectures

When the optimisation problem is decomposed into multiple sub-problems, a new challenge emerges: the coordination of the sub-problems. Many architectures are based on a two-level decomposition of the MDO problem: an upper system-level and a lower one with multiple subsystems. The coordination can then be achieved by nesting the lower level optimisations within the upper level optimisation.

3.3.1 Multilevel Optimisation by Linear Decomposition and CSSO

In [15], Sobieski et al. described one of the earliest examples of this approach. A multilevel optimisation procedure based on a hierarchical linear decomposition was presented in [10, 16]. In 1988, Sobieszczanski-Sobieski introduced Concurrent Subspace Optimisation (CSSO) [17, 18], a distributed architecture decomposing the problem along interdisciplinary lines, allowing to account for non-hierarchical relations between CAs. In CSSO, the design variables are partitioned and assigned to the CAs that are most influenced by them; for each CA, all constraints are combined into a single cumulative constraint. Each CA is individually optimised using a first order approximation of the coupled disciplines. This allows the work in each discipline to be done in parallel. Coordination between different CAs is controlled by parameters assigned by a coordination optimisation problem.

3.3.2 Global Sensitivity Equations

All these approaches rely on the formulation of the Global Sensitivity Equations (GSEs), for the computation of $\frac{dy}{dx}$. Alternative formulations and approaches for the solution of GSEs were presented in [19, 20]. The solution methods are not easily implemented but present computational advantages when compared with finite differencing, as discussed, for example, in [21].

3.3.3 BLISS

A more recent architecture is Bilevel Integrated System Synthesis (BLISS) [22]. Similar to CSSO, BLISS is a distributed architecture, which decomposes the problem in two levels and solves each discipline in parallel at the subsystem level. The coordination is ensured by dividing the design variables in shared variables and local variables: shared variables are modified in the system coordination problem, while instead local variables are exclusively modified by the subsystem optimisation problems.

3.3.4 Collaborative Optimisation and Its Extension

In 1995, Braun and Kroo introduced a decomposition known as Collaborative Optimisation (CO) [23, 24]. This is another example of a distributed approach, with a system level optimisation problem and subsystem level optimisation problems defined by breaking the interdisciplinary couplings. In this sense, the approach is similar to CSSO, but in the case of CO, the sub-spaces receive copies of the shared design variables, which are fixed by the system level optimiser. More recently, the approach was improved by providing each subsystem with information on the constraints of other subsystems. The new approach, named Extended Collaborative Optimisation (ECO) and presented in [25, 26], follows the idea of keeping the system level problem small, which is common to the original approach (CO).

3.3.5 Quasi-Separable Decomposition

Most distributed approaches are based on decompositions, which lack a mathematical justification, and may therefore not converge to the optimum. Haftka and Watson [27] identified a class of optimisation problems, Quasi-Separable Problems (QSP), for which it is possible to define a decomposition, supported by a rigorous mathematical theory and with proven convergence properties. The *quasi-separable subsystem problem* is defined as:

$$\begin{aligned} \underset{\mathbf{x}_0,\mathbf{x}_i}{\text{minimise}} \quad & f_0(\mathbf{x}_0) + \sum_{i=1}^{N} f_i(\mathbf{x}_0, \mathbf{x}_i) \\ \text{subject to} \quad & g^{(0)}(\mathbf{x}_0) \leq 0 \\ & g^{(i)}(\mathbf{x}_0, \mathbf{x}_i) \leq 0 \qquad\qquad i = 1, \ldots, N \end{aligned} \tag{3.9}$$

The associated decomposition is:

$$\begin{aligned} \underset{\mathbf{x}_0,b_i}{\text{minimise}} \quad & f_0(\mathbf{x}_0) + \sum_{i=1}^{N} b_i \\ \text{subject to} \quad & g^{(0)}(\mathbf{x}_0) \leq 0 \\ & \mu_i(\mathbf{x}_0, b_i) \leq 0 \quad\; i = 1, \ldots, N \end{aligned} \tag{3.10}$$

$$\begin{aligned} \underset{\mathbf{x}_i}{\text{minimise}} \quad & \mu_i \\ \text{subject to} \quad & g^{(i)}(\mathbf{x}_0, \mathbf{x}_i) - \mu_i \leq 0 \\ & f_i(\mathbf{x}_0, \mathbf{x}_i) - b_i - \mu_i \leq 0 \quad i = 1, \ldots, N \end{aligned} \tag{3.11}$$

The idea is to provide a budget b_i for f_i, which is a maximum allowable value for f_i. In this way, the upper level problem does not directly require the computation of f_i and is therefore independent from $\mathbf{x}_i$. The upper level is solved for the optimal $\mathbf{x}_0$ and b_i, where the constraints $g^{(i)}$ are substituted by a constraint on the margin $-\mu_i$. The lower level problem operates on the local design variables $\mathbf{x}_i$ to maximise the margin $-\mu_i$, ensuring that the value of $f_i(\mathbf{x}_0, \mathbf{x}_i)$ is contained within the allowed b_i. The decomposition does not introduce spurious solutions, local solutions of the decomposed problem correspond to local solutions of original problem under convexity assumptions, and finding the global optimum of the decomposed problem is equivalent to finding the global optimum of the original quasi-separable problem.

An extension of the architecture to include mixed-integer problems at the lower level and the corresponding conditions can be found in [28].

3.3.6 Analytical Target Cascading

A more general architecture known as Analytical Target Cascading (ATC) was developed by Kim et al. and presented in [29, 30]. ATC is a distributed multilevel hierarchical decomposition for MDO. The decomposition hierarchy follows a tree structure, in which the master node at the top level is unique, and each node can have children. Each node is coupled with its unique parent and its children but not with siblings. The coupling variables are split into targets and responses. Each node represents an optimisation problem, which matches the target assigned by the parent problem with its own response, i.e., the result of the analysis associated with the node. In this sense, the targets are cascaded from parent to child, hence the name of the method. The responses of the children are treated as design variables by the parent node. To ensure the coupling of the targets with the children true responses, consistency constraints are formulated in the optimisation problem, for which a budget is included in the objective. In [31], Michelena et al. showed that under convexity conditions, ATC converges to the optimum of the original problem. The architecture has been successfully applied to industrial problems [32, 33].

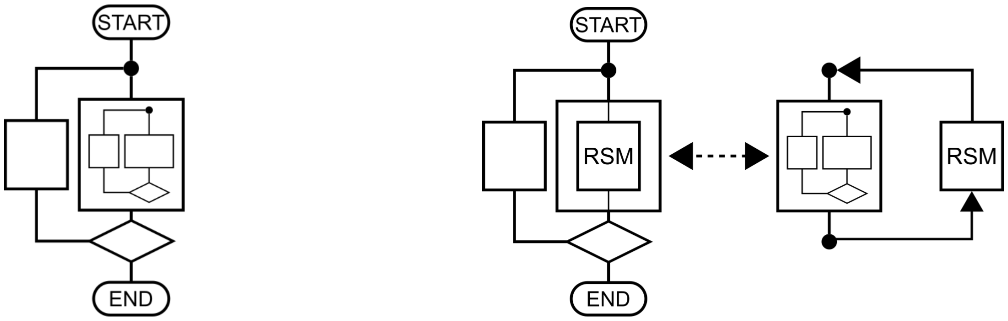
3.3.7 Augmented Lagrangian Decomposition

In [34], Tosserams et al. used augmented Lagrangian relaxation to improve the efficiency of ATC. In [35], they showed how the method can be applied to the solution of quasi-separable problems. Then in [36], they presented a decomposition for a more general class of problems, with its corresponding solution strategy being a generalisation of both ATC and the architecture proposed by Haftka and Watson [27].

3.3.8 Use of Response Surface Methods with Distributed Architectures

Many of the methods presented rely on isolated lower-level procedures, which are called repeatedly within each iteration of an upper-level procedure. To take a further advantage from the decomposition and completely decouple the upper- and lower-level procedures, it is possible to interpose a response surface, as illustrated in Figure 3.10. The lower level procedure is used to build and update the response surface, which is interrogated by the upper level procedure. This idea has been implemented in combination with

CSSO [37], BLISS [38], CO [39] and Quasi-Separable Decomposition (QSD) [40].



(a) Two-level architecture, with a nested optimisation loop.

(b) A response surface (RSM) can be substituted for the nested procedure. The RSM can be created and updated in parallel, effectively decoupling the two optimisation procedures.

Figure 3.10: A response surface can be used to decouple a two-level architecture.

3.3.9 Choice of Architecture

In [41], Vanaret et al. have shown that the choice of the architecture is always influenced by the problem under consideration. The choice of the architecture critically affects the performance of the optimisation, but it is impossible to tell a priori which architecture will perform better.

Since it is impractical to identify a promising architecture by trial and error, various software frameworks have been developed to combine optimisers and discipline specific software for the solution of MDO problems. Some of these can be used to easily benchmark and compare different architectures. An early example is pyMDO [42, 43], developed in Python by Tedford and Martins and used to compare many well-known architectures [44]. They found that monolithic architectures were better than distributed. Other examples are openMDAO [45, 46], developed by NASA, and GEMS [47, 48], developed at IRT Saint Exupery.

In [41], Vanaret et al. presented a methodology to compare architectures by replacing each CA with a scalable analytic replacement function. Furthermore, they have shown that the performance of an MDO architecture depends on the number of coupling and design variables.

3.4 Applications of MDO architectures to the problem of airframe sizing

Distributed architectures are commonly adopted for the design of aircraft and aircraft wings in particular. The usual approach is to decompose the problem into two levels: the global wing optimisation and the local sizing of selected panels.

3.4.1 Alternate Execution of Global and Local Optimisation

In [49], Ciampa et al. described an approach for the preliminary design of aircraft wings. They applied a global–local decomposition to minimise the mass of the structure. At a global level, they modelled the entire wing with spars and ribs and adopted a smeared stiffness approach for the stringers. At the local level, they modelled isolated panels in detail for the evaluation of stress and buckling constraints. To couple the two levels, they applied global stress and displacement fields to the local panel, while the optimised local design was used to update the global properties. In order to integrate the two optimisation strategies, they alternately performed global and local optimisation.

3.4.2 Nested Execution of Global and Local Optimisation

Instead of alternating between global and local optimisation, most authors propose to treat the local optimisation as a nested procedure.
In [50], Noevere and Wilhite describe a global–local approach for the weight minimisation of a wingbox. The approach is limited to linear statics and considers strength, buckling and maximum displacement constraints. At the global level, the weight of the upper and lower skin is minimised by modifying the parameters of a stiffness distribution function. In each iteration of the global optimisation, a nested local optimisation procedure is selected for each panel, which receives the boundary loads and the assigned stiffness as an input and provides weight and feasibility as output. At the local level, the weight of isolated panels was minimised under strength and buckling constraints. For each panel modelled with stringers, an equivalent panel with smeared stiffness was computed by applying Classical Laminate Theory (CLT). Following this approach, they were able to take as design variables the entries of the ABD matrix. Local constraints were evaluated through response surfaces, which provide the constraint compliance/violation as a function of the stiffness entries. Thanks to the use of stiffness variables and a response surface, the local optimisation resulted in a linear programming problem, solved

using the simplex algorithm.

Other examples of a global–local approach, in which the local optimisation is implemented as a nested procedure, can be found in the work of Kapania and his group [51, 52, 53, 54, 55]. Their work is related to the design and optimisation of metallic wings with curvilinear spars and ribs. At the global level, the wing internal layout of spars and ribs is optimised. At the local level, single panels are sized, as described in [56]. The global procedure consists of two steps. In the first one, Particle Swarm Optimisation (PSO) is used to optimise the number of spars and ribs. In the second one, gradient based optimisation is used to optimise the shape variables. The local panel optimisation, used to size the element thicknesses and optimise the shape of the stiffeners, is called a nested procedure, which can be used for each design to be evaluated. The approach considers multiple loadcases and multiple disciplines at the wing level, while instead, only statics are used at the panel level to evaluate stress and buckling constraints when enforcing the global displacements as boundary conditions. The approach was extended from weight minimisation to multi-objective optimisation [57], studying the compromise between weight and flutter speed.

In [58], Zhao and Kapania proposed a bi-level nested approach not based on a wing-panel decomposition. The top level operates on the configuration of SpaRibs to satisfy flutter constraints, using PSO, as described in [54]. The lower level minimises the wing root bending moment by operating on the control surfaces rotations.

Stanford et al. in [59] and Stanford in [60] propose a bi-level nested strategy for the optimisation of a wingbox. The upper level defines the topology and layout of the wingbox, using a surrogate based optimisation strategy. The article compares a case with straight stiffeners and ribs against one with curvilinear reinforcements. In the first case, the upper optimisation modifies number, rotation and spacing of the stiffeners and number and spacing of the ribs. In the second case, the optimiser additionally operates on some shape parameters. The lower level sizes spars, ribs, skins and stiffeners using a gradient based optimisation.

3.4.3 Parallel Execution through Response Surfaces

Instead of nesting the local optimisation within the global one, some authors propose the use of a response surface model with the results of the local panel optimisation, effectively decoupling the two optimisations, which can then be run in parallel.

An example of this strategy is presented in [61] by Liu et al. The authors considered the entire wing in the upper level and isolated panels in the lower one. At wing level, they minimised the weight under strength and buckling constraints by modifying the thicknesses of the various plies. At panel-level, after rounding the number of plies to an integer value, they maximised the buckling resistance for the given in-plane loads by optimising the stacking-sequence with GA.

In [62], Elham et al. propose a bilevel optimisation strategy, in which the optimisation problem is decomposed into top-level and multiple sub-level optimisations. The top-level minimises a combination of weight and drag by modifying the planform geometry of the aircraft. The two levels are coupled via consistency constraints in the top-level. Therefore, the top-level defines targets for the sub-level optimisations by operating on the values of drag coefficients, area of equivalent panel, lift curve slopes and airfoil pitching moments. The sub-level optimisations modify the airfoil shapes to minimise the distance to the top-level target values. This decomposition strategy is applied in combination with response surfaces.

In [63, 64], Ragon et al. described another weight minimisation procedure based on a wing-panel decomposition. At the global level, the wing was sized on a coarse model. At the local level, a detailed model of the panel was used for a precise sizing. The panel was subject to global in-plane loads and was designed to match the stiffness requirements resulting from the global optimisation. As in [61], the authors suggested the use of a response surface to avoid nesting the local optimisation in the global one.

3.4.4 Comparison of the Three Approaches

It may not be possible to apply the method of alternating between a global and a local optimisation, as this requires decomposing the original problem into two independent optimisation problems. This strategy effectively reduces the size of the optimisation problem and avoids the computational cost associated with a nested optimisation procedure, but may not converge to the optimal solution or may show a reduced speed of convergence, as it alternatively fixes part of the design variables. Furthermore, there is not a clear criterion to decide when to switch over to the other optimisation. Conversely, the strategy based on a nested local optimisation is applicable to a larger class of problems and may lead to a better solution but requires performing the local optimisation until convergence in each iteration of the global optimisation. As it has been shown previously (Figure 3.10), a response surface can be used in place of a nested local optimisation. This allows for parallel execution as it effectively decouples global and local optimisation, but the accuracy of the procedure is limited to that of the response surface. The comparison of two level approaches is summarised in Table 3.1.

Table 3.1: Comparison of two-level approaches.

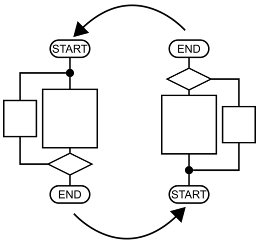
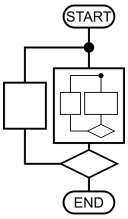
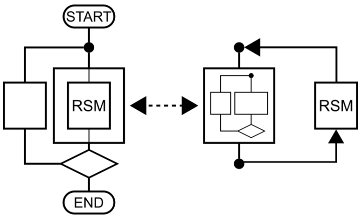
Two-Level Approaches	Comparison
Alternate global–local References: Ciampa et al. [49]	
	Advantages: reduced computational cost Disadvantage: not always applicable, possible optimality or convergence issues
Nested Local References: Noevere and Wilhite [50], Kapania et al. [51, 52, 53, 54, 55, 56, 57]	
	Advantages: accurate and always applicable Disadvantages: computationally expensive

Table 3.1: *Cont.*

Two-Level Approaches	**Comparison**
Nested response surface References: Liu et al. [61], Ragon et al. [63, 64]	
	Advantages: reduced computational cost Disadvantages: limited to the accuracy of the response surface

3.5 Global-local analysis techniques

3.5.1 Specified boundary displacements

A simple way of coupling a global analysis and a local analysis, based on a refined model, is to enforce the solution at interface. In the context of displacement based FE-analyses, this coupling is known as Specified Boundary Displacements (SBD). It can be enforced via a master-slave elimination procedure [65] or using a Lagrange multipliers formulation [66]. It is also possible to develop an iterative procedure to apply a global solution at the boundary of a local model and use the local solution to correct the global results, as in [67, 68, 69]. When combined with an interpolation of the solution fields, this approach can be used with non-conforming meshes [70] and to couple the solution field obtained with different numerical methods [71].

3.5.2 Specified boundary stiffness/force

A second coupling strategy, known as Specified Boundary Stiffness/Force (SBSF) and introduced in [72], specifies the solution further from the interface. As SBD, it applies a term to the load vector, but additionally it defines a stiffness contribution to account for the part of the structure between the Degrees Of Freedom (DOFs), for which the displacement field is specified, and the interface.

Let a structure to analyse be partitioned in two subdomains, where z are the DOFs related to one subdomain, o are the DOFs related to the other subdomain and i are the DOFs at the interface between the subdomains.

The linear system of equations for the analysis of the whole model is:

$$\begin{bmatrix} K_{zz} & K_{zi} & 0 \\ K_{iz} & K_{ii}^1 + K_{ii}^2 & K_{ao} \\ 0 & K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} z \\ i \\ o \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p_i^1 \\ p_i^2 \end{bmatrix} \tag{3.12}$$

where K_{ii}^1 is the stiffness contribution of one subdomain and K_{ii}^2 is the stiffness contribution of the other subdomain. The same holds for the force entry $p_i + p_i^1$.

Let $\bar{o}$ be an approximation of the displacement field over one subdomain. If it is applied as Dirichlet BC, one obtains:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii}^1 + K_{ii}^2 \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p_i^1 - K_{ao}\bar{o} \end{bmatrix} \tag{3.13}$$

The method is called SBSF, because it modifies the linear system of one subregion by adding a stiffness, K_{ii}^2, and a force, $p_i^1 - K_{ao}\bar{o}$.

The accuracy of the method depends on the approximation of the displacement field, if $\bar{o}$ is exact the method is exact. One way to obtain $\bar{o}$ could be to solve a model for the entire structure, with a coarse representation for z. The solution o obtained could then be used to compute z on a finer model.

3.5.3 Static condensation

Another approach for the coupling of FE-analyses is static condensation [73], also known as substructuring. In this case, the stiffness of one model is condensed, by computing the Schur complement of its stiffness matrix, and used for the solution of the other model.

Given a model with the discrete equation:

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} a \\ o \end{bmatrix} = \begin{bmatrix} p_a \\ p_o \end{bmatrix} \tag{3.14}$$

where a are the DOFs at the interface with the other model, the condensed system is:

$$\left[K_{aa} - K_{ao}K_{oo}^{-1}K_{oa}\right] a = p_a - K_{ao}K_{oo}^{-1}p_o \tag{3.15}$$

Stiffness and load vector from the condensed system can then be added to the discrete equation of the other model to realise the coupling. In the case of a static analysis, this results in an exact procedure (Appendix A).

3.5.4 Zooming

Another method derived from condensation to efficiently update a coarse solution with a more detailed analysis and known as zooming was described in [74, 75]. The method is based on a single model with different levels of local refinement, in which the region of interest is known a priori. Instead of having a local analysis coupled with the global analysis, zooming allows to efficiently compute the solution on fine mesh based on an available solution for a coarser mesh of the same part.

3.6 Summary

A large number of MDO exists, as well as several techniques for the coupled analysis of multiple models. Nevertheless, when combining multiple FE-models with different levels of accuracy in the context of MDO, the general approach is to adopt a distributed architecture, which splits the optimisation problem in multiple ones. This is a valid approach to exploit additional local design freedom, without having to consider a larger set of design variables in a single optimisation problem, as a monolithic architecture would require. On the other hand, a distributed approach is less capable of capturing the interaction between global and local design variables and relies on the assumption that local changes do not have a macroscopic effect on the load carrying behaviour of the overall structure. Given the strong influence of “non-regular areas” on the load carrying behaviour of the structure and practitioners focus on ensuring local feasibility rather than enabling detailed local design, a monolithic architecture combined with global-local analysis techniques is in this case more appropriate.

Chapter 4

Methodology

This chapter defines the problem to be considered, clarifies the modelling assumptions and presents the theoretical formulation of the approach. In particular, a global-local analysis strategy based on static condensation is presented and the associated sensitivity analysis is derived. The procedure is considered for two possible types of disciplines, static analysis and static aeroelasticity, and multiple local models.

4.1 Problem statement

To address the challenge of insufficient level of detail in the Global Finite Element Model (GFEM) for the early stage MDO, in this research, a global-local design approach is adopted. In the following a global-local multidisciplinary optimisation procedure is presented.

4.1.1 Optimisation problem

The underlying optimisation problem has the standard form:

$$\begin{cases} \text{find} & \text{argmin}_x\, f(x) \\ \text{such that} & g(x, u(x)) \geq 0 \end{cases} \tag{4.1}$$

where f is a functional to be minimised, g are the constraints, x are the design variables and u is the solution of a multidisciplinary global-local analysis. Given that for a typical industrial problem the number of design variables and constraints considered is large, a gradient-based optimisation approach will be adopted. Because of this, as it will be further discussed in Chap. 6, the computation of sensitivities will be the main driver of computational cost.

4.1.2 Disciplines considered

The term multidisciplinary refers to the fact that two different disciplines will be considered:

1. linear static analysis, and
2. static aeroelasticity.

These are not coupled, they provide the solution to different subcases, which affect different constraints $g(x, u(x))$.

Therefore, u denotes

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \tag{4.2}$$

where each u_i is the solution of a different subcase, corresponding to either one of the specified disciplines above.

4.1.3 Global-local modelling

The term global-local refers to the fact that the structure is modelled using multiple FE models, as it will be described in subsection 4.2. Because of the subdivision in multiple models the solution of the analyses and the computation of the sensitivities require a special formulation detailed in in subsections 4.3 and 4.4.

4.1.4 Global-local MDO architecture

The approach is based on a monolithic multidisciplinary feasible architecture, therefore it is implemented as one optimisation procedure, in which in each iteration:

- the multidisciplinary global-local analysis is solved for u,
- the responses f and g are evaluated and
- the sensitivities $\frac{df}{dx}$, $\frac{dg}{dx}$ are computed.

If the convergence criteria are not met, the optimiser uses the gradients $\frac{df}{dx}$, $\frac{dg}{dx}$ to compute the design update and the procedure is repeated, as depicted in Fig. 3.1.

4.2 Modelling assumptions

The approach relies on the assumption that the structure to be analysed requires two different levels of detail. For most of the structure a coarse modelling is deemed sufficient, while instead some parts of the model require a detailed representation of the geometry and a finer mesh. The structure is therefore represented with multiple models: one global model, which represents most of the structure using a coarse mesh, and multiple local models, used to represent detailed parts with a finer mesh.

It is also assumed that the structure is modelled following three main assumptions:

- non-overlapping domains,
- no local-local interfaces,
- conforming interfaces.

The global and local models do not overlap, so that no part of the structure is modelled twice. The structure is partitioned into multiple models.

It is further assumed, that each local model is interfaced only with the global one, so that no local to local interfaces exist, as represented in Fig 4.1.

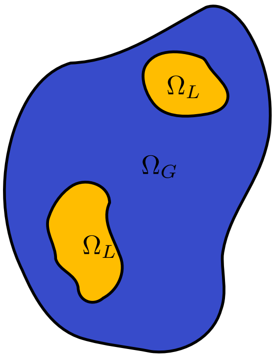


Figure 4.1: The proposed global-local MDO procedure relies on non-overlapping global-local models without local-local interfaces.

At last, without loss of generality, the case with only one local model is considered. With all these assumptions the global solution field u^G can be partitioned into:

- global internal solution z, and
- solution at the global interface i

and the local solution field u^L can be to partitioned into:

- local internal solution o, and
- solution at the local interface a.

Lastly, the interface between global and local models is conforming.

In practice, this means that for each FE-node at the interface of the global model, there is a matching FE-node at the interface of the local model, as represented in Fig 4.2.

Whenever this is not the case, as long as global and local model share an interface with the same geometry, the proposed approach can still be applied, by connecting non-matching meshes using multi-point constraint elemnts. In NASTRAN notation these would be RBE2 and RBE3 elements, which define a rigid body connection and linear homogeneous coupling respectively.

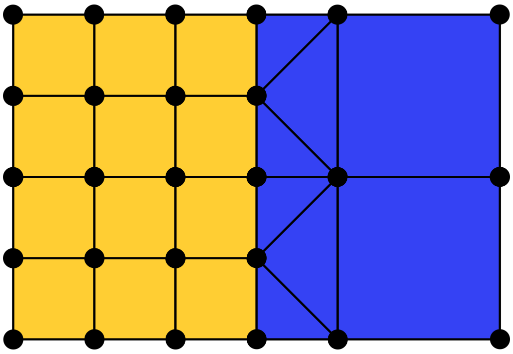


Figure 4.2: The proposed global-local MDO procedure relies on conforming interfaces. Non-matching meshes can be adapted using connecting elements like RBE2 and RBE3.

Since global and local DOFs match, a solution field u on the interface, defined by the global DOFs i, is represented on the local mesh by the same vector of local DOFs. The coupling at the boundary is formulated by simply matching the DOFs:

$$i = a \tag{4.3}$$

therefore the mapping between i and a is the identity matrix.

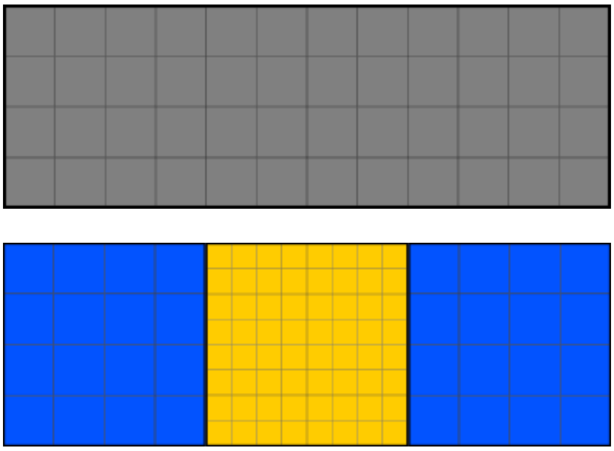


Figure 4.3: Modelling of a structure as a single FE-model (above) and using a global and a local model (below), respectively highlighted in blue and yellow.

When considering a static aeroelasticity subcase, it is assumed that it is sufficient to only interface the global degrees of freedom with the aeroelastic forces, while the local models deform as in a static analysis subcase. This is coherent with the reference procedure, in which the local model does not exist and the aeroelastic loads are injected at the global nodes.

It is further assumed that each design variable can be uniquely assigned to either the global or the local model. Therefore a design variable cannot be part of the global and the local model at the same time.

4.3 Global-local analysis

4.3.1 Discrete form of the equations for static analysis

In the case of a static analysis subcase, the discrete equation is:

$$\begin{bmatrix} K \end{bmatrix} \begin{bmatrix} u \end{bmatrix} = \begin{bmatrix} p \end{bmatrix} \tag{4.4}$$

where K is the stiffness matrix, p is the load vector and u is the vector of nodal displacements.

This holds for both the global:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i \end{bmatrix} \tag{4.5}$$

and the local model:

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} a \\ o \end{bmatrix} = \begin{bmatrix} p_a \\ p_o \end{bmatrix} \tag{4.6}$$

4.3.2 Discrete form of the equations for static aeroelasticity

In the case of static aeroelasticity subcase (Appendix B), the discrete equation is:

$$\begin{bmatrix} K \end{bmatrix} \begin{bmatrix} u \end{bmatrix} = \begin{bmatrix} p \end{bmatrix} + f^A_{rigid} + C \begin{bmatrix} u \end{bmatrix} \tag{4.7}$$

where f^A_{rigid} is the rigid part of the aeroelastic load vector and C is the aeroelastic stiffness matrix.

Since it was assumed that only the global model shares an interface with the aeroelastic forces, the discrete equation of the global model is:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i \end{bmatrix} + f^A_{rigid} + C \begin{bmatrix} z \\ i \end{bmatrix} \tag{4.8}$$

while instead for the local model the discrete equation is again the same as given in eq. 4.6.

4.3.3 On the solution approach

The global-local analysis is based on three steps:

- condensation of the local model, depicted in Fig. 4.4a,
- global solution, depicted in Fig. 4.4b,
- local solution, depicted in Fig. 4.4c.

The static condensation of the local model reduces the system in eq. 4.6 to:

$$\begin{bmatrix} K_{aa} - K_{ao}K_{oo}^{-1}K_{oa} \end{bmatrix} i = \begin{bmatrix} p_a - K_{ao}K_{oo}^{-1}p_o \end{bmatrix} \tag{4.9}$$

$$K^\dagger_{aa} i = p^\dagger_a \tag{4.10}$$

as proven in appendix A. As it will be detailed in section 5.2.1, the implementation does not actually require to compute the inverse matrix K_{oo}^{-1}.

In the second step, the local condensed information is added to the global model:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} \end{bmatrix} \rightarrow \begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} + K^\dagger_{aa} \end{bmatrix} \tag{4.11}$$

$$\begin{bmatrix} p_z \\ p_i \end{bmatrix} \rightarrow \begin{bmatrix} p_z \\ p_i + p^\dagger_a \end{bmatrix} \tag{4.12}$$

In the case of a static analysis subcase, the global system (eq. 4.5) with the local contributions becomes:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} + K^\dagger_{aa} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p^\dagger_a \end{bmatrix} \tag{4.13}$$

While instead, in the case of a static aeroelasticity subcase (eq. 4.8), the system is given by:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} + K^\dagger_{aa} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p^\dagger_a \end{bmatrix} + f^A_{rigid} + C \begin{bmatrix} z \\ i \end{bmatrix} \tag{4.14}$$

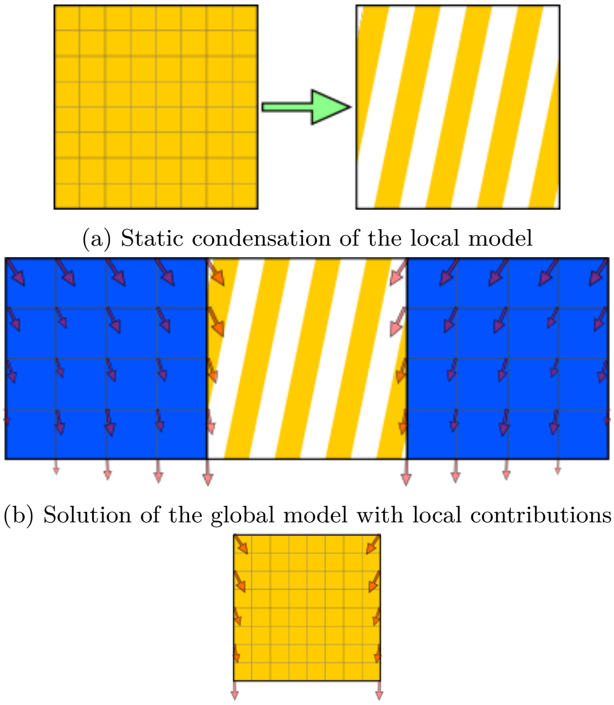
The proof that solving the global system after having added the local condensed contributions is equivalent to solving the global and the local system, while enforcing the coupling (eq. 4.3) is given in appendix C.

With the condensed local contributions, the global solution can then be computed by solving either eq. 4.13 or eq. 4.14 for $u^G = \begin{bmatrix} z \\ i \end{bmatrix}$.

Lastly, the global solution (i) is applied as a Dirichlet boundary condition at the interface of the local model:

$$K_{oo} o = p_o - K_{oa} \bar{i} \tag{4.15}$$

With this, the local system becomes solvable and o can be computed.



(a) Static condensation of the local model

(b) Solution of the global model with local contributions

(c) Solution of the local model with global solution as boundary condition

Figure 4.4: Global-local analysis steps.

4.4 Global-local sensitivity analysis

4.4.1 Objective and constraints evaluation

The MDO procedure is based on a single objective function. In the following, the objective will always be mass, but the methodology could be extended to other objectives, e.g. range. Since it was assumed that local and global model do not overlap, the mass of the whole structure is the sum of the mass of the global model and the mass of the local model:

$$f = f^G + f^L \tag{4.16}$$

The constraint vector can be assembled by joining global and local constraint vectors:

$$g = \begin{bmatrix} g^G \\ g^L \end{bmatrix} \tag{4.17}$$

4.4.2 Sensitivities of objective and constraints

When computing the sensitivities, the design variables can be divided in global x^G and local x^L. In the case of the objective function:

$$\frac{df}{dx} = \begin{bmatrix} \frac{df}{dx^G} & \frac{df}{dx^L} \end{bmatrix} \tag{4.18}$$

and since the mass of a model does not depend on the design variables of other models:

$$\frac{df}{dx} = \begin{bmatrix} \frac{df^G}{dx^G} & \frac{df^L}{dx^L} \end{bmatrix} \tag{4.19}$$

Thus, the sensitivity of the objective function is obtained by assembling independent contributions from the global and the local model. The global-local formulation does not require any special treatment.

As for the constraints vector, computing the derivative with respect to global and local design variables one obtains:

$$\frac{dg}{dx} = \begin{bmatrix} \frac{dg^G}{dx^G} & \frac{dg^G}{dx^L} \\ \frac{dg^L}{dx^G} & \frac{dg^L}{dx^L} \end{bmatrix} \tag{4.20}$$

$$= \begin{bmatrix} \frac{\partial g^G}{\partial x^G} + \frac{\partial g^G}{\partial u^G}\frac{du^G}{dx^G} & \frac{\partial g^G}{\partial u^G}\frac{du^G}{dx^L} \\ \frac{\partial g^L}{\partial u^L}\frac{du^L}{dx^G} & \frac{\partial g^L}{\partial x^L} + \frac{\partial g^L}{\partial u^L}\frac{du^L}{dx^L} \end{bmatrix} \tag{4.21}$$

Within the off-diagonal sub-blocks, the terms $\frac{du^L}{dx^G}$ and $\frac{du^G}{dx^L}$ represent the coupling between global and local sensitivities. The next section explains how these can be computed.

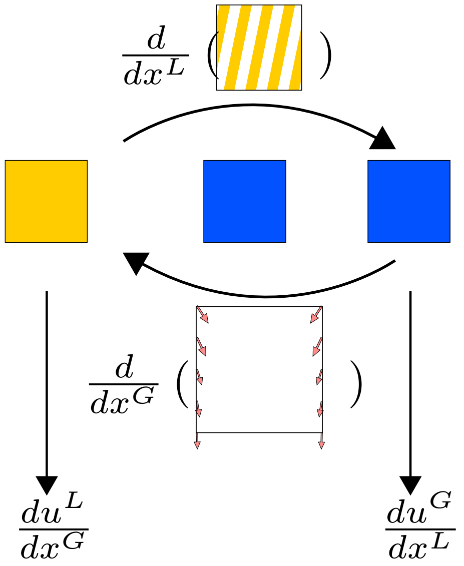


Figure 4.5: Computation of solution field sensitivities. The global model requires the derivative of the condensed contributions with respect to x^L to compute $\frac{du^G}{dx^L}$. The local model requires the derivative of the global solution field with respect to x^G to compute $\frac{du^L}{dx^G}$.

4.5 Coupled sensitivities of the solution field

4.5.1 Coupled sensitivity of the global solution field

The term $\frac{du^G}{dx^L}$ represents the sensitivity of the global solution field with respect to local design variables. The global solution field can be the solution of a static analysis or of a static aeroelastic analysis.

In the case of static analysis, u^G is the solution of:

$$\begin{bmatrix} K^G \end{bmatrix} \begin{bmatrix} u^G \end{bmatrix} = \begin{bmatrix} p^G \end{bmatrix} \tag{4.22}$$

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} + K^{\dagger}_{aa} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p^{\dagger}_a \end{bmatrix} \tag{4.13 revisited}$$

The term $\frac{du^G}{dx^L}$ can be obtained by computing the derivative of all terms in eq. 4.22 with respect to x^L. Using the fact that the components of the global stiffness matrix (K_{zz},K_{zi}, K_{iz} and K_{ii}), as well as the components of the global load vector (p_z and p_i)

do not depend on the local design variables, it follows that:

$$K^G \frac{du^G}{dx^L} = \frac{dp^G}{dx^L} - \frac{dK^G}{dx^L} u^G \tag{4.23}$$

$$K^G \frac{du^G}{dx^L} = \begin{bmatrix} \frac{dp_z}{dx^L} \\ \frac{dp_i}{dx^L} + \frac{dp_a^\dagger}{dx^L} \end{bmatrix} - \begin{bmatrix} \frac{dK_{zz}}{dx^L} & \frac{dK_{zi}}{dx^L} \\ \frac{dK_{iz}}{dx^L} & \frac{dK_{ii}}{dx^L} + \frac{dK_{aa}^\dagger}{dx^L} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} \tag{4.24}$$

$$K^G \frac{du^G}{dx^L} = \begin{bmatrix} \cdot \\ \frac{dp_a^\dagger}{dx^L} - \frac{dK_{aa}^\dagger}{dx^L} i \end{bmatrix} \tag{4.25}$$

If instead u^G is the solution of a static aeroelasticity analysis, then the discrete equation is:

$$K^G u^G = p^G \qquad\qquad + f^A_{rigid} + C\, u^G \tag{4.26}$$

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} + K_{aa}^\dagger \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p_a^\dagger \end{bmatrix} + f^A_{rigid} + C \begin{bmatrix} z \\ i \end{bmatrix} \tag{4.14}$$

Since neither the matrix of aerodynamic coefficients C nor the rigid part of the aeroelastic load vector f^A_{rigid} depend on the local design variables, deriving eq. 4.26 with respect to x^L yields:

$$K^G u^G = p^G + f^A_{rigid} + C u^G \tag{4.27}$$

$$\frac{dK^G}{dx^L} u^G + K^G \frac{du^G}{dx^L} = \frac{dp^G}{dx^L} + \frac{df^A_{rigid}}{dx^L} + \frac{dC}{dx^L} u^G + C \frac{du^G}{dx^L} \tag{4.28}$$

$$K^G \frac{du^G}{dx^L} = \frac{dp^G}{dx^L} - \frac{dK^G}{dx^L} u^G + C \frac{du^G}{dx^L} \tag{4.29}$$

where the static part of the pseudo-load vector can be simplified as in the static analysis case:

$$\frac{dP^G}{dx^L} = \frac{dp^G}{dx^L} - \frac{dK^G}{dx^L} u^G \tag{4.30}$$

$$= \begin{bmatrix} \cdot \\ \frac{dp_a^\dagger}{dx^L} - \frac{dK_{aa}^\dagger}{dx^L} i \end{bmatrix} \tag{4.31}$$

yielding:

$$K^G \frac{du^G}{dx^L} = \begin{bmatrix} \cdot \\ \frac{dp_a^\dagger}{dx^L} - \frac{dK_{aa}^\dagger}{dx^L} i \end{bmatrix} + C \frac{du^G}{dx^L} \tag{4.32}$$

Finally, $\dfrac{dp_a^\dagger}{dx^L}$ and $\dfrac{dK_{aa}^\dagger}{dx^L}$ must be computed. By deriving $K_{aa}^\dagger$ and $p_a^\dagger$ as defined in eq. 4.9:

$$p_a^\dagger = p_a - K_{ao}K_{oo}^{-1}p_o \tag{4.33}$$

$$K_{aa}^\dagger = K_{aa} - K_{ao}K_{oo}^{-1}K_{oa} \tag{4.34}$$

one obtains [1]:

$$\frac{dp_a^\dagger}{dx} = \frac{dp_a}{dx} - \frac{dK_{ao}}{dx}K_{oo}^{-1}p_o + K_{ao}K_{oo}^{-1}\frac{dK_{oo}}{dx}K_{oo}^{-1}p_o - K_{ao}K_{oo}^{-1}\frac{dp_o}{dx} \tag{4.35}$$

and

$$\frac{dK_{aa}^\dagger}{dx} = \frac{dK_{aa}}{dx} - \frac{dK_{ao}}{dx}K_{oo}^{-1}K_{oa} + K_{ao}K_{oo}^{-1}\frac{dK_{oo}}{dx}K_{oo}^{-1}K_{oa} - K_{ao}K_{oo}^{-1}\frac{dK_{oa}}{dx} \tag{4.36}$$

4.5.2 Coupled sensitivity of the local solution field

The remaining coupled sensitivity of the solution field to be computed is:

$$\frac{du^L}{dx^G} = \begin{bmatrix} \frac{da}{dx^G} \\ \frac{do}{dx^G} \end{bmatrix} \tag{4.37}$$

To get $\dfrac{da}{dx^G}$, one must exploit the coupling relationship defined in eq. 4.3.
It follows that:

$$\frac{da}{dx^G} = \frac{da}{di}\frac{di}{dx^G} = \frac{di}{dx^G} \tag{4.38}$$

where $\dfrac{da}{di}$ cancels out because it is the identity matrix.
$\dfrac{do}{dx^G}$ is computed by deriving the discrete equation of the local model, for which only static analysis is considered, i.e. eq 4.15. Since neither the components of the local stiffness matrix nor those of the local load vector are directly influenced by the global design variables, it follows that:

$$\frac{dK_{oo}}{dx^G}o + K_{oo}\frac{do}{dx^G} = \frac{dp_o}{dx^G} - \frac{dK_{oa}}{dx^G}i - K_{oa}\frac{di}{dx^G} \tag{4.39}$$
$$\frac{do}{dx^G} = -K_{oo}^{-1}K_{oa}\frac{di}{dx^G} \tag{4.40}$$

[1] The derivative of an inverse matrix can be computed as $\dfrac{dA^{-1}}{dx} = -A^{-1}\dfrac{dA}{dx}A^{-1}$

4.6 Sensitivity of the local model

The sensitivity of the local model must also account for the coupling with the global model. The derivative of the local constraints with respect to local design variables is given by:

$$\frac{dg^L}{dx^L} = \frac{\partial g^L}{\partial x^L} + \frac{\partial g^L}{\partial u^L}\frac{du^L}{dx^L} \tag{4.41}$$

$\dfrac{\partial g^L}{\partial u^L}$ and $\dfrac{\partial g^L}{\partial x^L}$ do not require any special treatment, but the sensitivity of the solution field $\dfrac{du^L}{dx^L}$ hides a dependency on the coupled sensitivity of the global solution field:

$$\frac{du^L}{dx^L} = \begin{bmatrix} \frac{da}{dx^L} \\ \frac{do}{dx^L} \end{bmatrix} \tag{4.42}$$

The term $\dfrac{da}{dx^L}$ depends on the sensitivity of the global solution at the interface and, as in 4.38, it is obtained by deriving Eq. 4.3:

$$\frac{da}{dx^L} = \frac{da}{di}\frac{di}{dx^L} = \frac{di}{dx^L} \tag{4.43}$$

where $\dfrac{di}{dx^L}$ has been computed as part of Eq. 4.25 or Eq. 4.32, depending on the subcase. An equation for the other term, $\dfrac{do}{dx^L}$, can be obtained by deriving both terms of Eq. 4.15:

$$\frac{dK_{oo}}{dx^L}o + K_{oo}\frac{do}{dx^L} = \frac{dp_o}{dx^L} - K_{oa}\frac{di^G}{dx^L} - \frac{dK_{oa}}{dx^L}i^G \tag{4.44}$$

$$K_{oo}\frac{do}{dx^L} = \frac{dp_o}{dx^L} - \frac{dK_{oo}}{dx^L}o - K_{oa}\frac{di^G}{dx^L} - \frac{dK_{oa}}{dx^L}i^G \tag{4.45}$$

With equation 4.45 and 4.43 one can compute both components of $\dfrac{du^L}{dx^L}$.

4.7 Summary

The global-local MDO methodology presented is based on a monolithic architecture. As such it is sufficient to derive a methodology to extend the solution of the analysis and the computation of sensitivities in each optimisation iteration. The modified global-local analysis is based on three steps: Guyan condensation of each local model, solution of the global model using the computed local information, solution of the local model using the global solution as a boundary condition applied at the interface. The computation of sensitivities is the core of the developed methodology. In particular, for the computation of the derivatives of the constraints it is possible to reuse the available functionalities for the computation of the derivative of global constraints with respect to global design variables. Instead, the mixed derivatives of the constraints require special care, particularly for the computation of the derivative of the solution fields, which must account for the coupling between global and local models as defined in the global-local analysis procedure.

Chapter 5

Implementation

This chapter presents the implementation of the global-local MDO introduced in Chap. 4, as an extension of the *Lagrange* software, currently used by Airbus Defense and Space. The detailed description of each algorithm will then be used in the following chapter, when assessing the computational cost of the procedure.

5.1 Overall software architecture

The implementation is based on *Lagrange*, a software developed at Airbus Defense and Space for constrained gradient-based multidisciplinary design optimisation. *Lagrange* implements its own linear FE-solver and computes semi-analytic sensitivities. This means that when computing sensitivities the formulae are developed analytically until the derivative of the stiffness matrix appears. Then $\frac{dK}{dx}$ is computed with a finite difference approximation, so for each design variable x_i:

$$\frac{dK}{dx_i} = \frac{K(x_i + \Delta x_i) - K(x_i)}{\Delta x_i} \tag{5.1}$$

Lagrange is designed to work with one FE-model defined in one input file, but this limitation can be overcome, by using the *Lagrange*/Python API, which controls the analysis or optimisation procedure and provides access to the internal data.

The global-local approach is implemented by extending *Lagrange* through its Python interface. Multiple *Lagrange* instances are created:

1. an instance for the global model (*LAGRANGE.global*),
2. an instance for each local model (*LAGRANGE.local*) and
3. an instance for the optimiser (*LAGRANGE.optimiser*).

A Python script acts as an intermediary between *Lagrange* instances.

The *LAGRANGE.global* and the multiple *LAGRANGE.local* instances are based on standard *Lagrange* input files containing the FE-models and the optimisation information on design variables x and constraints g_i, which are separately defined for each instance. The *LAGRANGE.optimiser* instance is spawned with a dummy *Lagrange* input file. Additional information regarding the optimisation, such as the choice of optimisation algorithm, convergence criteria and optimisation parameters such as the maximum number of optimisation iterations are separately defined in a .json file, which is read by the Python script.

Since the procedure is based on a monolithic architecture, the general flowchart is based on a single optimisation loop as depicted in 3.1. The functionality for the convergence check and the design update is provided by the *LAGRANGE.optimiser* instance.

The multidisciplinary analysis and sensitivity analysis blocks are instead performed at python level and encapsulated in a *GlobalLocalOptimizationProblem* instance. This class internally computes the solution to the multidisciplinary analysis and provides the *LAGRANGE.optimiser* instance with the values of f, g, $\frac{df}{dx}$ and $\frac{dg}{dx}$. To this end the class coordinates the interaction between the *LAGRANGE.global* and the *LAGRANGE.local* instances, through requests to the *Lagrange*/Python API. The main information exchanges are summarised in Fig. 5.1 for the solution of the multidisciplinary analysis and in Fig. 5.2 for the computation of f, g and their sensitivities.

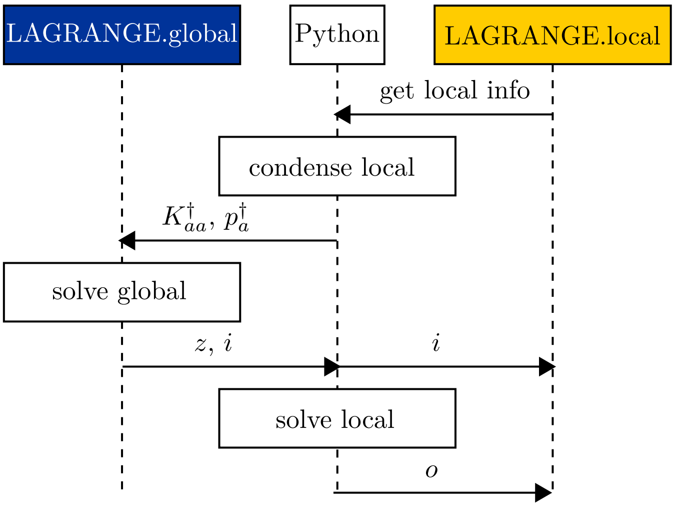


Figure 5.1: Sequence diagram of the global-local analysis.

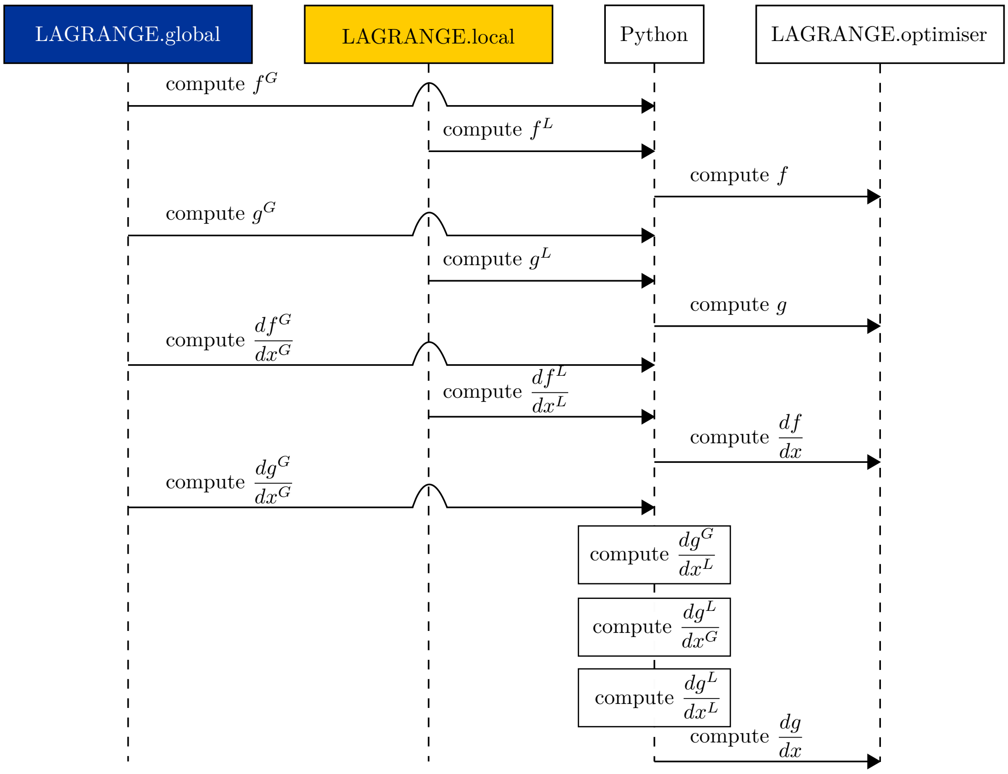


Figure 5.2: Sequence diagram of the global-local sensitivity analysis.

5.2 Multidisciplinary analysis implementation

5.2.1 Condensation

To obtain the local condensed contributions in eq. 4.10 the first step is to divide K^L and p^L in their subblocks.

Then instead of computing K_{oo}^{-1}, K_{oo} is factorised with a sparse LU decomposition. Since K_{oo} is symmetric, a Cholesky decomposition could also have been used.

To compute $K_{ao}K_{oo}^{-1}$, let:

$$y = K_{ao}K_{oo}^{-1} \tag{5.2}$$

Thus:

$$K_{oo}^{\mathsf{T}}y^{\mathsf{T}}= K_{ao}{}^{\mathsf{T}} \tag{5.3}$$

which is solved for y^{T}using the factorised K_{oo}.

The local condensed contributions can then be computed as:

$$K_{aa}^{\dagger} = K_{aa} - yK_{oa} \tag{5.4}$$

$$K_{aa}^{\dagger} = K_{aa} - K_{ao}K_{oo}^{-1}K_{oa} \tag{5.5}$$

and

$$p_a^\dagger = p_a - y p_o \tag{5.6}$$

$$p_a^\dagger = p_a - K_{ao} K_{oo}^{-1} p_o \tag{5.7}$$

Finally the values of $K_{aa}^\dagger$, $p_a^\dagger$, K_{oo}^{-1}, K_{ao}, K_{oa}, p_o, $y = K_{ao} K_{oo}^{-1}$ are cached. The cache is emptied whenever the local design is modified.

The procedure is summarised in Alg. 1.

Algorithm 1: static condensation of local information

```
Function condense
    for subcase = 1 to $n$ do
        if $K$ not cached then
            $K^L$, $p^L$ ← local-interface;
            Kaa, Kao, Koa, Koo = compute_subblocks($K^L$);
            Koo_inv = LU_factorisation(Koo);
            y_trs = solve_transpose(Koo_inv, Kao);
            y = transpose(y_trs);
            x = y × Koa;
            ${K_{aa}}^\dagger$ = Kaa − x;
            cache;
        end
        if $p$ not cached then
            pa, po = compute_subblocks($p^L$);
            y ← cache;
            ${p_o}^\dagger$ = po − y × pa;
            cache;
        end
    end
end
```

5.2.2 Global solution

The local contributions are added to the global model after the global assembly of the stiffness matrix and load vector. Then for each subcase, static or aeroelastic, the global analysis is solved internally by *Lagrange* as a standard FE-analysis.

In the case of a static aeroelastic subcase, rather than bringing $C \begin{bmatrix} z \\ i \end{bmatrix}$ to the left hand side and decomposing $(K - C)$, eq. 4.14 is solved iteratively, because C is dense and non-symmetric.

Algorithm 2: solution of global analyses

```
Function solve_global
    for subcase = $1$ to $n$ do
        $K^{\dagger}_{aa}$, $p^{\dagger}_{a}$ ← local-interface;
        $K^{\dagger}_{aa}$, $p^{\dagger}_{a}$ → global-interface;
        assemble_global() ;                    /* done within LAGRANGE */
        solve() ;                              /* done within LAGRANGE */
        $z$, $i$ ← global-interface;
    end
end
```

5.2.3 Local solution

Lastly, the internal part of the local solution is computed as:

$$K_{oo}o = p_o - K_{oa}\bar{i} \tag{5.8}$$

where p_o and K_{oa} have been cached and K_{oo} has already been factorised.

Algorithm 3: solution of local analyses

```
Function solve_local
    for subcase = $1$ to $n$ do
        $i$ ← global-interface;
        Koa, po ← cache;
        Koo_inv ← cache;
        _p = po − Koa ×$i$;
        $o$ = solve(Koo_inv, _p);
        $o$ → local-interface;
    end
end
```

5.3 Sensitivity analysis implementation

5.3.1 Objective, constraints and gradient of the objective

The values of f, g and $\frac{df}{dx}$ are obtained by assembling their global and local contributions, which are internally computed from the corresponding global and local instances. The procedure is described in Algorithms 4, 5 and 6.

Instead, the computation of the gradient of the constraints $\frac{dg}{dx}$ can be decomposed in the computation of its sub-blocks:

$$\frac{dg}{dx} = \begin{bmatrix} \frac{dg^G}{dx^G} & \frac{dg^G}{dx^L} \\ \frac{dg^L}{dx^G} & \frac{dg^L}{dx^L} \end{bmatrix} \tag{5.9}$$

Algorithm 4: computation of the objective

```
Function compute_f
    f_G ← global-interface ;                /* computed within LAGRANGE */
    f = f_G;
    for local = 1 to $n^L$ do
        f_L ← local-interface ;             /* computed within LAGRANGE */
        f = f + f_L;
    end
end
```

Algorithm 5: computation of the constraints vector

```
Function compute_g
    g_G ← global-interface ;                /* computed within LAGRANGE */
    g = [ g_G ] ;
    for local = 1 to $n^L$ do
        g_L ← local-interface ;             /* computed within LAGRANGE */
        g = [ g, g_L ] ;
    end
end
```

Algorithm 6: computation of the gradient of the objective

```
Function compute_dfdx
    df_G ← global-interface ;               /* computed within LAGRANGE */
    df = [ df_G ] ;
    for local = 1 to $n^L$ do
        df_L ← local-interface ;            /* computed within LAGRANGE */
        df = [ df, df_L ] ;
    end
end
```

as summarised in Alg. 7 and detailed in the following subsections.

Algorithm 7: compute gradients of constraints

```
Function compute_dg
    dg_GG = compute_dg_GG();
    dg_GL, du_GL = compute_dg_GL();
    dg_LG = compute_dg_LG();
    dg_LL = compute_dg_LL(du_GL);
    dg = [ [ dg_GG, dg_GL ], [ dg_LG, dg_LL ] ];
end
```

5.3.2 Gradient of the constraints

The computation of $\frac{dg^G}{dx^G}$ is done internally by *Lagrange*, as described in Alg.8. Instead, the computation of $\frac{dg^G}{dx^L}$, $\frac{dg^L}{dx^G}$ and $\frac{dg^L}{dx^L}$ is coordinated at Python level. These subblocks depend on $\frac{du^G}{dx^L}$, $\frac{du^L}{dx^G}$ and $\frac{du^L}{dx^L}$ respectively, whose computation is based on the coupling between global and local models and thus must be performed at Python level. As detailed in Alg. 9, 10 and 11, once $\frac{du^G}{dx^L}$, $\frac{du^L}{dx^G}$ and $\frac{du^L}{dx^L}$ have been computed, the computation of the implicit and explicit parts of $\frac{dg^G}{dx^L}$, $\frac{dg^L}{dx^G}$ and $\frac{dg^L}{dx^L}$ is delegated to *Lagrange*.

Algorithm 8: computation of the global-global subblock of dg

```
Function compute_dg_GG(args)
    dg_GG ← global-interface;
end
```

Algorithm 9: computation of the global-local subblock of dg

```
Function compute_dg_GL(args)
    for subcase = 1 to $n$ do
        du_GL_sc = compute_dudx_GL(subcase) ;
        du_GL_sc → global-interface;
    end
    du_GL ← global-interface;
    dg_GL = compute_implicit_dgdx_GL(du_GL) ;        /* computed within
     LAGRANGE */
end
```

Algorithm 10: computation of the local-global subblock of dg

```
Function compute_dg_LG(args)
    dg_LG = [ ] ;
    for local = 1 to $n^L$ do
        for subcase = 1 to $n$ do
            du_LG_sc = compute_dudx_LG(subcase) ;
            du_LG_sc → local-interface;
        end
        du_LG ← local-interface;
        dg_LG_local = compute_implicit_dgdx_LG(du_LG) ;        /* computed
         within LAGRANGE */
        dg_LG = [ dg_LG, dg_LG_local ] ;
    end
end
```

Algorithm 11: computation of the local-local subblock of dg

```
Function compute_dg_LL(du_GL)
    dg_LL = [ ] ;
    for local = 1 to $n^L$ do
        for subcase = 1 to $n$ do
            du_GL_sc ← du_GL;
            du_LL_sc = compute_dudx_LL(subcase, du_GL_sc) ;
            du_LL_sc → local-interface;
        end
        du_LL ← local-interface;
        dg_LL_impl = compute_implicit_dgdx_LL(du_LL) ; /* computed within
         LAGRANGE */
        dg_LL_expl ← local-interface ;      /* computed within LAGRANGE */
        dg_LL_local = dg_LL_expl + dg_LL_impl;
        dg_LL = [ dg_LL, dg_LL_local ] ;
    end
end
```

5.4 Sensitivity of the solution fields implementation

5.4.1 Coupled sensitivity of the global displacement field

The computation of $\frac{du^G}{dx^L}$ is based on eq. 4.25, in the case of a static analysis subcase, and on eq. 4.32, in the case of a static aeroelastic subcase.

Both can be solved internally by *Lagrange* once the global instance has been provided with the static part of the pseudo-load vector $\frac{dP^G}{dx^L}$, as described in Alg. 12.

As for the aeroelastic analysis, eq. 4.32 is solved for $\frac{du}{dx}$ with an iterative scheme, instead of inverting $(K - C)$, because C is dense and non-symmetric.

The static part of the pseudo-load vector $\frac{dP^G}{dx^L}$ is obtained according to eq. 4.31 by computing $\frac{dp_a^\dagger}{dx}$ and $\frac{dK_{aa}^\dagger}{dx}$, which must be computed at Python level. The former is computed as defined in eq. 4.35; regarding the latter, it is instead less computationally demanding to directly compute the product $\frac{dK_{aa}^\dagger}{dx}i$, as in:

$$\frac{dK_{aa}^\dagger}{dx}i = \frac{dK_{aa}}{dx}i - \frac{dK_{ao}}{dx}K_{oo}^{-1}K_{oa}i + K_{ao}K_{oo}^{-1}\frac{dK_{oo}}{dx}K_{oo}^{-1}K_{oa}i - K_{ao}K_{oo}^{-1}\frac{dK_{oa}}{dx}i \quad (5.10)$$

By directly computing $\frac{dK_{aa}^\dagger}{dx}i$ instead of $\frac{dK_{aa}^\dagger}{dx}$ the total number of operations is reduced, taking advantage of the fact that i has a single column, allowing all products to be computed between a matrix and a vector.

The terms p_a, p_o, K_{ao}, K_{oa} and a factorisation for K_{oo}^{-1} have already been obtained during the condensation step. The remaining terms $\frac{dp_a}{dx^L}$, $\frac{dp_o}{dx^L}$, $\frac{dK_{aa}}{dx^L}$, $\frac{dK_{ao}}{dx^L}$, $\frac{dK_{oa}}{dx^L}$ and $\frac{dK_{oo}}{dx^L}$ can be obtained as partitions of $\frac{dp^L}{dx^L}$ and $\frac{dK^L}{dx^L}$, which are computed as semi-analytic sensitivities by the local instance within *Lagrange*, as in eq. 5.1.

Algorithm 12: Computation of $\frac{du^G}{dx^L}$

```
Function compute_dudx_GL(subcase)
    for local = 1 to $n^L$ do
        dP_GL_sc = compute_dPdx_GL(local);
        dP_GL_sc → global-interface;
        du_GL_sc ← global-interface ;    /* dependent on subcase type */
    end
end
```

5.4.2 Coupled sensitivity of the local displacement field

The sensitivity of the local solution field with respect to global design variables $\frac{du^L}{dx^G}$ is given by two subblocks, $\frac{da}{dx^G}$ and $\frac{do}{dx^G}$, as defined in eq. 4.37.

To compute $\frac{da}{dx^G}$, according to eq. 4.38, one must use $\frac{di}{dx^G}$, which has been computed internally by *Lagrange*, as part of $\frac{du^G}{dx^G}$, when solving for:

$$\frac{dg^G}{dx^G} = \frac{\partial g^G}{\partial x^G} + \frac{\partial g^G}{\partial u^G}\frac{du^G}{dx^G} \tag{5.11}$$

Thus it can be obtained from the global instance.

The second component can then be computed by solving eq. 4.40, where K_{oa} has been cached and K_{oo} has already been factorised.

The procedure is summarised in Alg. 13.

Algorithm 13: Computation of $\frac{du^L}{dx^G}$

```
Function compute_dudx_LG(subcase)
    di_GG_sc ← global-interface;
    da_LG_sc = di_GG_sc;
    Koa, Koo_inv ← local-interface;
    do_LG_sc = - solve(Koo_inv, Koa × di_GG_sc);
    du_LG_sc = [ da_LG_sc, do_LG_sc ];
end
```

5.4.3 Sensitivity of the local displacement field

The sensitivity of the local solution field with respect to local design variables, $\frac{du^L}{dx^L}$, is also given by two subblocks, $\frac{da}{dx^L}$ and $\frac{do}{dx^L}$, as defined in eq. 4.42.

To compute $\frac{da}{dx^L}$, according to eq. 4.43, one must use $\frac{di}{dx^L}$, which has already been computed as part of $\frac{du^G}{dx^L}$, when solving either eq. 4.25 or eq. 4.32. Thus it can be computed as part of Alg. 9 and passed down from Alg. 7 to Alg. 11 and finally to Alg. 14.

The second component can then be computed by solving eq. 4.45, where K_{oa} and a factorisation for K_{oo} have been cached, i and o have been obtained when solving the multidisciplinary analysis, $\frac{dp_o}{dx^L}$, $\frac{dK_{oa}}{dx^L}$ and $\frac{dK_{oo}}{dx^L}$ can be obtained as partitions of $\frac{dp^L}{dx^L}$ and $\frac{dK^L}{dx^L}$, internally computed by the local instance as semi-analytic sensitivities (eq. 5.1).

Algorithm 14: Computation of $\dfrac{du^L}{dx^L}$

```
Function compute_dudx_LL(subcase, du_GL_sc)
    di_GL_sc ← du_GL_sc;
    da_LL_sc = di_GL_sc;
    Koa, Koo_inv ← local-interface;
    dpo_LL, dKoo_LL, dKoa_LL ← local-interface;
    i, o ← local-interface;
    _dp = dpo_LL- dKoo_LL × o- Koa × di_GL_sc- dKoa_LL × i;
    do_LL_sc = solve(Koo_inv, _dp);
    du_LL_sc = [ da_LL_sc, do_LL_sc ];
end
```

5.5 Summary

The procedure introduced, for the global-local optimisation of aircraft structures, can be successfully implemented as an extension of the existing software *Lagrange*, by leveraging the available *Lagrange*-Python interface, exchanging information between instances for FE-analysis and an optimiser. The implementation of operations, which were not already available within *Lagrange*, like static condensation and the computation of global-local coupled sensitivities of the solution fields, was done directly at Python level. Despite the fact that in this particular case the procedure was implemented to extend *Lagrange*, the ideas behind the implementation are general and can be replicated by connecting other available software for FE-analysis and optimisation.

Chapter 6

Computational cost

This chapter identifies the main drivers of computational cost and explains why a global-local strategy has a limited impact on the overall cost of the MDO procedure.

6.1 Components of computational cost

The cost analysis presented in this chapter is just intended as a means to investigate the influence of various variables on the computational cost and to estimate the effect of introducing a global-local formulation in terms of added complexity. Therefore the focus will be on the procedure, while instead other components of costs, such as those related to the interface between *Lagrange* and Python, are neglected.

A direct measurement of the computational runtime was not performed, as it would not have been indicative of the actual computational cost. The newly implemented part of the procedure, meant as a proof of concept, has not been optimised, contrary to the core of *Lagrange*. Therefore an analysis of runtime would have overestimated the additional computational cost. Furthermore the current implementation is based on a serial formulation of the algorithms, an estimation of the complexity allows to evaluate the benefits that would come from parallelisation.

The computational cost of a monolithic MDO procedure, before the global-local modifications are introduced, would be approximately given by:

$$C_{\text{procedure}} \sim C_{\text{iteration}} \times n^{\text{iterations}} \tag{6.1}$$

This is an approximation, because it does not account for initialisation and finalisation procedures, but mainly because the cost of each iteration, $C_{\text{iteration}}$, is not constant.

Depending on the optimisation algorithm, some information computed in previous iterations may be reused instead of updated and, most importantly, the number of active constraints and design variables considered may vary, dramatically influencing the size of the matrices computed in the sensitivity analysis and the computational cost of the associated iteration.

Nevertheless, the cost of each iteration can be divided into three components:

$$C_{\text{iteration}} = C_{\text{analysis}} + C_{\text{sensitivity}} + C_{\text{design update}} \tag{6.2}$$

where the cost of analysis, C_{analysis}, is the cost of solving all subcases for their solution fields, the cost of sensitivity, $C_{\text{sensitivity}}$, is the cost of computing f, g and their derivatives and the cost of design update, $C_{\text{design update}}$, is the cost of updating the design based on the sensitivities.

The global-local procedure introduced in chapter 4 directly affects C_{analysis} and $C_{\text{sensitivity}}$. It also has an indirect influence on $C_{\text{design update}}$, since this component of cost is influenced by the number of design variables and constraints. The focus of the following sections will be on the first two components, for which it is possible to analyse in detail the effect of the global-local formulation. Instead, the effect on $C_{\text{design update}}$ will only be discussed in general, which does not compromise the validity of this cost analysis, since the other components are more relevant and in particular $C_{\text{sensitivity}} \gg C_{\text{design update}}$.

6.2 Computational cost of analysis

The computational cost of the analysis can be studied by separately considering the cost of condensation, global solution and local solution.

6.2.1 Computational cost of condensation

The cost of condensation is given by

- the cost of dividing the matrix K^L in its four subblocks
- the cost of factorising K_{oo}
- the cost of solving eq. 5.5-5.7

If only the number of multiplications is considered, the cost of condensation is given by the cost of factorising K_{oo}, the cost of computing $y = K_{ao}K_{oo}^{-1}$ and the cost of multiplying yK_{oa} and yp_o. Let n_o be the number of local internal DOFs and let n_a be the number of DOFs at the interface, it follows that:

- $n_o \times n_o$ is the size of K_{oo},
- $n_a \times n_o$ is the size of K_{ao},
- $n_o \times n_a$ is the size of K_{oa}.

Using schoolbook algorithms for the multiplications, the cost is then given by:

$$C_{\text{condensation}} = C_{\text{LU}}(n_o) + \qquad \text{(factorisation of } K_{oo}\text{)} \tag{6.3}$$

$$+\, n_a n_o n_o + \qquad \left(\text{multiplication of } K_{ao} K_{oo}^{-1}\right) \tag{6.4}$$

$$+\, n_a n_o n_a + \qquad \text{(multiplication of } yK_{oa}\text{)} \tag{6.5}$$

$$+\, n_a n_o \qquad \text{(multiplication of } yp_o\text{)} \tag{6.6}$$

where it can be assumed that $n_o \gg n_a$ and $C_{\text{LU}}(n_o) \sim \frac{2}{3}{n_o}^3 = O({n_o}^3)$.

6.2.2 Computational cost of global solution

The cost of solving the global analysis is given by the cost of computing the Cholesky decomposition of the global stiffness matrix K^G, which is of size $n_z \times n_z$, plus the cost of solving the analysis. In the case of a static analysis subcase, the cost of solving the analysis is given by a backward and a forward substitution, which can be assumed to be $O({n_z}^2)$. Instead, in the case of an aeroelatic analysis subcase, eq. 4.14 must be solved iteratively, which means that a linear system of equations of size n_z must be solved using the available decomposition of K^G a given number of times $n_{\text{IT}}^{\text{aero}}$, one per each iteration. The cost iteration will therefore be the cost of a backward and a forward substitution plus the cost of computing the right-hand-side of the linear system. In total the cost per iteration can be assumed to be $O({n_z}^2)$.

The cost of a static subcase is therefore:

$$C_{\text{global}} = C_{\text{Cholesky}}(n_z) + O({n_z}^2) \tag{6.7}$$

The cost of an aeroelastic subcase is instead:

$$C_{\text{global}} = C_{\text{Cholesky}}(n_z) + n_{\text{IT}}^{\text{aero}} \cdot O({n_z}^2) \tag{6.8}$$

where $C_{\text{Cholesky}}(n_z) \sim \frac{1}{3}{n_z}^3 = O({n_z}^3)$.

The total cost of all subcases is actually less than the sum of the cost of each subcase. That is because the stiffness matrix K^G to be decomposed is often shared by more than one subcase. The matrices K^G are obtained by reducing the global stiffness matrix, which depends only on the FE-mesh and therefore is unique and not subcase-dependent. The reduction depends on the boundary conditions, which are common to many subcases. Let $n_{sc}^{\text{indep.}}$ be the number of independent subcases, then the total cost of the global analysis is:

$$\begin{aligned} C_{\text{global}} &= n_{sc}^{\text{indep.}} \cdot C_{\text{Cholesky}}(n_z) + && \text{(cost of Cholesky)} \qquad (6.9) \\ &+ n_{sc}^{\text{static}} \cdot O({n_z}^2) + && \text{(cost of static analyses)} \qquad (6.10) \\ &+ n_{sc}^{\text{aeroelastic}} \cdot \bar{n}_{\text{IT}}^{\text{aero}} \cdot O({n_z}^2) && \text{(cost of aeroelastic analyses)} \qquad (6.11) \end{aligned}$$

where $\bar{n}_{\text{IT}}^{\text{aero}}$ is the average number of iterations required to solve an aeroelastic analysis subcase.

6.2.3 Computational cost of local solution

The cost of solving the local analysis is the cost of solving the linear system in eq. 4.15 times the total number of subcases:

$$C_{\text{local}} = \left(n_{sc}^{\text{static}} + n_{sc}^{\text{aeroelastic}}\right) \cdot O({n_o}^2) \qquad (6.12)$$

The cost is greatly reduced by the fact that the matrix K_{oo} has already been decomposed, thus, only a backward and a forward substitution are needed.

6.2.4 Total computational cost of the analysis

The total computational cost is then given by the sum of cost of condensation, global solution and local solution:

$$\begin{aligned} C_{\text{analysis}} &= C_{\text{condensation}} + C_{\text{global}} + C_{\text{local}} && (6.13) \\ &= n^{\text{LOCALS}} \left(C_{\text{LU}}(n_o) + n_a {n_o}^2\right) + && (6.14) \\ &+ n_{sc}^{\text{stat}} \left(C_{\text{Cholesky}}(n_z) + O({n_z}^2)\right) + && (6.15) \\ &+ n_{sc}^{\text{aero}} \left(C_{\text{Cholesky}}(n_z) + n_{\text{IT}}^{\text{aero}} \cdot O({n_z}^2)\right) + && (6.16) \\ &+ n^{\text{LOCALS}} \cdot n_{sc}^{\text{total}} \left(O({n_o}^2)\right) && (6.17) \end{aligned}$$

where the cost of computation and local solutions have been multiplied by the number of local models (n^{LOCALS}).

6.3 Computational cost of sensitivity

The cost of sensitivity $C_{\text{sensitivity}}$ is generally the largest contributor to the total computational cost and is itself dominated by the cost of computing $\dfrac{dg}{dx}$.

Since generally in structural optimisation the number of constraints is larger than the number of design variables, the following cost analysis is based on the assumption that a direct method is used (see Appendix E).

Since

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \frac{\partial g}{\partial u}\frac{du}{dx} \tag{6.18}$$

$C_{\text{sensitivity}}$ can be divided in the cost of computing the explicit part $\dfrac{\partial g}{\partial x}$, the cost of computing $\dfrac{\partial g}{\partial u}$, the cost of computing $\dfrac{du}{dx}$ and the cost of the product $\dfrac{\partial g}{\partial u} \cdot \dfrac{du}{dx}$ to compute the implicit part.

Of these components the global-local procedure directly affects only the computation of $\dfrac{du}{dx}$. Since $\dfrac{du}{dx}$ is computed differently depending on whether the the displacement field is global or local and on whether the design variables are global or local, the following sections will describe the cost of computing $\dfrac{du^G}{dx^L}$, $\dfrac{du^L}{dx^G}$ and $\dfrac{du^L}{dx^L}$ separately.

6.3.1 Computational cost of $\dfrac{du^G}{dx^L}$

In the case of $\dfrac{du^G}{dx^L}$, the cost will be given by the cost of computing the right-hand side and solving either eq. 4.25 or eq. 4.32 depending on whether the global subcase is a static analysis or an aeroelastic analysis.

The cost of computing $\dfrac{dP}{dx}$ can be estimated in the order of $O({n_o}^3)$, as proven in the Appendix D. If the subcase is an aeroelastic analysis, eq. 4.32 is solved iteratively and in every iteration the right hand side must be updated with the product $C\dfrac{du^G}{dx^L}$.

The linear system to solve is of n_z size and K^G has already been decomposed.

With this assumption the cost of computing $\dfrac{du^G}{dx^L}$ is given by:

$$\begin{aligned} C_{\text{dudx}} &= n_{sc}^{\text{total}} O({n_o}^3) \cdot n_L + n_{sc}^{\text{aero}} \bar{n}_{\text{IT}}^{\text{aero}} O({n_z}^3) \cdot n_L && \text{(cost of computing RHS)} \quad (6.19) \\ &+ n_{sc}^{\text{stat}} O({n_z}^2) \cdot n_L + && \text{(solving eq. 4.25)} \quad (6.20) \\ &+ n_{sc}^{\text{aero}} \bar{n}_{\text{IT}}^{\text{aero}} O({n_z}^2) \cdot n_L && \text{(solving eq. 4.32)} \quad (6.21) \end{aligned}$$

6.3.2 Computational cost of $\dfrac{du^L}{dx^G}$

The cost of computing $\frac{du^L}{dx^G}$ is instead given by:

$$C_{\text{dudx}} = n_{sc}^{\text{total}} \cdot n_o n_a n_a n_G \qquad (\text{computing } K_{oa}\frac{di}{dx}) \tag{6.22}$$

$$+\, n_{sc}^{\text{total}} \cdot O({n_o}^2) n_G \qquad (\text{cost of computing } \frac{du}{dx} \text{ for each subcase}) \tag{6.23}$$

The cost of solving for $\dfrac{du^L}{dx^G}$ is given only by a backward and a forward substitution because K_{oo}^{-1} is already factorised.

6.3.3 Computational cost of $\dfrac{du^L}{dx^L}$

Regarding the cost of computing $\dfrac{du^L}{dx^L}$, $\dfrac{da}{dx^L}$ has already been computed, while the cost of computing $\dfrac{do}{dx^L}$ is given by the cost of assembling the right-hand-side of eq. 4.45 and solving the linear system. To compute the right-hand-side all components are already available so the cost is essentially the cost of computing $\dfrac{dK_{oo}}{dx^L}o$, $K_{oa}\dfrac{di^G}{dx^L}$ and $\dfrac{dK_{oa}}{dx^L}i^G$. The cost of solving the linear system is that of a backward and forward substitution, since K_{oo} has already been factorised.

$$C_{\text{dudx}} = n_{sc}^{\text{total}} \cdot O({n_o}^2) n_L \qquad (\text{cost of solving the linear system}) \tag{6.24}$$

$$+\, n_{sc}^{\text{total}} \cdot O({n_o}^3) n_L \qquad (\text{cost of computing } \frac{dK_{oo}}{dx^L}o) \tag{6.25}$$

$$+\, n_{sc}^{\text{total}} \cdot O(n_o {n_a}^2) n_L \qquad (\text{cost of computing } K_{oa}\frac{di^G}{dx^L}) \tag{6.26}$$

$$+\, n_{sc}^{\text{total}} \cdot O(n_o {n_a}^2) n_L \qquad (\text{cost of computing } \frac{dK_{oa}}{dx^L}i^G) \tag{6.27}$$

6.4 Effect of global-local methodology on computational cost

In a typical problem of industrial size the number of constraints is in the range of millions (10^6) the number of degrees of freedom can reach the hundreds of thousands (10^5) and the number of design variables ranges from several hundreds (10^2) up to thousands (10^3). To reduce the impact on computational time and memory requirements that the size of the problem would otherwise have, an active set strategy is used, which effectively reduces the number of constraints considered, by ignoring a part of them based on the severity of the constraint violation. Furthermore, since for a given constraint the sensitivity to a change of some design variables are important, but for most design variables the sensitivity is negligible, it is also possible to apply a filtering technique to reduce the set of design variables taken into account in an optimisation iteration.

The size of the typical problem and the use of design variables filtering and active set strategy are important aspects to consider, when evaluating the effect of a global-local methodology on the total computational cost and consider the impact of different terms. From the analysis of cost components, the cost of computing $\frac{du}{dx}$ alone is larger than the cost of solving the analyses C_{analysis}, because it scales with the number of design variables. It follows, that the cost of sensitivities $C_{\text{sensitivity}}$ has a major impact on the cost of the whole procedure.

The cost of sensitivity is not only given by the cost of computing $\frac{du}{dx}$. Since the number of constraints is much larger than the number of design variables the cost of computing $\frac{\partial g}{\partial u}$ is large when compared to the cost of computing $\frac{du}{dx}$. Lastly, simply due to the size of the components, the computation of the product $\frac{\partial g}{\partial u} \cdot \frac{du}{dx}$ has a significant impact on the overall computational cost.

Ultimately, the number of constraints and design variables has an impact on the cost of computing the design update $C_{\text{design update}}$. This depends on the chosen optimisation algorithm and is secondary to $C_{\text{sensitivity}}$, but one can assume that it scales with the number of constraints and design variables with cubic complexity.

Therefore one can conclude that controlling the number of constraints and design variables is the most important aspect of containing the total computational cost. The general impact of adopting a global-local strategy is limited, thanks to the use of active set strategy and design variables filtering.

6.4.1 Effect of local design variables

Whenever local design variables are updated local condensed information must be recomputed, which affects the cost of the analysis as well as the cost of computing $\frac{dg^G}{dx^L}$ and $\frac{dg^L}{dx^L}$. Controlling the update of local design variables would therefore lead to computational savings.

In the extreme case of $n_L = 0$, local condensation would be performed only once and the size of $\frac{dg^G}{dx^L}$ and $\frac{dg^L}{dx^L}$ would be zero. Whenever local condensed information can be saved and reused either because $n_L = 0$ or because the local model has not been updated, this leads to computational savings. The $n_L = 0$ example would still be an advantageous procedure compared to the reference one, because, although the design of local model would be fixed, it would be possible to evaluate local constraints and to modify global design variables to ensure local feasibility.

In any case, it is advisable to contain the number of local design variables to limit computational cost, which should not be a problem, since the motivation for this work is to ensure local feasibility, rather than improve the design of local components. Another reason to reduce the number of local design variables, besides the general cost of additional design variables, is that it affects the size of $\dfrac{du^G}{dx^L}$ and $\dfrac{du^L}{dx^L}$ and from the analysis of cost components, it is clear that the cost of computing $\dfrac{du^G}{dx^L}$ and $\dfrac{du^L}{dx^L}$, is larger than the cost of computing $\dfrac{du^L}{dx^G}$.

6.4.2 Effect of local constraints

As for design variables, additional constraints have in general an impact on the overall computational cost, so the definition of local constraints must be limited and their impact on the cost of the procedure must be measured, considering the effect on the total number of constraints.

Nevertheless, if the global-local methodology is coupled with an active set strategy, the computational cost is only affected by the number of active local constraints. Whenever inactive, the additional constraints introduced at local level have a negligible impact.

Whenever a constraint is inactive, $\dfrac{du^L}{dx}$ must still be computed if the corresponding displacement field is relevant for another active constraint. If instead all constraints related to a given $\dfrac{du^L}{dx}$ are inactive, it would be possible to save computational effort on the solution of the analysis and on the computation of $\frac{dg^L}{dx^G}$ and $\frac{dg^L}{dx^L}$. In any case, when a constraint is inactive the cost of sensitivity and design update are reduced.

One can assume that most local constraints will be inactive most of the time and that the advantages derived from an active set strategy will apply to local constraints in the same way as they apply to all other constraints.

6.4.3 Effect of number of local models

Having multiple local models adds to the total computational cost of the procedure, if a serial implementation is assumed. Nevertheless, thanks to the assumption that there are not any local to local interfaces, the costs related to condensation, local solution and sensitivity of displacements fields can all be contained by performing the computation in parallel.

6.4.4 Comparison with local refinement

An natural alternative to adopting a global-local strategy is to refine the model, but the global-local strategy offers some computational advantages. For example, due to the fact that only static analysis is considered at local level, for an aeroelastic subcase the iterative procedure to solve the analysis is performed on a smaller set of degrees of freedom at global level, while the local degrees of freedom are solved only once at convergence. Further, savings come from the fact that thanks to the global-local decomposition, whenever the local design variables are not modified, all local quantities cached during condensation and used in the solution of the analysis and computation of sensitivities do not have to be recomputed. Since this effect depends on the frequency of local updates, combining a global-local strategy with design variables filtering will amplify these savings. Thus the global-local analysis is cheaper and, analogously, the same is true for the sensitivity analysis. The comparison is summarised in table 6.1.

6.5 Summary

It has been shown that the cost of the procedure is approximately given by the number of optimisation iterations multiplied by the average cost of an iteration, which can be decomposed in cost of analysis C_{analysis}, cost of sensitivity $C_{\text{sensitivity}}$ and cost of updating the design $C_{\text{design update}}$.

The cost of sensitivity is considerably larger than the cost of analysis, since it scales linearly with the number of design variables and constraints. The cost of updating the design depends on the optimisation algorithm and has not been studied in detail, but one can assume that, although it scales with the number of constraints and design variables with cubic complexity, it is secondary to the cost of sensitivity.

The main drivers of computational cost are therefore the number of constraints and the number of design variables considered. The use of techniques such as active set strategy and design variables filtering which effectively reduce the number of design variables and active constraints have a critical impact on the overall computational cost. Therefore the true cost of adopting a global-local approach must be studied in conjunction with the adoption of theses strategies.

The introduction of additional local constraints tends to increase the computational cost, but the effect is limited, if the procedure uses an active set strategy: local constraints will be mostly inactive in the same way global constraints are. Similarly the introduction of additional local design variables is countered by filtering. Nevertheless, since the number of design variables is generally much smaller than the number of constraints, the impact of additional design variables is larger. Furthermore, when analysing the computational cost of computing the components of $\frac{du}{dx}$, it has been shown that the complexity of the components, which scale with the number of local design variables, is larger.

To limit the computational cost one should limit both additional constraints and additional design variables, but particularly design variables, which should not be a problem, since the motivation for this work was to prevent local infeasibility and not allow for detailed local design. In conclusion, if the global-local approach is combined with techniques commonly used in MDO, like design variables filtering, active set strategy and the possibility to analyse multiple local models in parallel, the total cost of the procedure is sufficiently contained.

Comparing the cost of the described global-local methodology against simple local refinement, it has been shown that the cost of the global-local methodology is generally smaller. One reason for this is that the global-local approach only considers static analysis at local level, whenever the global subcase is aeroelastic, the iterative procedure, required for the solution of analysis and sensitivity, is solved only for the global DOFs, while the local solution is computed only once at convergence. Instead, with a local refinement all degrees of freedom would have to be considered during the iterative solution of analysis and sensitivity. Further computational savings may come from the fact that the decomposition of the stiffness matrix has cubic complexity and that in the case of local refinement all degrees of freedom are treated at once, while instead in the global-local approach a global and a local stiffness matrix are separately decomposed. Therefore the global-local methodology could be more efficient, since $O({n_z}^3) + O({n_o}^3) < O((n_z + n_o)^3)$. The reality is that stiffness matrices are symmetric positive definite and most importantly sparse, therefore the complexity of decomposing them is less than cubic. When this is taken into account, the cost of other operations with quadratic complexity becomes relevant and it is harder to identify the most efficient approach. Lastly, the presented global-local methodology is easily parallelised. Since local models do not have any mutual dependency, all operations related to a local model such as static condensation, local solution and computation of coupled sensitivities, can be parallelised across local models.

	local refinement	global-local
C_{analysis} (statics)	$O((n_z+n_o)^3)+O((n_z+n_o)^2)$	$O({n_z}^3)+O({n_z}^2)+O({n_o}^3)+O({n_o}^2)$
C_{analysis} (aeroelasticity)	$O((n_z+n_o)^3)+O((n_z+n_o)^2)+\bar{n}_{\text{IT}}^{\text{aero}}\cdot O((n_z+n_o)^2)$	$O({n_z}^3)+O({n_z}^2)+\bar{n}_{\text{IT}}^{\text{aero}}\cdot O({n_z}^2)+O({n_o}^3)+O({n_o}^2)$
$C_{\text{sensitivity}}$: $\dfrac{du}{dx}$ (statics)	$O(n_z+{n_o}^3)+O((n_z+n_o)^2)$	$O({n_z}^3)+O({n_z}^2)+O({n_o}^3)+O({n_o}^2)$
$C_{\text{sensitivity}}$: $\dfrac{du}{dx}$ (aeroelasticity)	$O(n_z+{n_o}^3)+\bar{n}_{\text{IT}}^{\text{aero}}O((n_z+n_o)^3)+\bar{n}_{\text{IT}}^{\text{aero}}O((n_z+n_o)^2)$	$O({n_z}^3)+\bar{n}_{\text{IT}}^{\text{aero}}O({n_z}^3)+\bar{n}_{\text{IT}}^{\text{aero}}O({n_z}^2)+O({n_o}^3)+O({n_o}^2)$

Table 6.1: Comparison of local refinement and global-local procedure.

Chapter 7

Verification

In this chapter the correct implementation of the procedure is verified, by comparing the results of a global-local problem with its single-model equivalent alternative. First, the concept of an equivalent problem is defined, then a series of examples verifies the correct computation of analysis solutions and sensitivities. In these examples the global-local MDO methodology was evaluated for static and aeroelastic load cases, for simple models with coarse mesh to more complex models of the aircraft components. It will be demonstrated that, for coarse local models models, the resulting error for the solution fields are in the range of machine precision. For finer local models, the error is larger, but the results remain sufficiently accurate. The computation of sensitivity proves to be accurate for simple models, while instead for larger models the numerical error can be relatively large for the smallest entries of the Jacobian. Nevertheless, when the quality of the information is evaluated by comparing the implied search directions the method proves to be accurate.

7.1 Theory of verification

In this chapter, the global-local procedure is verified against *Lagrange*: multiple tests ensure that the procedure has been correctly implemented. In each test the same problem is defined as a single FE-model and with an equivalent global-local decomposition. Then the results of *Lagrange* are compared with the results of the global-local MDO procedure.

This can be done because, applying the global-local methodology to enforce, for a static analysis subcase, Eq. 4.5 and Eq. 4.6, while enforcing the coupling at the interface defined in eq. 4.3, is equivalent to solving:

$$\begin{bmatrix} K_{zz} & K_{zi} & \cdot \\ K_{iz} & K_{ii}+K_{aa} & K_{ao} \\ \cdot & K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} z \\ i \\ o \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p_a \\ p_o \end{bmatrix} \tag{7.1}$$

which would be the system of a reference model, in which all degrees of freedom are considered at once and $i = a$ is denoted as i.

Analogously, in the case of an aeroelastic analysis using the global-local methodology to enforce Eq. 4.8, Eq. 4.6) and the coupling Eq. 4.3), is equivalent to solve system of equations for a reference model representing the entire structure:

$$\begin{bmatrix} K_{zz} & K_{zi} & \cdot \\ K_{iz} & K_{ii}+K_{aa} & K_{ao} \\ \cdot & K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} z \\ i \\ o \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p_a \\ p_o \end{bmatrix} + \begin{bmatrix} f^A_{rigid} \\ \cdot \end{bmatrix} + \begin{bmatrix} C \begin{bmatrix} z \\ i \end{bmatrix} \\ \cdot \end{bmatrix} \tag{7.2}$$

Thus, a global-local decomposition of a structure is equivalent to its representation as a single FE-model, if global and local models considered together yield the exact same mesh. Additionally, in the local model the nodes at the interface are marked with ASET cards. Boundary conditions, loads, material properties, etc. must also be the same. Design variables and constraints must be the same and must be assigned to either the global or the local model: design varibales or constraints defined over multiple models are not allowed. Lastly, all optimisation parameters must be set in the same way.

If the problem is an analysis, the global-local procedure applied to an equivalent decomposition is expected to yield the same solution field. If the problem is an optimisation, the global-local procedure should also yield the same results of *Lagrange*. In practice this is not necessarily the case, minor numerical errors in the computation of the solution fields and their sensitivities cause, as the optimisation progresses, minor differences in the designs being considered. This can lead to even larger differences in the sensitivities and may cause the optimiser to follow a different optimisation path. It is therefore necessary to verify two different things. The first one is that with the same sensitivities the optimiser follows the same optimisation history, as it will be proven with some simple tests. The second thing that must be verified is that the sensitivities are correctly computed and differences are only caused by numerical errors.

7.2 Verification of analysis

7.2.1 Two squares model

The first example, depicted in figure 7.1, is made of two square planes. A force is applied at the center node of the left square and the four corners are fully constrained. The left square, in green, represents the global model, the right one, in blue, represents the local model. Figure 7.2 shows the two separate models. As it can be seen in figure 7.2b, the local model contains additionally some ASET cards defined at the interface.

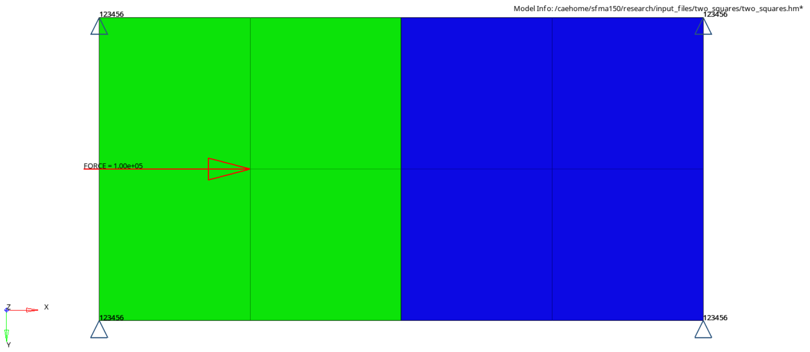
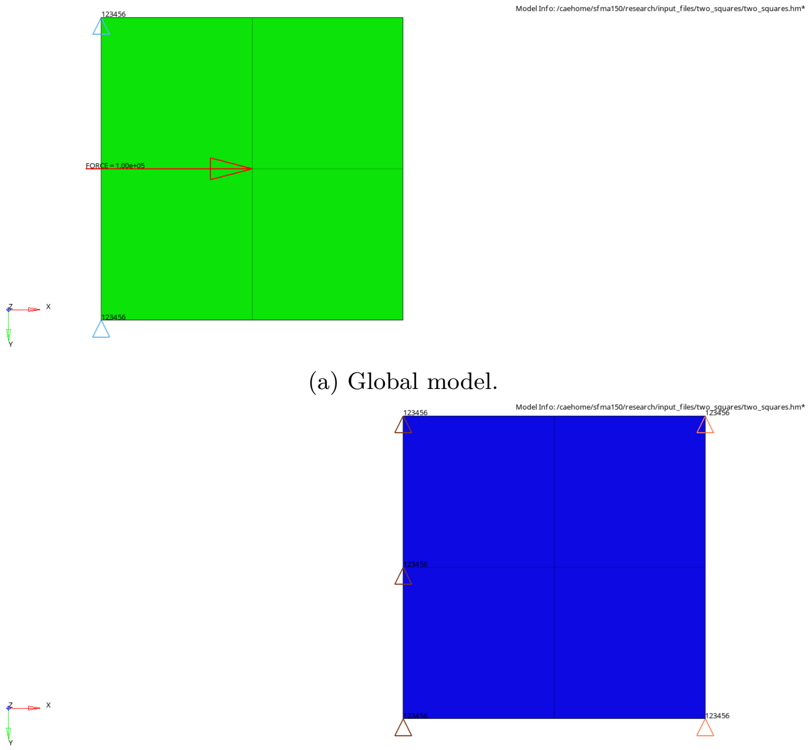

Model Info: /caehome/sfma150/research/input_files/two_squares/two_squares.hm*
123456
123456
FORCE = 1.00e+05
Z
X
Y
123456
123456

Figure 7.1: Example 1: Two-squares example, represented as a single model.


Model Info: /caehome/sfma150/research/input_files/two_squares/two_squares.hm*
123456
FORCE = 1.00e+05
Z
X
Y
123456
Model Info: /caehome/sfma150/research/input_files/two_squares/two_squares.hm*
123456
123456
123456
123456
123456
Z
X
Y

(a) Global model.

(b) Local model.

Figure 7.2: Two squares example, modelled with separate global (a) and local (b) models.

The model can be solved with a linear static analysis of the whole model represented in figure 7.1 or, equivalently, using the global-local procedure for the models in 7.2.

In order to compare the displacement fields, one can measure for each DOF, i.e. for each entry i of the displacement vector, the absolute error:

$$e_{abs}(i) = |u_i - \overline{u_i}| \tag{7.3}$$

and the relative error:

$$e_{rel}(i) = \frac{|u_i - \overline{u_i}|}{|\overline{u_i}|} \tag{7.4}$$

or the percentage error:

$$e_{\%}(i) = \frac{|u_i - \overline{u_i}|}{|\overline{u_i}|} \cdot 100 \tag{7.5}$$

where the reference value $\overline{u_i}$ is the solution of the whole model. The percentage error is meaningless and tends to extreme values, if the displacements being compared are close to zero. For this reason, the computation of $e_{\%}$ has been neglected, whenever $|\overline{u_i}| < 1.000 \cdot 10^{-9}$.

For each error vector one can then compute the L^2 and infinity norm:

$$\|e\|_2 = \sum_{i=1}^{n} e_i^2 \tag{7.6}$$

$$\|e\|_\infty = \max(e_1, \ldots, e_n) \tag{7.7}$$

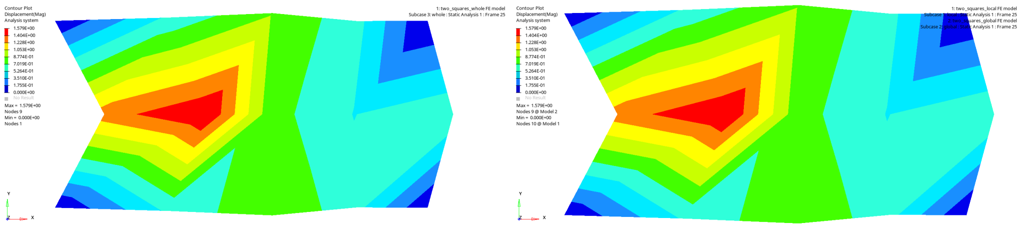
or consider the average entry value:

$$\overline{e} = \frac{e_1 + \ldots + e_n}{n} \tag{7.8}$$

Table 7.1 reports the values of the 2-norm and of the infinity norm of the absolute and percentage error vectors. Figure 7.3 compares the displacement field over the deformed shape scaled by a factor of 30. The comparison of stress and strain fields are not reported, because these quantities are derived from the displacement field.

	e_{abs}	$e_{\%}$
$\|\cdot\|_2$	$9.545 \cdot 10^{-16}$	$2.486 \cdot 10^{-12}$ %
$\|\cdot\|_\infty$	$4.441 \cdot 10^{-16}$	$2.458 \cdot 10^{-12}$ %

Table 7.1: Error metrics.



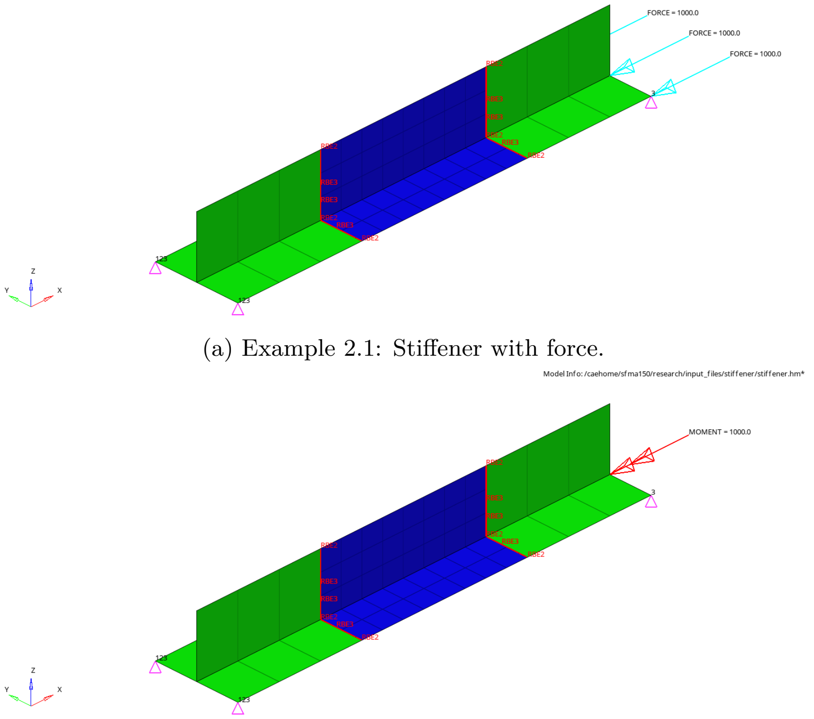
(a) whole

(b) global-local

Figure 7.3: Displacement field over the deformed shape

7.2.2 Stringer model

The second example represents a stringer (figure 7.4). A part in the middle (blue) is extracted from the global model and included in the local model. Two different load sub-cases will be analysed: i) the first model is subjected to horizontal forces (figure 7.4a), and ii) the second model is loaded by a moment (figure 7.4b), which causes a three-dimensional deformation. The translatory DOFs of the corners at the left hand end have been constrained. The corners at the right hand end are prevented from moving vertically.

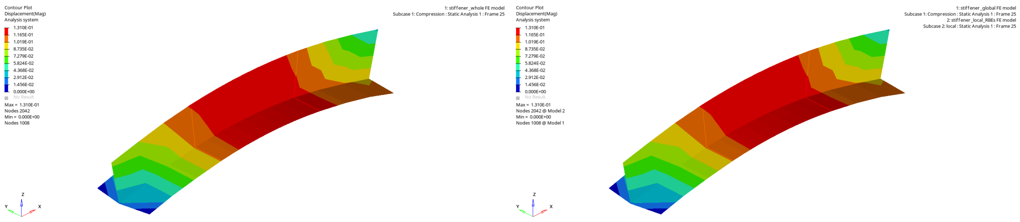


(a) Example 2.1: Stiffener with force.

(b) Example 2.2: Stiffener with moment.

Figure 7.4: Stiffener model (whole).

Contrary to the first example, the local part of the stringer is a 'floating' model, that is, taken alone, the model is underconstrained and cannot be solved. Another relevant difference, which makes this model more complex, is the fact that the local and global mesh are non-conforming and must be linked via multipoint constraints. Depending on whether the multipoint constraints are included in the global or in the local model, the boundary between global and local model changes and ASET cards must be defined accordingly. Both possibilities have been tested, yielding comparatively the same results. As for the first example, the results of the global-local approach can be compared with those of a static analysis of the whole model. Figures 7.5 and 7.6 show a comparison of the deformed shape, scaled by a factor of 300, for the second and the third example. Quantitative evaluations of the error are reported in tables 7.2 and 7.3.



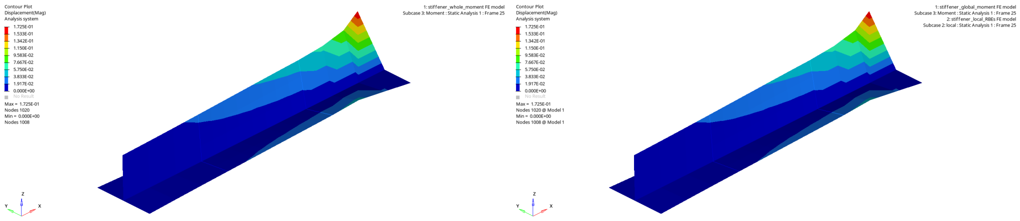
(a) whole

(b) global-local

Figure 7.5: Example 2.1: Stiffener with force. Displacement field over the deformed shape

	e_{abs}	$e_{\%}$
$\lVert\cdot\rVert_2$	$1.802 \cdot 10^{-12}$	$8.395 \cdot 10^{-6}$ %
$\lVert\cdot\rVert_\infty$	$6.949 \cdot 10^{-13}$	$4.683 \cdot 10^{-6}$ %

Table 7.2: Example 2.1: Error metrics.



(a) whole

(b) global-local

Figure 7.6: Example 2.2: Stiffener with moment. Displacement field over the deformed shape

	e_{abs}	$e_{\%}$
$\lVert\cdot\rVert_2$	$6.449 \cdot 10^{-13}$	$3.231 \cdot 10^{-5}$ %
$\lVert\cdot\rVert_\infty$	$2.528 \cdot 10^{-13}$	$2.122 \cdot 10^{-5}$ %

Table 7.3: Example 2.2: Error metrics.

7.2.3 Simple wing model

This an extremely simplified model of a wing that is used as a minimal example for the verification of the aeroelastic analysis.

The structural model represents a beam, with four beam elements with 6 DOFs per node (CBAR in NASTRAN notation), represented in blue in Fig. 7.7.

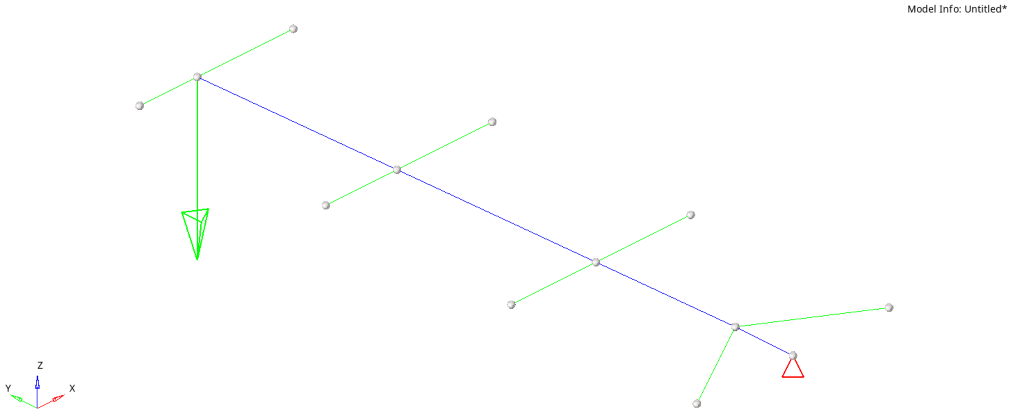


Figure 7.7: Reference model of the simple wing.

Four multi-point constraint elements (RBE2 in NASTRAN notation), represented in green, rigidly connect the beam with structural points whose displacement is linked with the aerodynamic model, represented in Fig. 7.8. The aerodynamic model consists of six panels.

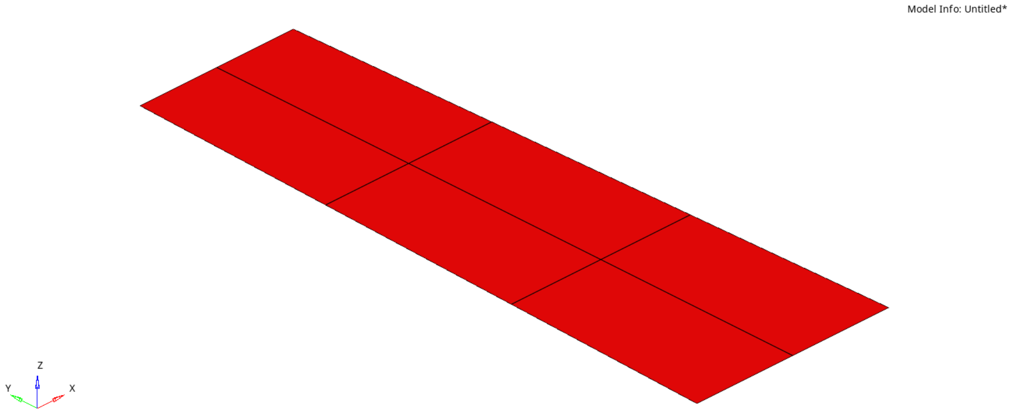


Figure 7.8: Aerodynamic model used in the simple wing.

In the global-local representation of the same structure, one of the middle CBAR elements, shown in yellow, is assigned to the local model, as represented in Fig. 7.9. The remaining part of the structure, represented in blue, is assigned to the global model.

A single SPC fully constrains the freedom of the grid point at the wing root. Two subcases are defined, i) subcase 1: a static analysis, with a downward force applied at

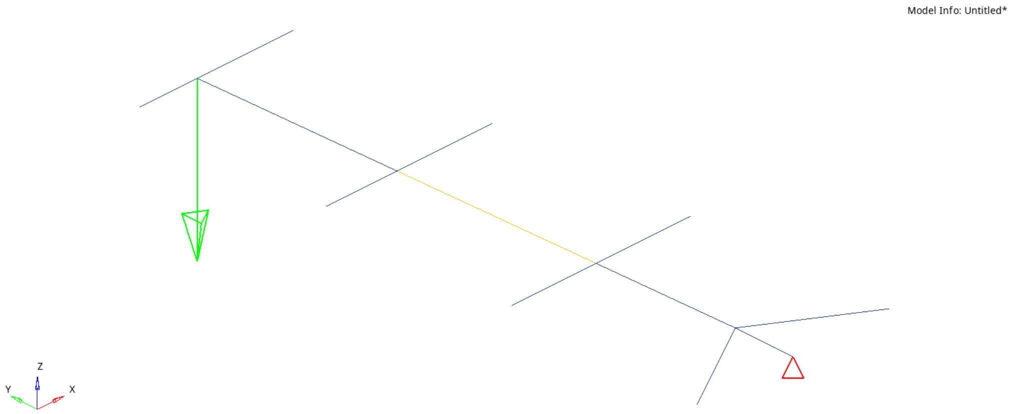


Figure 7.9: Global-local model of the simple wing. The beam element in the middle, highlighted in yellow, is assigned to the local model, all other elements constitute the global model.

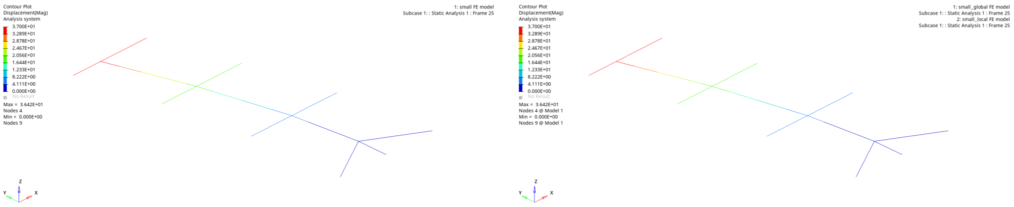


Figure 7.10: Results of the static analysis.

the wing tip, and ii) subcase 2: a static aeroelastic analysis. The model does not have an optimisation deck defined.

For both subcases the results obtained with the reference and the global-local approach are identical, meaning that each component of the solution field has exactly the same floating point representation. Thanks to the small number of Degrees Of Freedom it is possible to report the value of each entry as done in Tables 7.4 and 7.5 , which report the reference and global-local solution of subcase 1, and Tables 7.7 and 7.8 , which report the reference and global-local solution of subcase 2. Since the results compared match exactly, all error metrics report a 0.000, as it can be seen from Tables 7.6 and 7.9 for subcase 1 and subcase 2 respectively. This happens, because for this particular example the local model consists of a single element. Due to the absence of internal degrees of freedom o, condensation does not occur and when the global stiffness matrix is assembled with the local information the operation matches exactly the global assembly of element stiffness matrix in the 'whole' case. Thus the linear system of equations solved is identical and no roundoff errors arise. A visual comparison of the solution field is shown in Figure 7.10 for subcase 1 and in Figure 7.11 for subcase 2.

node ID	component ID					
	1	2	3	4	5	6
1	0.000	0.000	0.406	−0.034	−0.002	0.000
2	0.000	0.000	−5.545	−0.100	−0.004	0.000
3	0.000	0.000	−18.978	−0.161	−0.007	0.000
4	0.000	0.000	−36.416	−0.181	−0.007	0.000
5	0.000	0.000	−36.193	−0.181	−0.007	0.000
6	0.000	0.000	−18.729	−0.161	−0.007	0.000
7	0.000	0.000	−5.352	−0.100	−0.004	0.000
8	0.000	0.000	−0.524	−0.034	−0.002	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.605	−0.034	−0.002	0.000
11	0.000	0.000	−5.132	−0.100	−0.004	0.000
12	0.000	0.000	−18.400	−0.161	−0.007	0.000
13	0.000	0.000	−35.823	−0.181	−0.007	0.000

Table 7.4: Reference displacement field of subcase 1 (statics), by Grid ID (row) and component ID (column).

node ID	component ID					
	1	2	3	4	5	6
1	0.000	0.000	0.406	−0.034	−0.002	0.000
2	0.000	0.000	−5.545	−0.100	−0.004	0.000
3	0.000	0.000	−18.978	−0.161	−0.007	0.000
4	0.000	0.000	−36.416	−0.181	−0.007	0.000
5	0.000	0.000	−36.193	−0.181	−0.007	0.000
6	0.000	0.000	−18.729	−0.161	−0.007	0.000
7	0.000	0.000	−5.352	−0.100	−0.004	0.000
8	0.000	0.000	−0.524	−0.034	−0.002	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.605	−0.034	−0.002	0.000
11	0.000	0.000	−5.132	−0.100	−0.004	0.000
12	0.000	0.000	−18.400	−0.161	−0.007	0.000
13	0.000	0.000	−35.823	−0.181	−0.007	0.000

Table 7.5: Global-local displacement field of subcase 1 (statics), by Grid ID (row) and component ID (column).

	e_{abs}	e_{rel}
$\|\cdot\|_2$	0.000	0.000
$\|\cdot\|_\infty$	0.000	0.000

Table 7.6: error metrics for the static analysis (subcase 1). In this example the local model consists of a single element, condensation does not occur and the results are exact.

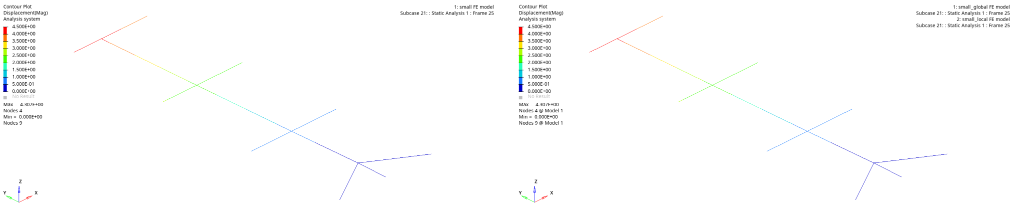


Figure 7.11: Results of the aeroelastic analysis.

node ID	component ID					
	1	2	3	4	5	6
1	0.000	0.000	0.004	0.005	0.002	0.000
2	0.000	0.000	0.927	0.013	0.004	0.000
3	0.000	0.000	2.535	0.017	0.006	0.000
4	0.000	0.000	4.307	0.018	0.006	0.000
5	0.000	0.000	4.124	0.018	0.006	0.000
6	0.000	0.000	2.323	0.017	0.006	0.000
7	0.000	0.000	0.745	0.013	0.004	0.000
8	0.000	0.000	0.079	0.005	0.002	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	−0.150	0.005	0.002	0.000
11	0.000	0.000	0.542	0.013	0.004	0.000
12	0.000	0.000	2.040	0.017	0.006	0.000
13	0.000	0.000	3.819	0.018	0.006	0.000

Table 7.7: Reference displacement field of subcase 21 (aeroelasticity), by Grid ID (row) and component ID (column).

node ID	component ID					
	1	2	3	4	5	6
1	0.000	0.000	0.004	0.005	0.002	0.000
2	0.000	0.000	0.927	0.013	0.004	0.000
3	0.000	0.000	2.535	0.017	0.006	0.000
4	0.000	0.000	4.307	0.018	0.006	0.000
5	0.000	0.000	4.124	0.018	0.006	0.000
6	0.000	0.000	2.323	0.017	0.006	0.000
7	0.000	0.000	0.745	0.013	0.004	0.000
8	0.000	0.000	0.079	0.005	0.002	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	−0.150	0.005	0.002	0.000
11	0.000	0.000	0.542	0.013	0.004	0.000
12	0.000	0.000	2.040	0.017	0.006	0.000
13	0.000	0.000	3.819	0.018	0.006	0.000

Table 7.8: Global-local displacement field of subcase 21 (aeroelasticity), by Grid ID (row) and component ID (column).

	e_{abs}	e_{rel}
$\|\cdot\|_2$	0.000	0.000
$\|\cdot\|_\infty$	0.000	0.000

Table 7.9: error metrics for the aeroelastic analysis (subcase 21). In this example the local model consists of a single element, condensation does not occur and the results are exact.

7.2.4 Coarse plate model

Geometry and boundary conditions

The structural model consists of three quadrilateral and one triangular shell element with 6 DOFs (CQUAD4 and CTRIA3 in NASTRAN notation), represented in Fig. 7.12a. In the global-local representation of the same structure, the middle plate, shown in yellow, is assigned to the local model, as represented in Fig. 7.12b. The remaining part of the structure, represented in blue, is assigned to the global model.

(a) Reference “coarse plate model”. (b) Global-local “coarse plate model”.

Figure 7.12: Alternative representations of the “coarse plate model”.

In both representations, one SPC fully constrains the freedom of the right end of the plate. The same three subcases are defined for both models, a static analysis and two static aeroelastic analyses.

The aerodynamic model, shared by the reference and the global-local models, consists of three panels, which overlap the structural quadrilateral elements exactly, as represented in Fig. 7.13.

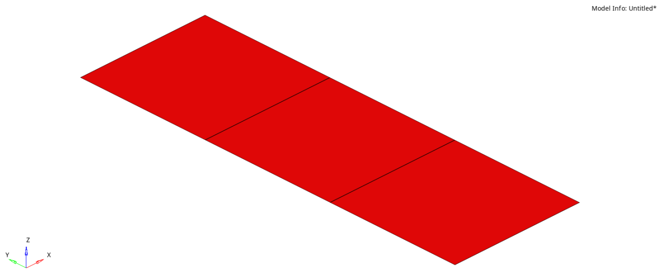
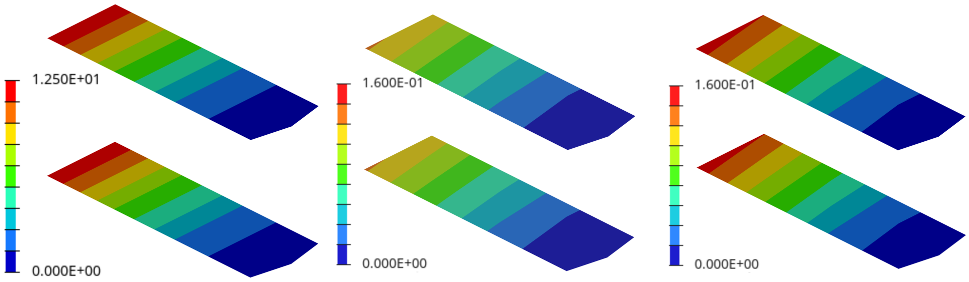


Figure 7.13: Aerodynamic model used in “coarse plate model”.

The optimisation problem associated to this model defines the thickness of each element as a design variable and prescribes a strength constraint per element. The objective is mass minimisation.

Model analysis and optimisation results

The results of the global-local procedure match exactly those of the standard FE-solution of the whole model. This is because for this particular model the global-local representation is such that the local model does not have any internal degree of freedom. In this

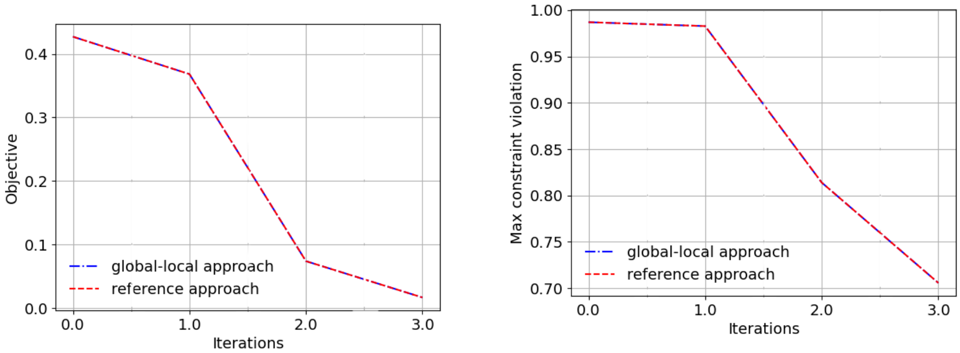


(a) Static subcase 1

(b) Aeroelastic subcase 21

(c) Aeroelastic subcase 22

Figure 7.14: Results of the analyses. Each subfigure compares a different subcase with the displacement field of the reference approach represented above and the global-local solution below.



(a) Objective optimisation history.

(b) Maximum constraint violation optimisation history.

Figure 7.15: Histories of the "coarse plate model" optimisation.

simple case, the element stiffness matrix is identical to the condensed one and does not introduce any numerical error. Thus, as expected, the global solution (eq. 4.13-4.14) is identical to the solution of the whole model (eq. 7.1-7.2).

The optimised design obtained with the two approaches is the same. As it can be seen in Fig. 7.15 objective and maximum constraints violation follow the same optimisation history.

7.2.5 Refined plate model

Geometry and boundary conditions

In the fine version of the model everything remains the same, except for the quadrilateral element in the middle, which is split into two, as represented in Fig. 7.16a.

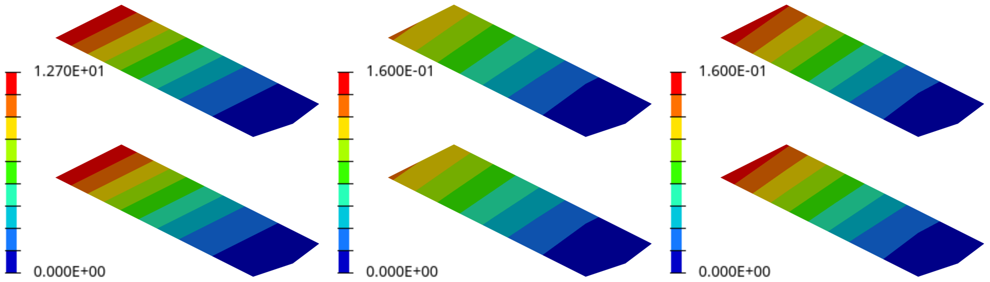
In the global-local representation, the local model consists of two quadrilateral shell elements, as represented in Fig. 7.16b. The optimisation problem is equivalent to the one of the coarse case, the objective remains mass minimisation, but since two quadrilateral

(a) Reference "fine plate model". (b) Global-local "fine plate model".

Figure 7.16: Alternative representations of the "fine plate model".

elements have replaced a single element, two strength constraints must be specified instead of one. The global model remains unchanged.

Model analysis and optimisation results



(a) Static subcase 1 (b) Aeroelastic subcase 21 (c) Aeroelastic subcase 22

Figure 7.17: Results of the analyses. Each subfigure compares a different subcase with the displacement field of the reference approach represented above and the global-local solution below.

As the internal degrees of freedom of the local model are condensed, a negligible numerical error is introduced.

Taking the solution of *Lagrange* u as a reference, one can measure the absolute and relative error of the solution u^{GL} of the global-local approach, as in eq. 7.3-7.4, and consider the L^2 and infinity norm, defined in eq. 7.6-7.7.

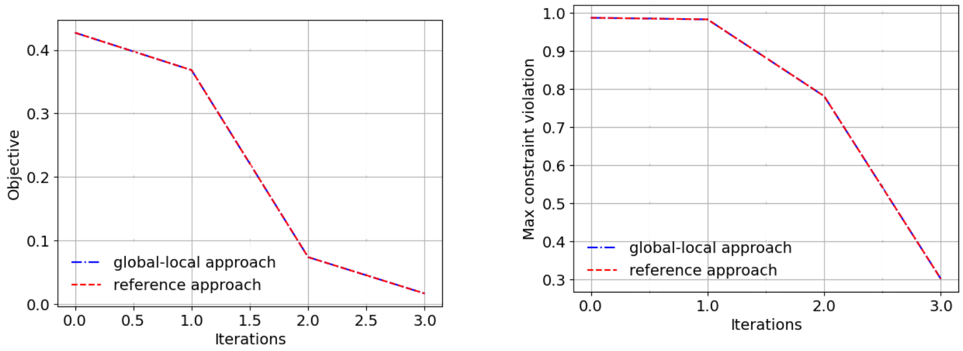
As it can be seen from Table 7.10, the results are not exact, but the error is quite close to machine precision.

Fig. 7.17 shows visual comparison of the displacement fields computed with the reference and the global-local approach.

	static subcase 1	aeroelastic subcase 21	aeroelastic subcase 22
$\|\|e_{abs}\|\|_2$	$5.645 \cdot 10^{-11}$	$6.078 \cdot 10^{-13}$	$7.341 \cdot 10^{-13}$
$\|\|e_{abs}\|\|_\infty$	$3.437 \cdot 10^{-11}$	$3.685 \cdot 10^{-13}$	$4.452 \cdot 10^{-13}$
$\|\|e_{rel}\|\|_2$	$1.407 \cdot 10^{-10}$	$1.743 \cdot 10^{-10}$	$1.651 \cdot 10^{-10}$
$\|\|e_{rel}\|\|_\infty$	$1.050 \cdot 10^{-10}$	$1.642 \cdot 10^{-10}$	$1.541 \cdot 10^{-10}$

Table 7.10: Error metrics for the analysis results.

The numerical error due to the condensation of the local model had a negligible effect on the optimisation results. As for the coarse model, the two approaches yield the same optimal design and the same optimisation history as shown in Fig. 7.18.



(a) Objective optimisation history.

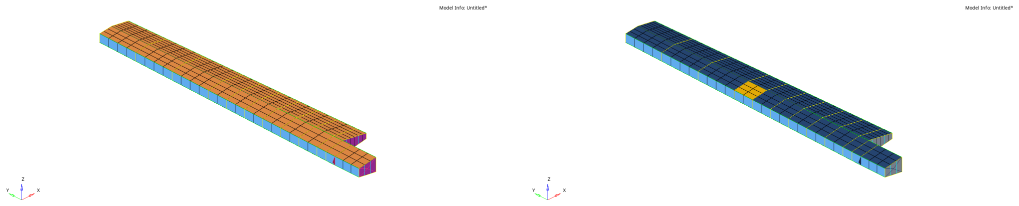
(b) Maximum constraint violation optimisation history.

Figure 7.18: Histories of the "fine plate model" optimisation.

7.2.6 Wingbox coarse analysis

Geometry and boundary conditions

The structural model of the wingbox is depicted in Fig. 7.19. Fig. 7.19a shows the reference structure defined as a single model, while instead Fig. 7.19b shows the global-local representation with the local model highlighted in yellow.



(a) Reference "wingbox coarse model".

(b) Global-local "wingbox coarse model".

Figure 7.19: Alternative representation of the "wingbox coarse model".

The structure is constrained by SPCs at the wing root, which is shown on the right side. Four different subcases are defined for this model. Three static analysis representing a "gust-up", a "landing" and a "manoeuvre" and an aeroelastic subcase, for which the aerodynamic model in Fig. 7.20 is used.

Qualitative evaluation

Figures 7.21, 7.22, 7.23, 7.24 qualitatively capture the accuracy of the computed displacement fields for the different subcases. In each figure, part (a) represents the reference solution, with a countour of the displacement field and a real size deformation.

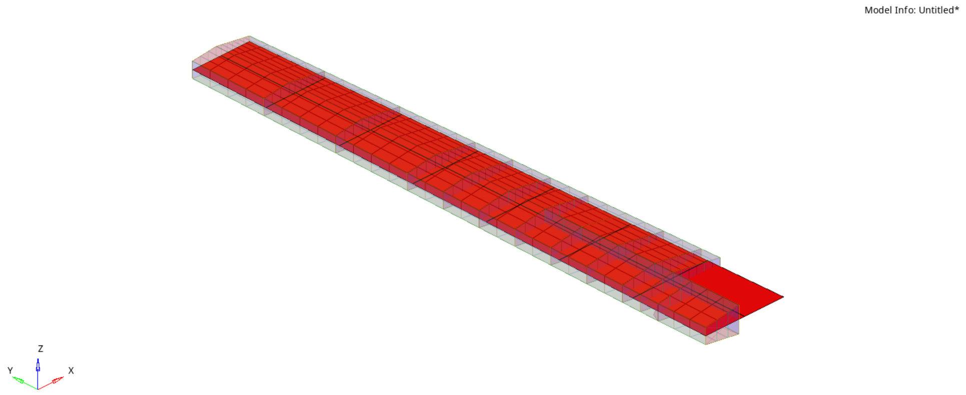


Figure 7.20: Aerodynamic model used in "wingbox coarse model", displayed over the transparent structure.

Part (b) shows a contour plot of the solution field error, that is the difference in displacement between the reference solution and the global local solution: $u^W - u^{GL}$. The deformation depicted represents the difference in displacements scaled by a factor of 1000.

As it can be seen in part (a) of the figures, all subcases lead to visible deformations, with a maximum tip displacement in the order of $1.0 \cdot 10^3$ mm. Part (b) of all figures appears undeformed despite the fact that the deformation is scaled up by a factor of 1000. The maximum displacement is in the order of $1.0 \cdot 10^{-2}$ mm.

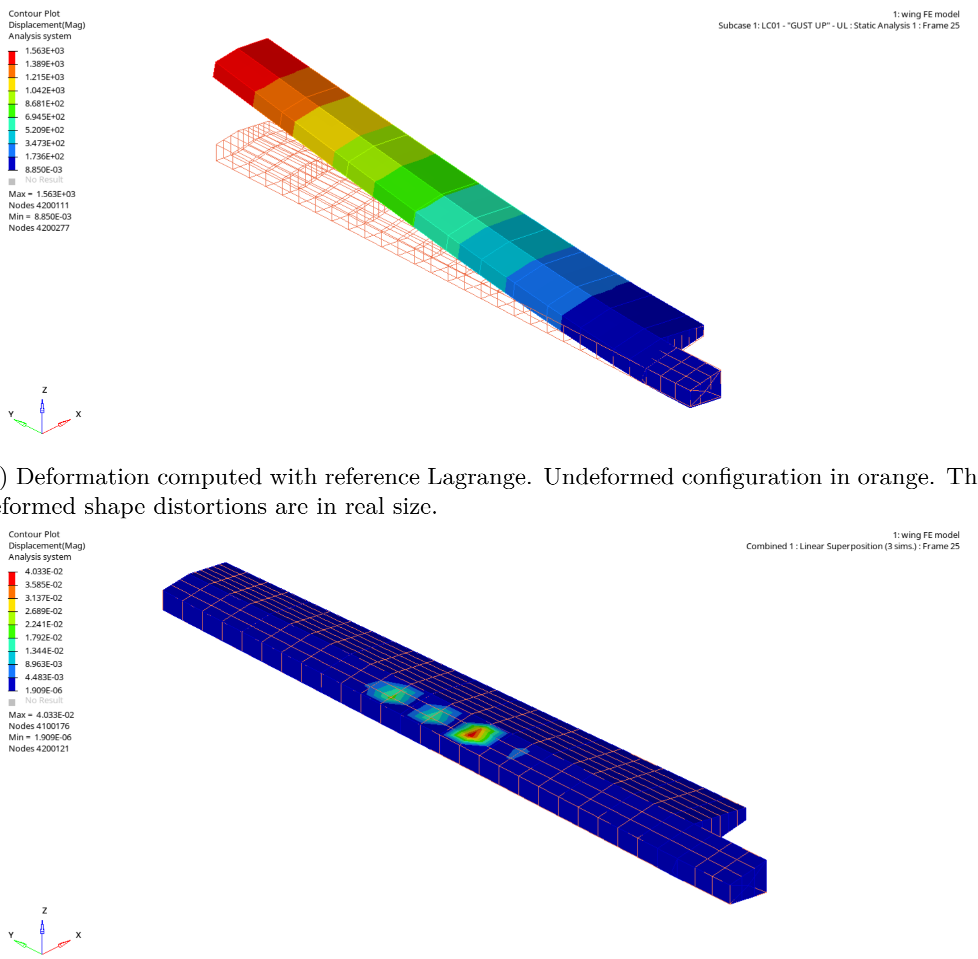
Quantitative evaluation

Tables 7.11, 7.12, 7.13, 7.14 show the average entry value and the infinity norm (eq. 7.8 and eq. 7.7) for the absolute and relative error (eq. 7.3-7.4) in the computation of the displacement fields. The relative error has not been computed for reference displacements smaller than 0.001 mm. Such a threshold value is small when compared with the actual displacements which are generally in the range 1.000 mm - 2500.000 mm.

The metrics reflect what was qualitatively presented in the previous section. The maximum absolute error, $\|e_{abs}\|_\infty$, is small for all subcases. The relative error can be large in some points as highlighted by $\|e_{rel}\|_\infty$, but is on average negligible as indicated by $\overline{e_{rel}}$.

	e_{abs}	e_{rel}
$\overline{\cdot}$	0.000	0.001
$\|\cdot\|_\infty$	0.040	0.124

Table 7.11: Error metrics for subcase 1. The relative error has been computed only for reference displacements larger than 0.001 mm.



(a) Deformation computed with reference Lagrange. Undeformed configuration in orange. The deformed shape distortions are in real size.

(b) Difference in deformation (absolute error) obtained with gl-lagrange. The deformed shape distortions are scaled by a factor of 1000.

Figure 7.21: Results of the static analysis (subcase 1).

	e_{abs}	e_{rel}
$\bar{\cdot}$	0.000	0.000
$\lVert\cdot\rVert_\infty$	0.018	0.032

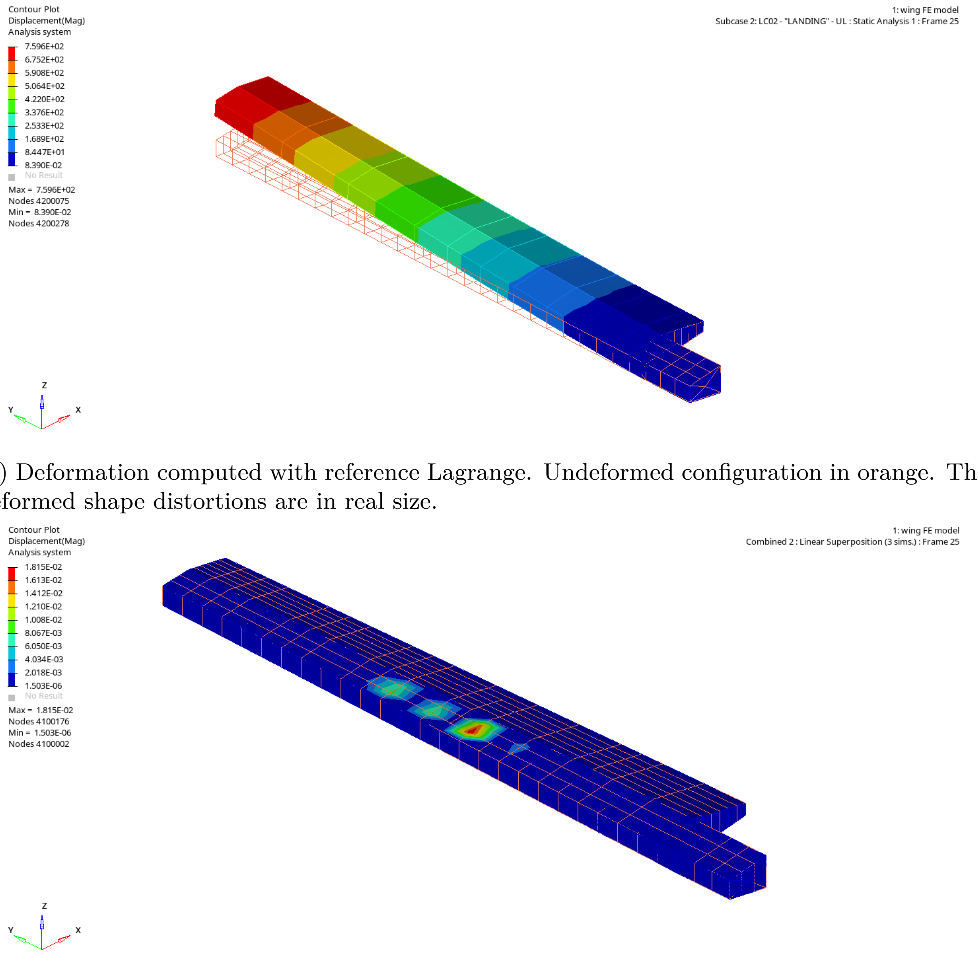
Table 7.12: Error metrics for subcase 2. The relative error has been computed only for reference displacements larger than 0.001 mm.

	e_{abs}	e_{rel}
$\bar{\cdot}$	0.000	0.000
$\lVert\cdot\rVert_\infty$	0.034	0.036

Table 7.13: Error metrics for subcase 3. The relative error has been computed only for reference displacements larger than 0.001 mm.

	e_{abs}	e_{rel}
$\bar{\cdot}$	0.002	0.001
$\lVert\cdot\rVert_\infty$	0.058	0.371

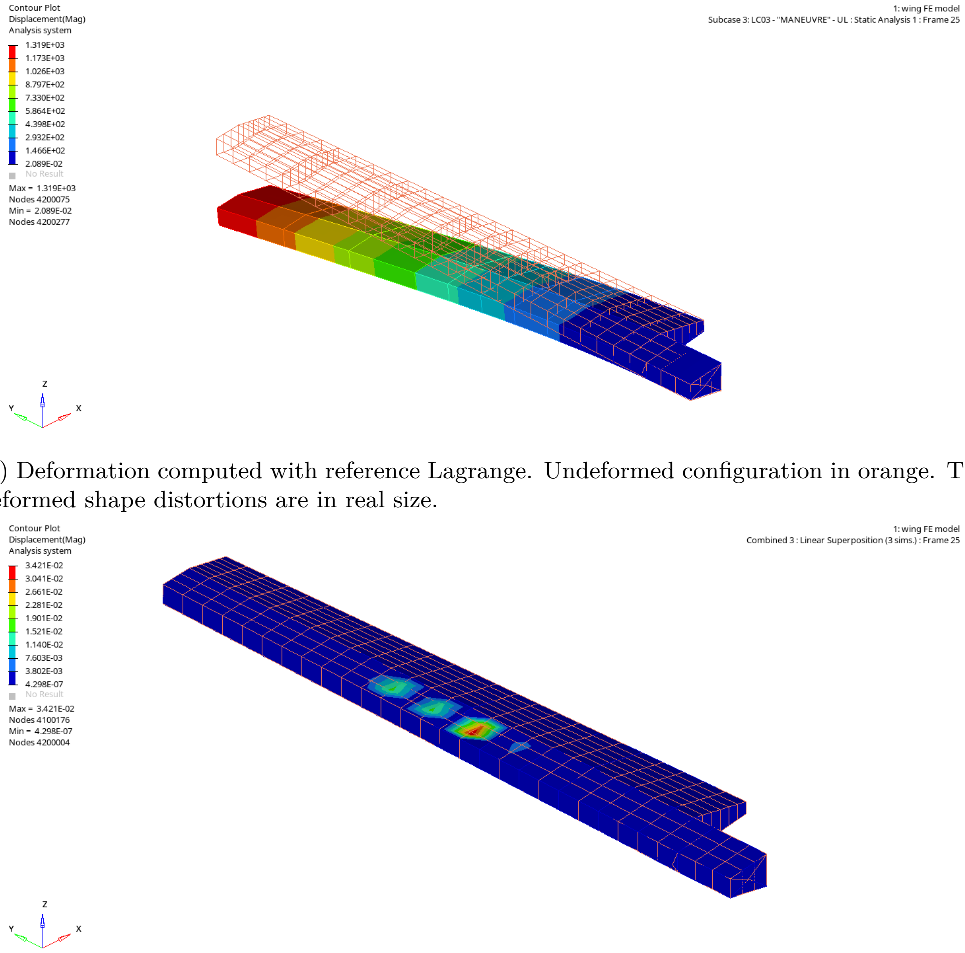
Table 7.14: Error metrics for subcase 23. The relative error has been computed only for reference displacements larger than 0.001 mm.



(a) Deformation computed with reference Lagrange. Undeformed configuration in orange. The deformed shape distortions are in real size.

(b) Difference in deformation (absolute error) obtained with gl-lagrange. The deformed shape distortions are scaled by a factor of 1000.

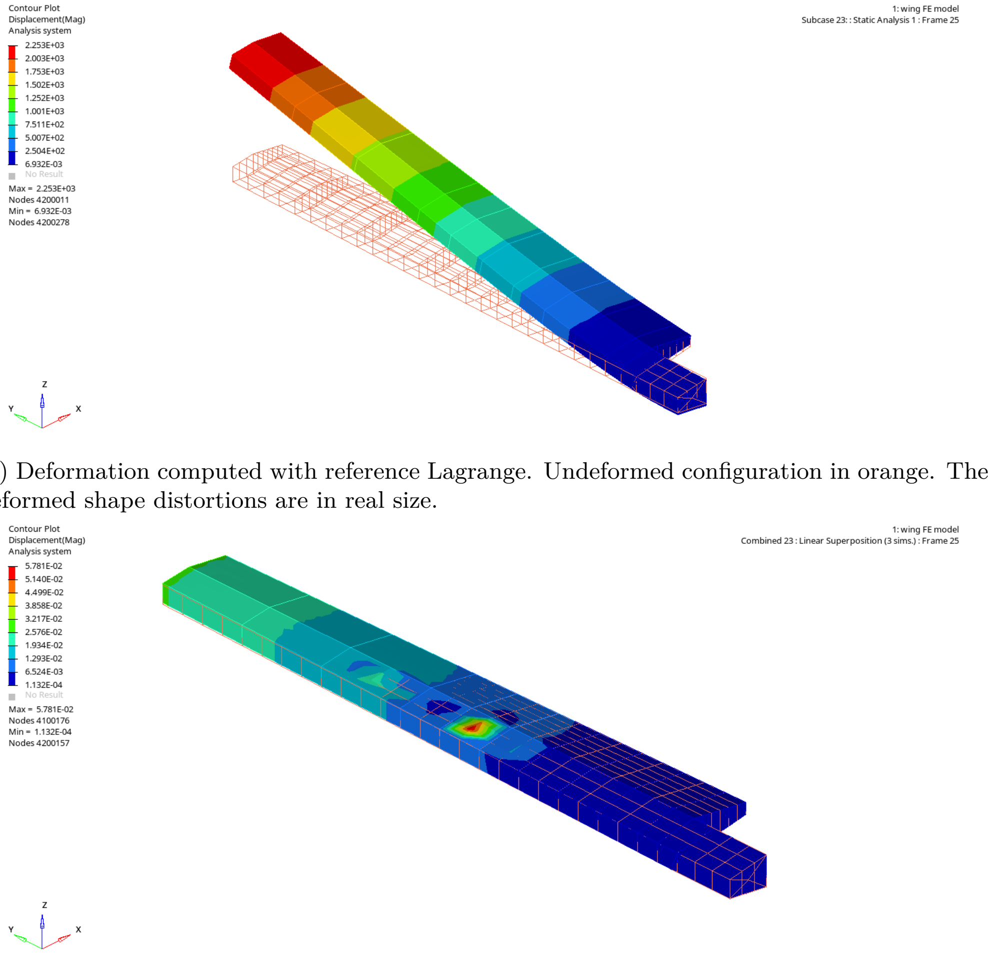
Figure 7.22: Results of the static analysis (subcase 2).



(a) Deformation computed with reference Lagrange. Undeformed configuration in orange. The deformed shape distortions are in real size.

(b) Difference in deformation (absolute error) obtained with gl-lagrange. The deformed shape distortions are scaled by a factor of 1000.

Figure 7.23: Results of the static analysis (subcase 3).



(a) Deformation computed with reference Lagrange. Undeformed configuration in orange. The deformed shape distortions are in real size.

(b) Difference in deformation (absolute error) obtained with gl-lagrange. The deformed shape distortions are scaled by a factor of 1000.

Figure 7.24: Results of the aeroelastic analysis (subcase 23).

7.3 Verification of sensitivity and optimisation

In this section all results are obtained with the “wingbox coarse” model.

7.3.1 Comparison of optimisation histories

A quick way to verify the global-local procedure is to compare a reference and a global-local optimisation, by looking, for example, at the histories of objective and maximum constraint violation as in Fig. 7.25.

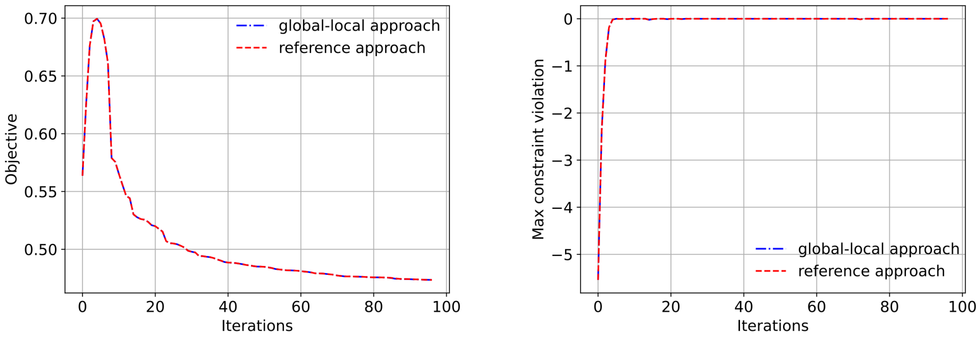


Figure 7.25: Optimisation histories.

In this case both procedures converge after 97 iterations, following almost identical optimisation histories. The final design are similar as shown in Fig. 7.26, because, as can be seen from the plot, the design variables, represented in blue points for reference and red points for the proposed global-local MDO solution, generally overlap with negligible deviation.

To quantify the difference, Table 7.15 reports in detail the relevant metrics for the absolute and relative error: each design variable is off by at most 1.304 % and on average by $7.523 \cdot 10^{-2}$ %.

	e_{abs}	e_{rel}
$\| \cdot \|_\infty$	0.011	0.013
$\bar{\cdot}$	0.001	0.001

Table 7.15: Error metrics for the optimal design. Global-local solution vs. reference Lagrange solution.

The optimisation of a reference model and its equivalent decomposition should yield the same results, if the same optimisation settings are used. In practice, this is not always the case: it may happen that the optimisation histories start to diverge and lead to different results. Justifiable phenomena that may lead the optimiser astray are:

1. different ordering of the constraints and their gradients,

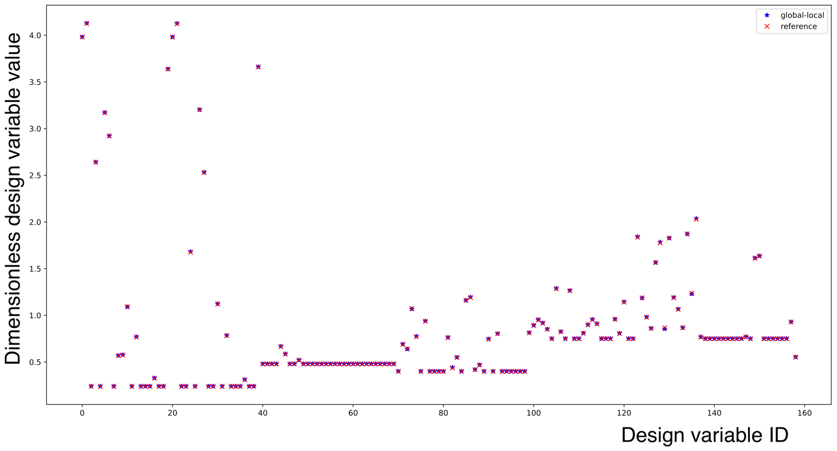


Figure 7.26: Comparison of optimal design variables.

2. minor differences in the gradients, due to the different numerical errors, arising from the different numerical procedures.

These may cause the optimiser to choose a slightly different optimisation path. Once this happens, the sensitivities are computed on slightly different designs and are not comparable one to one anymore. If the problem is not locally convex this will result in two different optimisation histories, converging to separate local minima.

For these reasons, while comparing the optimisation history and results is a quick way to test the implementation, it is not a proper verification procedure. A better alternative is to directly verify the sensitivities.

7.3.2 Verification of the entries of the sensitivity analysis

At any point in the design space an equivalent global-local decomposition should yield the same values of f, $\dfrac{df}{dx}$, g and $\dfrac{dg}{dx}$ as computed by standard Lagrange. Therefore, the correct computation of the sensitivities can be verified by sampling the design space at multiple random locations x and comparing the values of f, g and their derivatives.

The values of f, g and $\dfrac{df}{dx}$ are, in both procedures, computed by Lagrange internally, the global-local procedure only assembles the values obtained through the Python-API. Thus, the results are identical (same float representation). In the following, the focus will be on the analysis of the errors in the computation of $\dfrac{dg}{dx}$.

For the purpose of verification one might compare the entries of $\dfrac{dg}{dx}$ one by one. The corresponding error metrics are reported in Table 7.16. The reported high values of relative error are caused by entries with a negligible absolute error but an associated

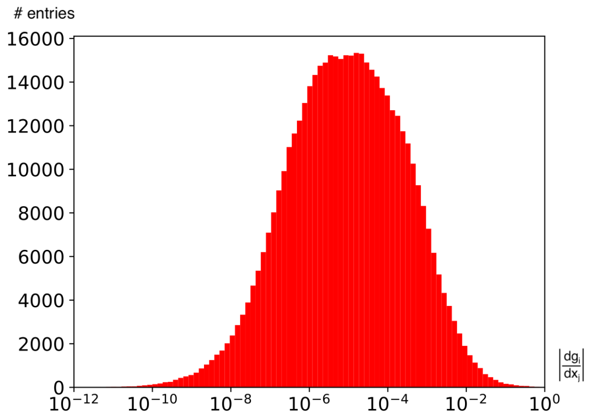


Figure 7.27: Distribution of absolute values of $\dfrac{dg}{dx}$ entries.

large relative error.

	e_{abs}	e_{rel}
$\lVert\cdot\rVert_\infty$	$4.599 \cdot 10^{-4}$	$52\,676.224$
$\bar{\cdot}$	$1.550 \cdot 10^{-7}$	0.172

Table 7.16: Error metrics for the sensitivities of the constraints (sample 0).

Figure 7.27 shows the distribution of the magnitude of all entries of $\dfrac{dg}{dx}$: most entries are smaller than $1.0 \cdot 10^{-4}$.

The relative error linked to these smaller values is generally high as shown in Fig. 7.28, whereas the computation of larger entries is generally more accurate. The vast number of small entries with an associated high relative error is what skews the metric. A threshold on the entries magnitude below which the relative error is not computed would generally be used to counter this effect, but in this case, large relative errors affects relevant portions of the dataset, so such a threshold does not exist. Nevertheless, Fig. 7.28 clearly shows how the values with an associated high relative error taper off as the magnitude of the entries grows.

If single a row of $\dfrac{dg}{dx}$ is considered, which represents the derivative of a single constraint with respect to all design variables, a pattern similar to that of the whole $\dfrac{dg}{dx}$ emerges, as shown in Fig. 7.29. For each constraint the entries cover multiple orders of magnitude and only a few are larger than $1.0 \cdot 10^{-4}$. This is to be expected and means that for each constraint most design variables have a negligible influence. This is important because it means that, in spite of the large relative error associated to most entries, the quality of the information obtained from the computed dataset might still be good.

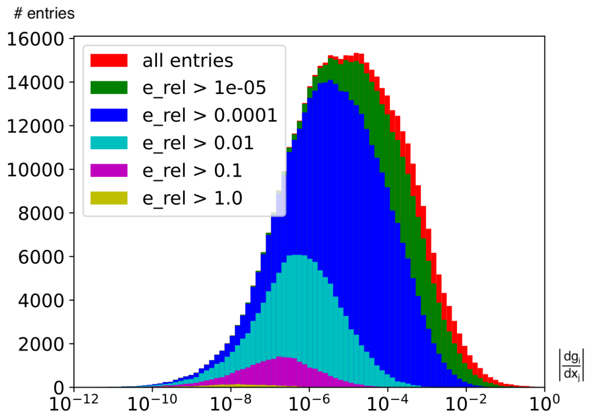


Figure 7.28: Distribution of absolute values of $\dfrac{dg}{dx}$ entries. Values are highlighted based on their corresponding relative error

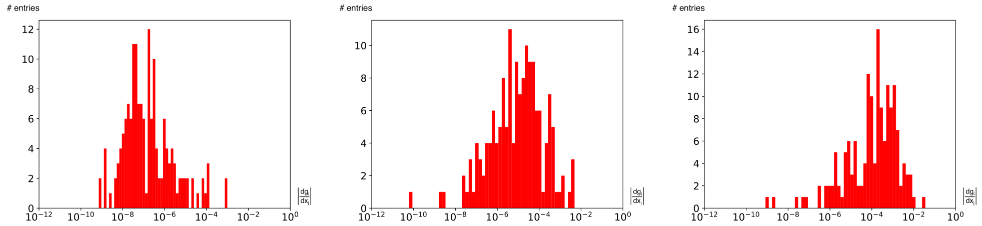


Figure 7.29: Distribution of absolute values of $\dfrac{dg}{dx}$ entries, for 3 randomly chosen rows.

7.3.3 Verification of the search directions implied by the sensitivity analysis

To evaluate the quality of the information provided to the optimiser, it is possible to examine the $\dfrac{dg}{dx}$ in terms of search directions.
Each row $\dfrac{dg}{dx}$ contains the components of a vector in the design space, which points towards a larger violation of the associated constraint. It is natural therefore to compare the reference vectors and the computed global-local vectors. Let $\mathbf{d}_i$ be the vector:

$$\mathbf{d}_i = \left(\frac{dg_i}{dx}\right)^W - \left(\frac{dg_i}{dx}\right)^{GL} \tag{7.9}$$

as represented in Fig. 7.30.

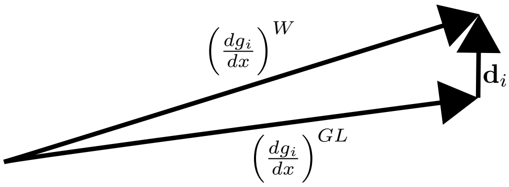

$\left(\frac{dg_i}{dx}\right)^W$
$\mathbf{d}_i$
$\left(\frac{dg_i}{dx}\right)^{GL}$

Figure 7.30: Graphical representation of $\dfrac{dg_i}{dx}$ interpreted as a search direction.

By computing the norm of $\mathbf{d}_i$ and comparing it to the norm of $\mathbf{ref}_i = \left(\dfrac{dg_i}{dx}\right)^W$ for each constraint, the influence of the smallest entries, which are also the most inaccurate, is significantly reduced. Table 7.17 reports the average and maximum norm of the reference search directions, of the associated $\mathbf{d}_i$ and the ratio between the two. The average norm of $\mathbf{d}_i$ is significantly smaller than the norm of $\left(\dfrac{dg_i}{dx}\right)^W$ and the ratio $\dfrac{\|\mathbf{d}_i\|_2}{\|\mathbf{ref}_i\|_2}$ is mostly negligible. Thus, the information in terms of search directions computed by the global-local procedure is generally accurate, which explains how the optimisation results described in section 7.3.1 could be obtained.

	$\|\mathbf{ref}_i\|_2$	$\|\mathbf{d}_i\|_2$	$\dfrac{\|\mathbf{d}_i\|_2}{\|\mathbf{ref}_i\|_2}$
$\bar{\cdot}$	0.064	$5.150 \cdot 10^{-6}$	$5.718 \cdot 10^{-4}$
$\|\cdot\|_\infty$	2.522	$8.923 \cdot 10^{-4}$	0.046

Table 7.17: Error metrics for the sensitivities of the constraints (sample 0). The ratio $\dfrac{\|\mathbf{d}_i\|_2}{\|\mathbf{ref}_i\|_2}$ was computed only when $\|\mathbf{ref}_i\|_2 > 1.000 \cdot 10^{-6}$.

7.3.4 Verification of the computed subblocks of the sensitivity analysis

The matrix $\dfrac{dg}{dx}$ has a size of 2980.000 × 159.000. The number of rows results from considering 745 constraints for 4 different subcases. Of these constraints, 739 are global and 6 are local. The design variables are divided in 158 global design variables and 1 local design variable. This means that most entries in $\dfrac{dg}{dx}$ belong to the $\dfrac{dg^G}{dx^G}$ subblock, which is computed internally by Lagrange. The portions of $\dfrac{dg}{dx}$ actually computed by the global-local procedure are only thin slices of the whole matrix.

Thus, while the fact that local constraints and design variables are significantly fewer is representative of a real industrial problem, to verify the implementation it is necessary to ensure that the accuracy reported by the error metrics is not a consequence of a negligible influence of the parts computed by the global-local procedure. In other words, one must show that the large number of $\dfrac{dg^G}{dx^G}$ entries is not concealing errors in the remaining global-local sensitivities.

To this end, it is possible to consider the $\frac{dg^G}{dx^L}$, $\frac{dg^L}{dx^G}$, $\frac{dg^L}{dx^L}$ subblocks separately.

	$\lVert\mathbf{ref}_i\rVert_2$	$\lVert\mathbf{d}_i\rVert_2$	$\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$
$\bar{\cdot}$	$1.634 \cdot 10^{-4}$	$5.581 \cdot 10^{-7}$	0.014
$\lVert\cdot\rVert_\infty$	0.012	$3.086 \cdot 10^{-4}$	2.181

Table 7.18: Error metrics for $\frac{dg^G}{dx^L}$ (sample 0). The ratio $\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$ was computed only when $\lVert\mathbf{ref}_i\rVert_2 > 1.000 \cdot 10^{-6}$.

	$\lVert\mathbf{ref}_i\rVert_2$	$\lVert\mathbf{d}_i\rVert_2$	$\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$
$\bar{\cdot}$	0.008	$1.685 \cdot 10^{-5}$	0.002
$\lVert\cdot\rVert_\infty$	0.018	$3.423 \cdot 10^{-5}$	0.004

Table 7.19: Error metrics for $\frac{dg^L}{dx^G}$ (sample 0). The ratio $\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$ was computed only when $\lVert\mathbf{ref}_i\rVert_2 > 1.000 \cdot 10^{-6}$.

	$\lVert\mathbf{ref}_i\rVert_2$	$\lVert\mathbf{d}_i\rVert_2$	$\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$
$\bar{\cdot}$	0.005	$6.076 \cdot 10^{-6}$	0.001
$\lVert\cdot\rVert_\infty$	0.009	$1.546 \cdot 10^{-5}$	0.003

Table 7.20: Error metrics for $\frac{dg^L}{dx^L}$ (sample 0). The ratio $\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$ was computed only when $\lVert\mathbf{ref}_i\rVert_2 > 1.000 \cdot 10^{-6}$.

Tables 7.18, 7.19, 7.20 report the corresponding error metrics. All metrics show negligible errors except for the $\frac{\lVert\mathbf{d}_i\rVert_2}{\lVert\mathbf{ref}_i\rVert_2}$ metrics of Table 7.18. For this particular case, since there is only one local variable, measuring the ratio of the norms is equivalent to measuring the relative error, which can be big for small values. The error metrics for $\lVert\mathbf{d}_i\rVert_2$ show that the overall accuracy is consistent with the one measured elsewhere.

7.4 Summary

It has been demonstrated that for each global-local problem it is possible to define an equivalent single-model problem, by essentially merging the FE-meshes. Furthermore, regarding the verification of the analysis it has been proven that condensation is theoretically exact and that errors arise only from round-off errors. Due to this source of numerical error, as the number of degrees of freedom condensed increases the accuracy of the solutions diminishes, but the overall accuracy remains acceptable. Regarding the accuracy of the sensitivities, the correct implementation has been verified. For large problems, common metrics of errors appear to prove the poor quality of the results, but further analysis has proven the accuracy of the computed sensitivity and identified what causes the error metrics to be misleading.

Chapter 8

Case studies

This chapter presents two case studies. The optimisation of an academic demonstrator aircraft, named *OptiMALE*, investigates the limitation of the current MDO procedure. A second example, in which a wingbox is optimised, compares the performance of the novel approach with the reference one.

8.1 Failure of the standard procedure to produce a locally feasible optimum: OptiMALE

8.1.1 MDO of an Aircraft with a Non-Regular Area

As an example of the problems that might arise without appropriately treating local areas, MDO is applied to the model represented in Figure 8.1. As shown in Figure 8.2, the right wing of the model presents a non-regular area, highlighted in yellow (Figure 8.2c). The part represents a cut-out in the lower-skin of the wing. Cut-outs like this are typically found on the lower skin and are known in the industry as manholes or access panels.

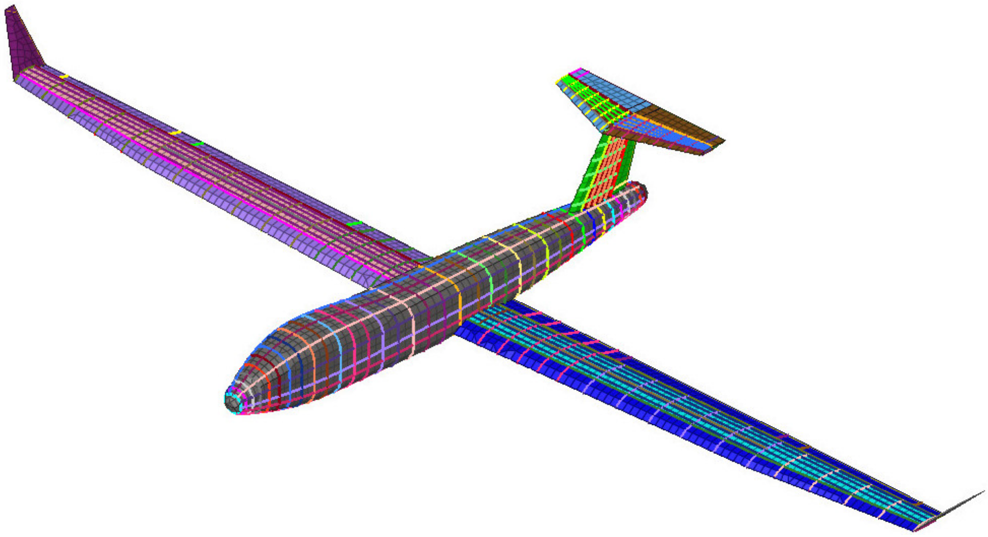
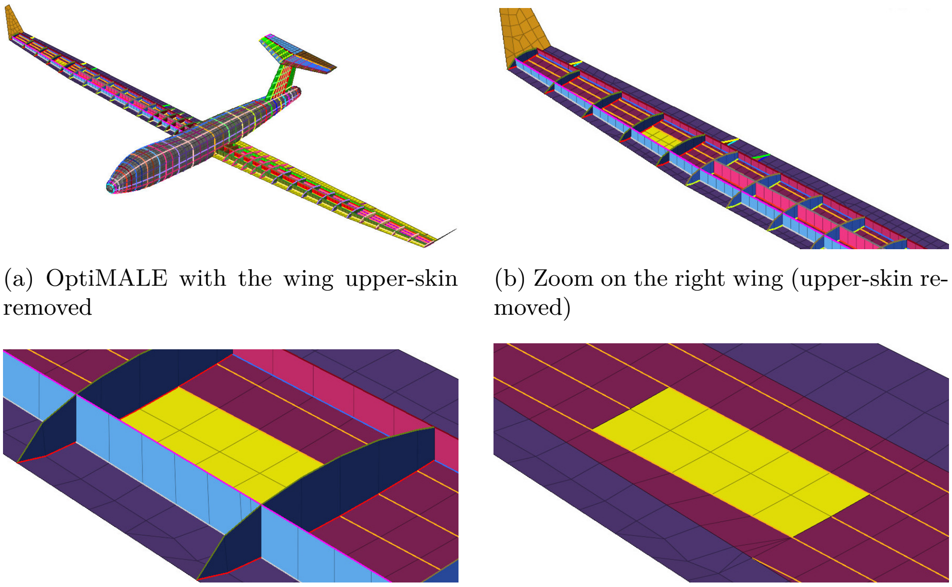


Figure 8.1: Finite element model of OptiMALE.



(a) OptiMALE with the wing upper-skin removed

(b) Zoom on the right wing (upper-skin removed)

(c) Zoom on the non-regular area (in yellow)

(d) Zoom on the non-regular area with surrounding spars and ribs removed

Figure 8.2: Non-regular area (in yellow) as represented in the aircraft model.

8.1.2 Modelling of the Non-Regular Area

In the global model, the non-regular area shown in Figure 8.2d is represented following a smeared stiffness approach, which greatly reduces the number of elements, but does not permit exactly representing the stiffness of the part, nor accurately capturing the internal displacement field. When separately modelled in detail, the part appears as in Figure 8.3.

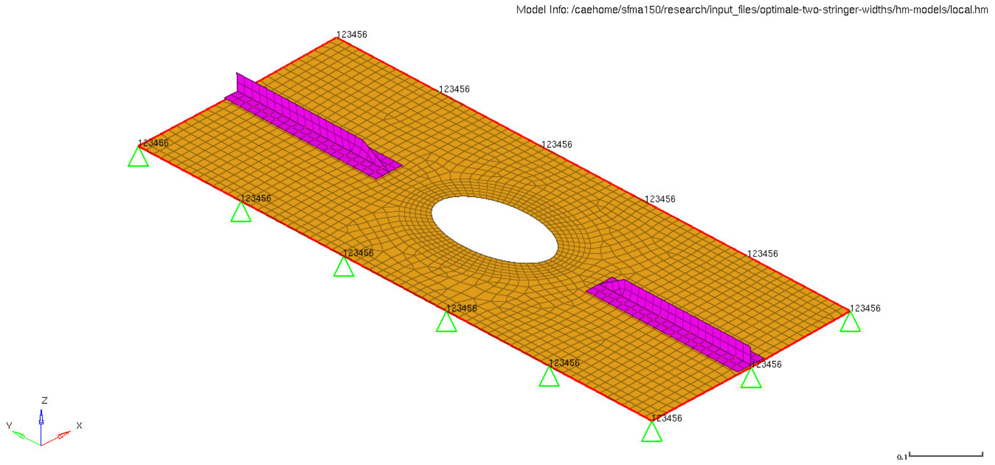

Model Info: /caehome/sfma150/research/input_files/optimale-two-stringer-widths/hm-models/local.hm
123456
123456
123456
123456
123456
123456
123456
123456
123456
123456
123456
123456
123456
Z
Y
X
0.1

Figure 8.3: Detailed representation of the non-regular area.

8.1.3 Optimisation and Subsequent Detailed Analysis

Using the simplified modelling of the part, the aeroplane is optimised, keeping the design of the non-regular area fixed without applying any constraint to it. After the MDO procedure is concluded and the design has been optimised, the non-regular area is separately designed in detail and analysed by enforcing at the boundaries the displacements from the aircraft model.

8.1.4 Constraint Violation

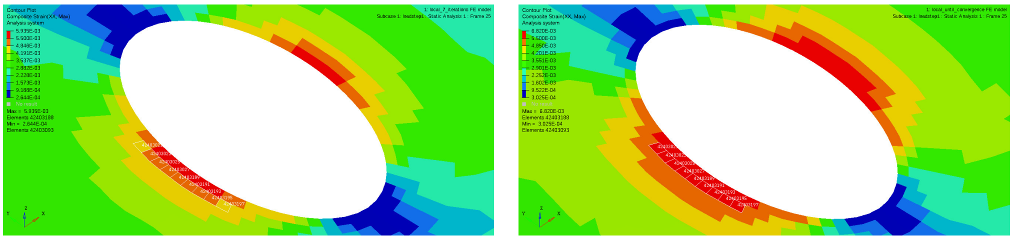
As the detailed model of the non-regular area is created and the part is analysed, a violation of the local axial strain constraints is revealed.

Figure 8.4 shows the maximum composite strain at the beginning of the optimisation, i.e. after 7 iterations (Figure 8.4a), and at the end of the optimisation, i.e. after 331 iterations (Figure 8.4b).

If the strain is measured close to the hole but avoids the elements directly linked to the edge where the strain concentration occurs, after 7 iterations, all measures are below the allowable value of 5.5×10^{-3}. This can be seen in Figure 8.4a and also from the data in Table 8.1. This means that the local area was initially sized correctly to sustain the loads at the beginning of the optimisation.

Nevertheless, as the optimisation reaches the optimum, the local values of strain exceed the allowable value. Table 8.1 reports the Reserve factors (RFs) after 7 iterations and at convergence for the elements highlighted in Figure 8.4 and highlights the violated constraints:

$$\mathrm{RF} = \frac{\text{allowable strain}}{\text{actual strain}}$$



(a) After 7 iterations

(b) At convergence

Figure 8.4: Violation of strain constraints in the local model.

Table 8.1: Composite axial strain.

Element ID		At 7 Iterations		At Convergence		Drop in RF
	Allowable	Actual	RF	Actual	RF	
42403021	5.5×10^{-3}	4.830×10^{-3}	1.14	5.564×10^{-3}	0.99	−0.15
42403023	5.5×10^{-3}	5.002×10^{-3}	1.10	5.757×10^{-3}	0.96	−0.14
42403025	5.5×10^{-3}	5.126×10^{-3}	1.07	5.898×10^{-3}	0.93	−0.14
42403027	5.5×10^{-3}	5.221×10^{-3}	1.05	6.003×10^{-3}	0.92	−0.14
42403189	5.5×10^{-3}	5.247×10^{-3}	1.05	6.030×10^{-3}	0.91	−0.14
42403191	5.5×10^{-3}	5.218×10^{-3}	1.05	5.993×10^{-3}	0.92	−0.14
42403193	5.5×10^{-3}	5.154×10^{-3}	1.07	5.916×10^{-3}	0.93	−0.14
42403195	5.5×10^{-3}	5.031×10^{-3}	1.09	5.772×10^{-3}	0.95	−0.14
42403197	5.5×10^{-3}	4.835×10^{-3}	1.14	5.545×10^{-3}	0.99	−0.15

8.1.5 On the Importance of Considering Local Sizing during Global MDO

This example shows that excluding non-regular areas from the optimisation and leaving them unconstrained can lead to a significant and unpredictable drop of the reserve factors, even if the local area was initially correctly sized.

This problem may be solved by sizing the local part, but modifying the local stiffness and mass can dramatically alter the loadpaths of the entire structure and potentially invalidate the design obtained by MDO. This is typical of non-regular areas like the one presented, especially in the case of a sequence of similar cut-outs along the wing span. Thus, local constraint violations like these, only discovered at a later stage, may force the designers to repeat a multidisciplinary optimisation and therefore result in a costly setback.

8.2 A successful application of the global-local MDO procedure: wingbox

8.2.1 Model and optimisation problem

In the following, a metallic wingbox is considered, with an inspection hole in the upper skin, as an example of a ‘non-regular area”.

Four different subcases are considered: three static analyses named ‘gust-up”, ‘landing” and ‘manoeuvre” and static aeroelastic analysis representing a steady flight at Mach 0.3 with a dynamic pressure of 2.800 kPa and an angle of attack of 5.000°.

The structure must be sized for minimal weight and is subject to strength constraints defined as $\sigma_{actual} < \sigma_{allowable}$. With the following definition of RF:

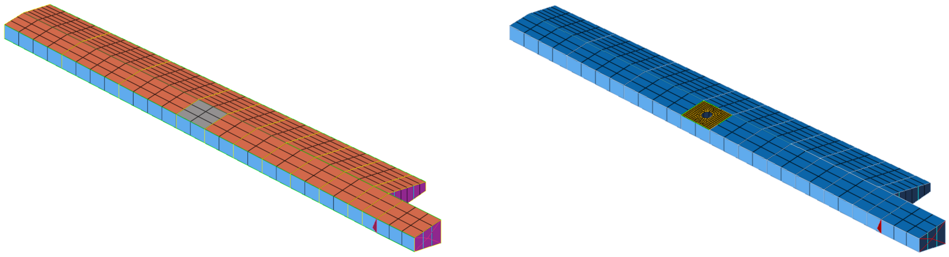
$$RF = \frac{\sigma_{allowable}}{\sigma_{actual}} \tag{8.1}$$

and the following definition of constraints:

$$g = 1 - \frac{1}{RF} = 1 - \frac{\sigma_{actual}}{\sigma_{allowable}} \tag{8.2}$$

a constraint violation is implied by $RF < 1$ or equivalently $g < 0$.

Two modelling strategies can be used to analyse the structure. One, represented in figure 8.5a, is based on a coarse model, which does not capture the geometry of the hole, but uses equivalent material properties to match the stiffness properties of the local area. The other one, represented in figure 8.5b uses a sufficiently fine model to accurately capture the deformation of the ‘non-regular area”, combined with a coarse model for the remainder of the structure.



(a) Coarse model of the wingbox.

(b) Global local model of the wingbox.

Figure 8.5: Coarse and global-local representations of the wingbox model. In the coarse model, the non-regular area is modelled with degraded material properties so that its overall stiffness is equivalent to that of the global-local representation. In the global-local representation, separate global (blue) and local (yellow) models are used. The local model presents a finer mesh and a more accurate representation of the geometry

In the following, two optimisation runs are compared. One, based on the coarse model, is performed by standard *Lagrange* and represents the current way of working. Therein the critical area (local model) is fixed and unconstrained, but sized to be initally locally feasible including a design factor of 1.3 (minimum RF = 1.3) to account for later load path changes. The other one, based on the global-local representation, follows the application of the global-local approach, which allows to define local design variables and constraints.

Figure 8.6 shows the thickness distribution of the initial design. The displacement fields computed for this design can be applied at the boundary of a refined local model to check for local constraints violations. As demonstrated by figure 8.7, which shows the value of minimum RF, the design is locally feasible: any constraints violation would be highlighted in red, dark blue represents a RF larger than 1.1 and green represents a RF larger than 1.3. The fact that the local model is entirely above this threshold proves not only that the design is feasible, but also that a conservative design factor provides a margin for changes of the surrounding structure.

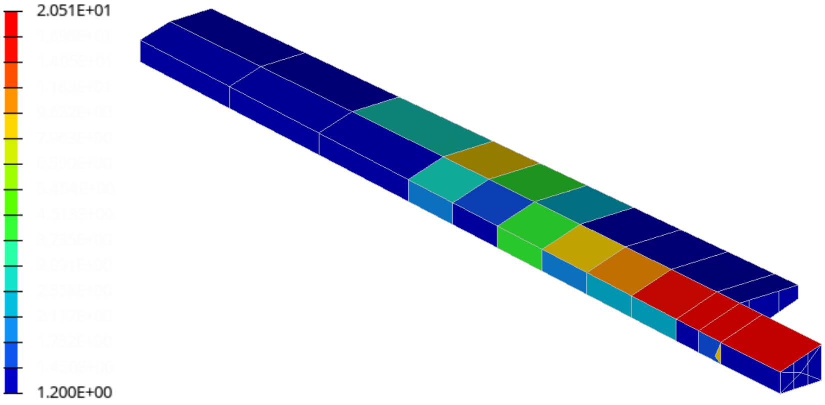


Figure 8.6: Initial thickness distribution. The scale used in the legend is logarithmic.

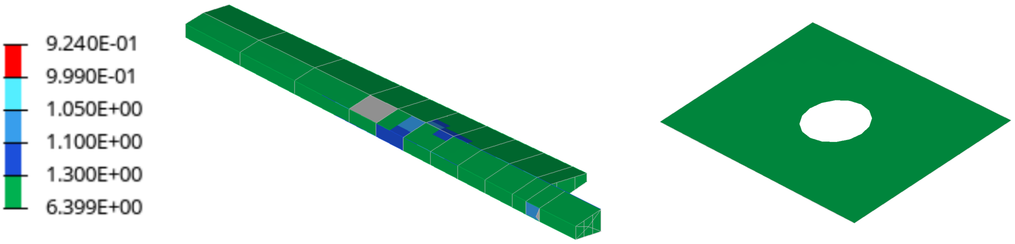


Figure 8.7: Initial minimum RF distribution. The initial design is feasible and the local design is conservative, as it can be seen from the green color which represents a RF larger than 1.3.

8.2.2 Feasibility of results

The optimal design found by the *Lagrange* optimisation procedure, while keeping the local fixed and unconstrained, as currently done during optimisation, leads to an unfeasible design as depicted in Fig. 8.8. This despite the fact that the local structure was conservatively sized with an initial reserve factor of 1.3.

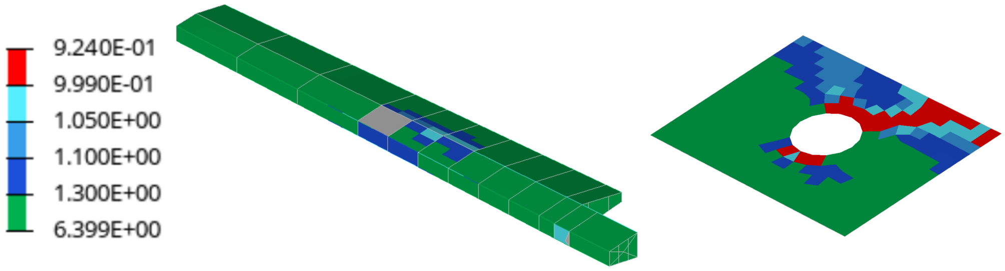


Figure 8.8: Minimum RF of *Lagrange* optimum. The global design on the left is feasible, but when the local model is analysed in detail after the optimisation the unfeasiblity, highlighted in red, is revealed.

The optimal design found with the proposed monolithic global-local optimisation procedure, using local design variables and constraints, is feasible, as shown in figure 8.9, which depicts the distribution of minimum RF.

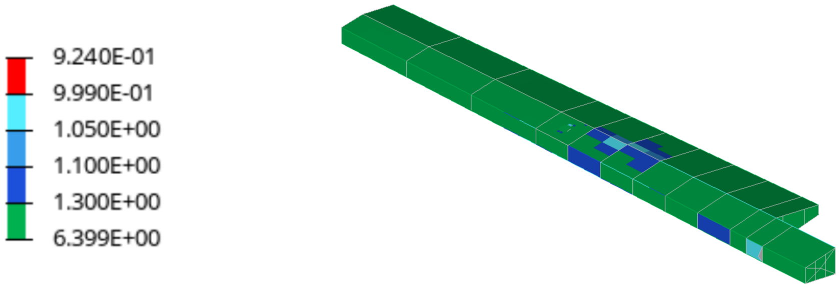


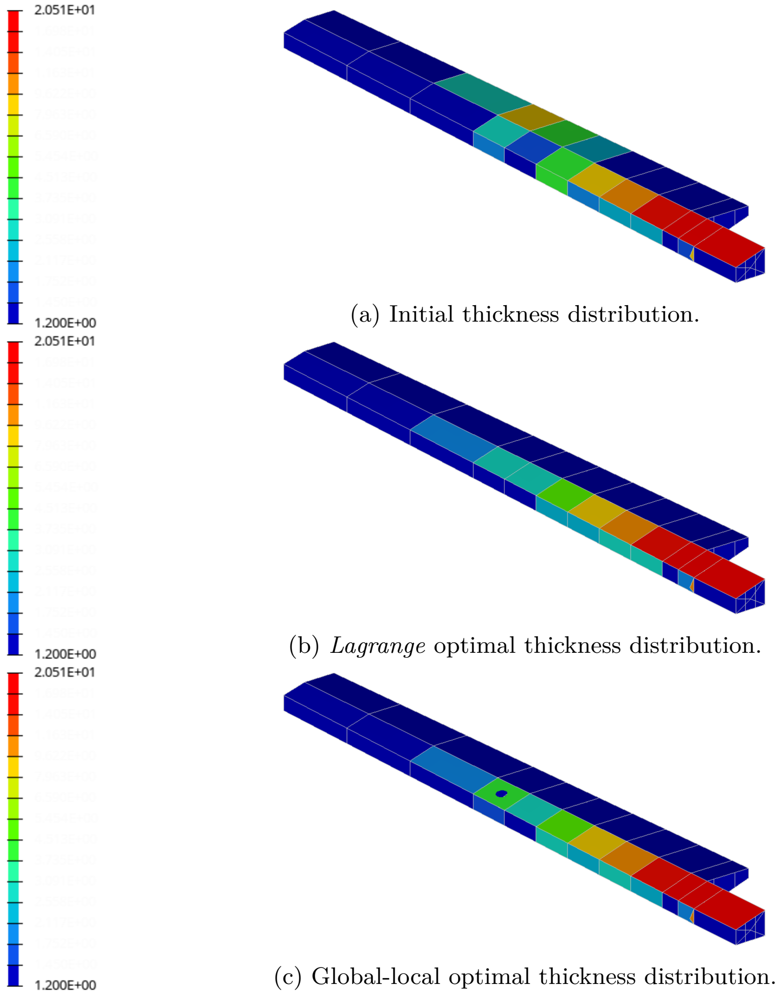
Figure 8.9: Minimum RF of global-local optimum. The design is feasible.

8.2.3 Comparison of designs

Figure 8.10 shows the initial design and the two optimal designs.

8.2.4 Comparison of histories

The optimal design found by the global-local strategy weighs 475.640 kg compared to the weight of 473.444 kg of the unfeasible design obtained by the reference approach, as shown in Fig. 8.11.



(a) Initial thickness distribution.

(b) *Lagrange* optimal thickness distribution.

(c) Global-local optimal thickness distribution.

Figure 8.10: Comparison of initial and final designs.

8.2.5 Correcting the local design

In the standard design procedure, if the optimised design was locally unfeasible, designers could attempt to resolve any constraint violation by manually modifying the design using their engineering judgement. This section demonstrates that such an approach is non-trivial, because optimal designs are generally close to the border of the infeasible region and very sensitive to design changes.

By increasing the thickness of the local area from 3.0 mm to 4.5 mm the weight of the structure becomes 475.271 kg, while the local design is still unfeasible and the unfeasibility has propagated beyond the local model, as shown in Fig. 8.12. In particular the weight of the local model increases from 3.653 kg to 5.480 kg.

A second way to mitigate the infeasible design would be to further adjust the design factor for the fixed local model. However this would require either an iterative approach by alternately optimizing and increasing the design factor, which would result in a

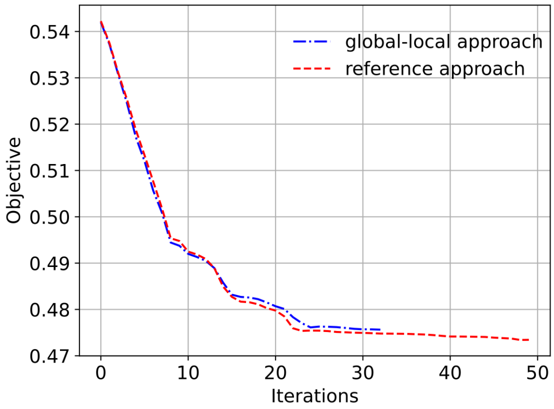


Figure 8.11: History comparison of the objective. The two optimisations follow a similar path. Coincidentally, the global-local approach satisfies the convergence criteria before the reference approach.

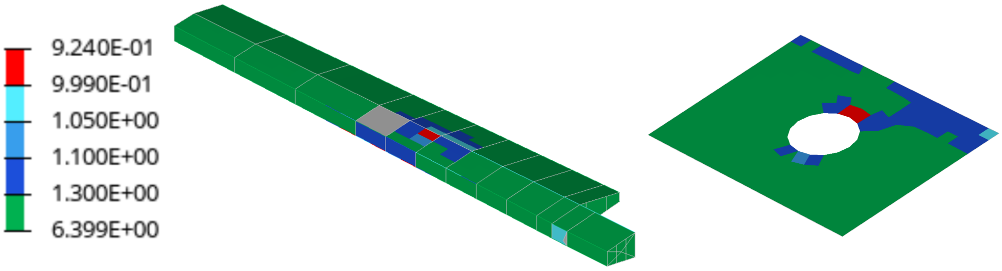


Figure 8.12: Increasing the thickness of the local model is not a viable way of correcting the design in Fig. 8.8. Before the local is sufficiently thick to become feasible, the additional loads attracted in that area cause elements in the surroundings to become unfeasible. This is often the case, since optimal designs are close to the boundaries of the feasible region in the design space.

prohibitive computational cost, or by initially choosing an even higher design factor, which would increase the mass even more and lead to a sub-optimal result. For a model with multiple local areas, an estimation for the design factor proves to be even more challeging and disadvantageous.

Hence, the example demonstrates that the standard design approach fails to find optimal solutions that satisfy all the constrains, while the proposed global-local approach finds a locally feasible optimal design, overcoming the limitations of the standard procedure.

8.3 Summary

Three case studies have been presented. The first one illustrated the shortcomings of the standard optimisation procedure, when a single local model of a manhole with stringer run-outs is applied to the model of an aircraft. The second case studies compared the standard procedure and the proposed global-local methodology. Despite the best efforts designers could make in avoiding local infeasibility the standard procedure was unable to succeed, whereas the proposed procedure produced a feasible design. Furthermore, it has been shown that correcting the infeasible design was not easy and would have lead either to considerable delay or to a suboptimal design.

Chapter 9

Conclusions

9.1 Summary and conclusions

A procedure for global-local MDO has been introduced to overcome the challenge posed by “non-regular areas”, which as described in chapter 1 and demonstrated in chapter 8 can cause the optimal design to be infeasible, or suboptimal or not accurately evaluated and compared with other designs.

The global-local MDO procedure described is based on a global-local analysis and an ad-hoc semi-analytical coupled sensitivity analysis. Furthermore, in the procedure presented, the evaluation of constraints requires only a static analysis at local level, while considering multiple disciplines at global level. This minimises the additional computational cost, while enhancing the reliability of the process and reducing the risk of major setbacks in the product development.

9.1.1 Research contributions

The literature on MDO architectures has been reviewed and presented in chapter 3, with a focus on their application to the optimisation of aircraft structures. The current use of multiple models with different levels of accuracy within an MDO procedure was evaluated. Furthermore, the available global-local analysis strategies have been reviewed, in search of techniques suitable for the evaluation of detailed models of “non-regular areas”. Static condensation and Specified Boundary Displacements (SBD) were chosen, as they are simple well established coupling techniques. It was also found, that despite the variety of global-local analysis strategies available, the global-local optimisation approaches are often based on a standard FE-analysis. Instead of integrating a global-local approach in the optimisation, the usual approach is to split the optimisation problem in two or more levels. While the use of distributed architectures is more common, a monolithic approach was deemed more appropriate for the case of “non-regular areas”. The main goal of the procedure introduced is not the detailed design of local features, but rather the enforcement of local feasibility. This is more easily achieved by considering not only the design of local part itself, but also the design of the surrounding structure and its influence on the loads injected into the local domain.

A novel global-local methodology suitable for the extension of an existing MDO procedure was developed and presented in chapter 4. The global-local analysis methodology, based on Guyan condensation and SBD, was complemented with a novel derivation of the implied sensitivity analysis formulation, based on a semi-analytic computation of the sensitivities.

The methodology was successfully implemented to extend the functionalities of *Lagrange*, by leveraging the available *Lagrange*/Python API, as described in chapter 5. The implementation of the procedure is based on the exchange of information between an instance for each FE-model and an optimiser. Furthermore some operations not already available, such as static condensation and the computation of coupled sensitivities was implemented directly in Python, leveraging as much as possible the functionalities available through the *Lagrange* interface. Despite the fact that the procedure was implemented to extend *Lagrange*, it can be implemented connecting other available software for FE-analysis and optimisation.

A study of the impact on computational cost was presented in chapter 6. The number of active constraints and design variables were identified as the main drivers of cost. Because of this, it was found that commonly adopted techniques, such as active set strategy and design variables filtering, play an important role in limiting the impact of constraints and design variables. It is recommended that particular care is taken in limiting the introduction of additional local design variables. In combination with the aforementioned techniques, the additional cost coming from the use of the presented global-local methodology is deemed limited. Compared to the cost of simply refining the model used for MDO, the presented global-local approach offers some computational advantages, mainly coming from the fact that only static analysis is considered at local level and that local models can be studied in parallel.

In chapter 7 the correct implementation was verified through a series of tests, meant to separately verify the implementation of the analysis or the computation of the sensitivities or the verification of the optimisation procedure, based on the assumption that analysis and sensitivities are correctly computed.

Lastly, in chapter 8 it was demonstrated that the reference strategy of leaving "non-regular areas" fixed and unconstrained can lead to designs with local infeasibility. It was also proved, that this is the case even if the initial local design is very conservatively pre-sized. Furthermore, it was shown that successive modifications of a locally infeasible optimal design are not a good solution, not only because they would lead to a sub-optimal design, but also because it may be hard to obtain a feasible design. On the contrary, the global-local MDO was capable of finding an optimal design, with a weight comparable to the one obtained with the reference approach, but feasible. Thus, it was proven that the developed global-local MDO procedure successfully overcomes the challenge posed by "non-regular areas" in the preliminary design of aircraft structures.

9.1.2 Limitations of the developed procedure

The developed global-local MDO procedure has not been extended to other disciplines already available through *Lagrange*, such as eigenvalue, frequency response, flutter and gust analysis. The implementation is limited to static analysis and static aeroelasticity at global level and static analysis only at local level.

The implementation relies on the *Lagrange*/Python interface and all newly developed core functionalities are coded in Python, rather than integrated within *Lagrange*.

The procedure is based on SBD for the solution of the local model. An advantage of this is that, direct enforcement of a displacement field at the boundary of a refined model is the current way to evaluate the internal stress distribution in later design stages. On the other hand, specifying boundary conditions in terms of loads instead of displacements would be preferred for the evaluation of buckling.

Thanks to the use of condensation, the procedure is suitable for treating any type of "non-regular area", regardless of its complex geometry, but it does not take advantage of the fact that many local areas are similar.

9.2 Future work

Future work to extend and improve the proposed procedure would include the following:

- The procedure should be implemented within *Lagrange*, making use of optimized algorithms and leveraging the potential for parallelisation between different local models and at algorithm level. A thorough evaluation of the computational cost should be considered. Moreover, one could explore the potential for further computational savings offered by matrix inverse update methods.

- The procedure should be extended to other disciplines like gust analysis at global level. Furthermore, the industry would be interested in the possbility to extend local analysis to FE-buckling. To this end, one could explore alternatives to SBD as a coupling formulation, such as SBSF.

As an alternative to the proposed procedure, future work could:

- investigate the potential of response surfaces and surrogate modelling techniques for cost efficient evaluation of local constraint violation. To this end one could identify a particular class of local areas and combine the parametrisation of the local area needed for analysis with the parametric generation of a detailed model for successive validation of the optimisation results.

Bibliography

[1] Robert Taylor, Jay Garcia, and Poh-Soong Tang. Using Optimization for Structural Analysis Productivity Improvement on the F-35 Lightning II. In *48th 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, April 2007.

[2] Robert Taylor, Jason Thomas, Nicholas Mackaron, Shawn Riley, and Mark Lajczok. Detail Part Optimization on the F-35 Joint Strike Fighter. In *47th 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, May 2006.

[3] Robert M Taylor. THE ROLE OF OPTIMIZATION IN COMPONENT STRUCTURAL DESIGN: APPLICATION TO THE F-35 JOINT STRIKE FIGHTER. page 11.

[4] David G. Ullman. *The mechanical design process.* McGraw-Hill series in mechanical engineering. McGraw-Hill Higher Education, Boston, 4th ed edition, 2010. OCLC: ocn244060468.

[5] Gerd Schuhmacher, Fernaß Daoud, Ögmundur Petersson, Markus Wagner, Rechliner Straße, Gerd Schuhmacher, and Oegmundur Petersson. MULTIDISCIPLINARY AIRFRAME DESIGN OPTIMISATION. page 13.

[6] Daniel Raymer. *Aircraft Design: A Conceptual Approach, Sixth Edition.* American Institute of Aeronautics and Astronautics, Inc., Washington, DC, September 2018.

[7] Stéphane Grihon. Structure Sizing Optimization Capabilities at AIRBUS. June 2017.

[8] Lucian Iorga, Vincent Malmedy, Olivia Stodieck, Joe Loxham, and Simon Coggon.

Preliminary Sizing Optimisation of Aircraft Structures - Industrial Challenges and Practices. October 2018.

[9] Fernass Daoud and Mircea Calomfirescu. Optimization of composite aircraft structures in consideration of postbuckling behavior. *Int. J. Str. Stab. Dyn.*, 10(04):905–916, October 2010. Publisher: World Scientific Publishing Co.

[10] J. Barthelemy Sobieszczanski-Sobieski. Aerospace engineering design by systematic decomposition and multilevel optimization. Toulouse, France, June 1984.

[11] Joaquim R. R. A. Martins and Andrew B. Lambe. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal*, 51(9):2049–2075, July 2013.

[12] Evin J. Cramer, Paul D. Frank, Gregory R. Shubin, John E. Dennis, and Robert Michael Lewis. On Alternative Problem Formulations for Multidisciplinary Design Optimization. 1992.

[13] E. Cramer, Jr. Dennis, J., P. Frank, R. Lewis, and G. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM J. Optim.*, 4(4):754–776, November 1994.

[14] R. T. Haftka. Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103, July 1985.

[15] J. Sobieszczanski-Sobieski, B. B. James, and A. R. Dovi. Structural optimization by multilevel decomposition. *AIAA Journal*, 23(11):1775–1782, 1985.

[16] Jaroslaw Sobieszczanski-Sobieski, Benjamin B. James, and Michael F. Riley. Structural sizing by generalized, multilevel optimization. *AIAA Journal*, 25(1):139–145, 1987. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/3.9593.

[17] Jaroslaw Sobieszczanski-Sobieski. Optimization by decomposition: A step from hierarchic to non-hierarchic systems. Hampton, VA, United States, September 1988.

[18] C. L. BLOEBAUM, P. HAJELA, and J. SOBIESZCZANSKI-SOBIESKI. Non-Hierarchic System Decomposition in Structural Optimization. *Engineering Optimization*, 19(3):171–186, May 1992.

[19] Jaroslaw Sobieszczanski-Sobieski. Sensitivity analysis and multidisciplinary optimization for aircraft design - Recent advances and results. *Journal of Aircraft*, 27(12):993–1001, December 1990.

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

[21] R. T. Haftka, J. Sobieszczanski-Sobieski, and S. L. Padula. On options for interdisciplinary analysis and design optimization. *Structural Optimization*, 4(2):65–74, June 1992.

[22] Jaroslaw Sobieszczanski-Sobieski, Jeremy Agte, and Jr. Robert Sandusky. Bi-level integrated system synthesis (BLISS). In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, 1998.

[23] R D Braun and I M Kroo. Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment. page 19, 1995.

[24] Robert David Braun. *Collaborative Optimization: An Architecture for Large-scale Distributed Design*. PhD Thesis, Stanford University, Stanford, CA, USA, 1996.

[25] B. D. Roth and I. M. Kroo. Enhanced Collaborative Optimization: Application to an Analytic Test Problem and Aircraft Design. 2008.

[26] Brian D. Roth and Ilan M. Kroo. Enhanced Collaborative Optimization: A Decomposition-Based Method for Multidisciplinary Design. pages 927–936. American Society of Mechanical Engineers Digital Collection, July 2009.

[27] Raphael T. Haftka and Layne T. Watson. Multidisciplinary Design Optimization with Quasiseparable Subsystems. *Optimization and Engineering*, 6(1):9–20, March 2005.

[28] Raphael T. Haftka and Layne T. Watson. Decomposition theory for multidisciplinary design optimization problems with mixed integer quasiseparable subsystems. *Optimization and Engineering*, 7(2):135–149, June 2006.

[29] Nestor Michelena, Harrison Kim, and Panos Papalambros. A System Partitioning And Optimization Approach To Target Cascading. July 2000.

[30] Hyung Min Kim, Nestor F. Michelena, Panos Y. Papalambros, and Tao Jiang. Target Cascading in Optimal System Design. *Journal of Mechanical Design*, 125(3):474–480, September 2003.

[31] Nestor Michelena, Hyungju Park, and Panos Y. Papalambros. Convergence Properties of Analytical Target Cascading. *AIAA Journal*, 41(5):897–905, 2003.

[32] Harrison Kim, M Kokkolaras, Loucas Louca, G.J. Delagrammatikas, N.F. Michelena, Z.S. Filipi, Panos Papalambros, and Dennis Assanis. Target Cascading in Vehicle Redesign: A Class VI Truck Study. *International Journal of Vehicle Design*, 29:199–225, January 2002.

[33] Hyung Min Kim, D. Geoff Rideout, Panos Y. Papalambros, and Jeffrey L. Stein. Analytical Target Cascading in Automotive Vehicle Design. *Journal of Mechanical Design*, 125(3):481–489, September 2003.

[34] S. Tosserams, L. F. P. Etman, P. Y. Papalambros, and J. E. Rooda. An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Struct Multidisc Optim*, 31(3):176–189, March 2006.

[35] S. Tosserams, L. F. P. Etman, and J. E. Rooda. An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Struct Multidisc Optim*, 34(3):211–227, September 2007.

[36] S. Tosserams, L. F. P. Etman, and J. E. Rooda. Augmented Lagrangian coordination for distributed optimal design in MDO. *International Journal for Numerical Methods in Engineering*, 73(13):1885–1910, 2008.

[37] R. Sellar, S. Batill, and J. Renaud. Response surface based, concurrent subspace optimization for multidisciplinary system design. In *34th Aerospace Sciences Meeting and Exhibit*, Reno,NV,U.S.A., January 1996. American Institute of Aeronautics and Astronautics.

[38] Srinivas Kodiyalam and Jaroslaw Sobieszczanski-Sobieski. Bilevel Integrated System Synthesis with Response Surfaces. *AIAA Journal*, 38(8):1479–1485, August 2000. Publisher: American Institute of Aeronautics and Astronautics.

[39] I. P. Sobieski and I. M. Kroo. Collaborative Optimization Using Response Surface Estimation. *AIAA Journal*, 38(10):1931–1938, October 2000.

[40] B. Liu, R.T. Haftka, and L.T. Watson. Global-local structural optimization using response surfaces of local optimization margins. *Struct Multidisc Optim*, 27(5):352–359, July 2004.

[41] Charlie Vanaret and Francois Gallard. On the Consequences of the "No Free Lunch" Theorem for Optimization on the Choice of MDO Architecture. page 29, June 2017.

[42] Nathan Tedford and Joaquim R. R. A. Martins. On the Common Structure of MDO Problems: A Comparison of Architectures. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2006-7080.

[43] Joaquim Martins, Christopher Marriage, and Nathan Tedford. pyMDO: An Object-Oriented Framework for Multidisciplinary Design Optimization. *ACM Trans. Math. Softw.*, 36, January 2009.

[44] Nathan P. Tedford and Joaquim R. R. A. Martins. Benchmarking multidisciplinary design optimization algorithms. *Optim Eng*, 11(1):159–183, February 2010.

[45] Justin Gray, Kenneth Moore, and Bret Naylor. OpenMDAO: An Open Source Framework for Multidisciplinary Analysis and Optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, Texas, September 2010. American Institute of Aeronautics and Astronautics.

[46] Justin S. Gray, John T. Hwang, Joaquim R. R. A. Martins, Kenneth T. Moore, and Bret A. Naylor. OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization. *Struct Multidisc Optim*, 59(4):1075–1104, April 2019.

[47] Franpa cois Gallard, Charlie Vanaret, Damien Guénot, Vincent Gachelin, Rémi Lafage, Benoît Pauwels, Pierre-Jean Barjhoux, and Anne Gazaix. GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation. January 2018.

[48] Franpa cois Gallard, Pierre-Jean Barjhoux, R. Olivanti, and Anne Gazaix. GEMS, a Generic Engine for MDO Scenarios : Key Features in Application. June 2019.

[49] Pier Davide Ciampa, Bjoern Nagel, and Michel Tooren. Global Local Structural Optimization of Transportation Aircraft Wings. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2010.

[50] August T. Noevere and Alan W. Wilhite. Bi-Level Optimization of a Conceptual Metallic Wing Box with Stiffness Constraints. In *57th AIAA/ASCE/AHS/ASC*

Structures, Structural Dynamics, and Materials Conference. American Institute of Aeronautics and Astronautics, 2016.

[51] Davide Locatelli, Ali Yeilaghi Tamijani, Sameer B. Mulani, Qiang Liu, and Rakesh K. Kapania. Multidisciplinary Optimization of Supersonic Wing Structures Using Curvilinear Spars and Ribs (SpaRibs). In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2013. _-eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2013-1931.

[52] Qiang Liu, Sameer B. Mulani, and Rakesh K. Kapania. Global/Local Multidisciplinary Design Optimization of Subsonic Wing. In *10th AIAA Multidisciplinary Design Optimization Conference*, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, January 2014.

[53] Qiang Liu, Mohamed Jrad, Sameer B. Mulani, and Rakesh K. Kapania. Integrated Global Wing and Local Panel Optimization of Aircraft Wing. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2015.

[54] Qiang Liu, Mohamed Jrad, Sameer B. Mulani, and Rakesh K. Kapania. Global/Local Optimization of Aircraft Wing Using Parallel Processing. *AIAA Journal*, 54(11):3338–3348, 2016.

[55] Joe Robinson, Steven Doyle, Grant Ogawa, Myles Baker, Shuvodeep De, Mohamed Jrad, and Rakesh Kapania. Aeroservoelastic Optimization of Wing Structure Using Curvilinear Spars and Ribs (SpaRibs). June 2016.

[56] Sameer B. Mulani, Wesley C. H. Slemp, and Rakesh K. Kapania. EBF3PanelOpt: An optimization framework for curvilinear blade-stiffened panels. *Thin-Walled Structures*, 63:13–26, February 2013.

[57] Mohamed Jrad, Shuvodeep De, and Rakesh K. Kapania. Global-local Aeroelastic Optimization of Internal Structure of Transport Aircraft wing. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2017.

[58] Wei Zhao and Rakesh K. Kapania. Bilevel Programming Weight Minimization of Composite Flying-Wing Aircraft with Curvilinear Spars and Ribs. *AIAA Journal*, 57(6):2594–2608, June 2019.

[59] Bret K. Stanford, Christine V. Jutte, and Christian A. Coker. Aeroelastic Sizing and Layout Design of a Wingbox Through Nested Optimization. *AIAA Journal*, 57(2):848–857, February 2019.

[60] B.K. Stanford. Shape, sizing, and topology design of a wingbox under aeroelastic constraints. volume 1 PartF, pages 1–11, 2020.

[61] B. Liu, R.T. Haftka, and M.A. Akgün. Two-level composite wing structural optimization using response surfaces. *Struct Multidisc Optim*, 20(2):87–96, October 2000.

[62] Ali Elham, Michel J. L. van Tooren, and Jaroslaw Sobieszczanski-Sobieski. Bilevel Optimization Strategy for Aircraft Wing Design Using Parallel Computing. *AIAA Journal*, 52(8):1770–1783, 2014. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.J052696.

[63] S. Ragon, Z. Gurdal, R. Haftka, T. Tzong, S. Ragon, Z. Gurdal, R. Haftka, and T. Tzong. Global/local structural wing design using response surface techniques. In *38th Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics.

[64] S. A. Ragon, Z. Gürdal, R. T. Haftka, and T. J. Tzong. Bilevel Design of a Wing Structure Using Response Surfaces. *Journal of Aircraft*, 40(5):985–992, 2003.

[65] J. B. Ransom and N. F. Knight. Global/local stress analysis of composite panels. *Computers & Structures*, 37(4):375–395, January 1990.

[66] Mohammad A. Aminpour, Susan L. McCleary, Jonathan B. Ransom, and Jerrold M. Housner. Global/local analysis method for treating details in structural design. volume 157, pages 119–137, 1992.

[67] John D. Whitcomb and Kyeongsik Woo. Evaluation of iterative global/local stress analysis. volume 10, pages 201–205, 1991.

[68] John D. Whitcomb and Kyeongsik Woo. Application of iterative global/local finite-element analysis. Part 1: Linear analysis. *Communications in Numerical Methods in Engineering*, 9(9):745–756, 1993.

[69] John Whitcomb, Kanthikannan Srirengan, and Clinton Chapman. Evaluation of homogenization for global/local stress analysis of textile composites. *Composite Structures*, 31(2):137–149, January 1995.

[70] R. K. Kapania, S. G. Haryadi, and R. T. Haftka. Global/local analysis of composite plates with cutouts. *Computational Mechanics*, 19(5):386–396, April 1997.

[71] S. G. Haryadi, R. K. Kapania, and R. T. Haftka. Global/local analysis of composite plates with cracks. *Composites Part B: Engineering*, 29(3):271–276, January 1998.

[72] C.C. Jara-Almonte and C.E. Knight. The specified boundary stiffness/force SBSF method for finite element subregion analysis. *International Journal for Numerical Methods in Engineering*, 26(7):1567–1578, 1988.

[73] R. J. GUYAN. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380–380, 1965.

[74] I. Hirai, B.P. Wang, and W.D. Pilkey. An efficient zooming method for finite element analysis. *International Journal for Numerical Methods in Engineering*, 20(9):1671–1683, 1984.

[75] I. Hirai, Y. Uchiyama, Y. Mizuta, and W.D. Pilkey. An exact zooming method. *Finite Elements in Analysis and Design*, 1(1):61–69, 1985.

[76] Sebastian Moritz Deinert. *Shape and Sizing Optimization of Aircraft Structures with Aeroelastic and Induced Drag Requirements.* PhD thesis.

Appendices

Appendix A

Static condensation

A.1 Condensation is exact

Given a system of equation:

$$Ku = p \tag{A.1}$$

and its block decomposition:

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} a \\ o \end{bmatrix} = \begin{bmatrix} p_a \\ p_o \end{bmatrix} \tag{A.2}$$

From the second block equation, it follows:

$$K_{oa}a + K_{oo}o = p_o \tag{A.3}$$

$$K_{oo}o = p_o - K_{oa}a \tag{A.4}$$

$$o = K_{oo}^{-1}p_o - K_{oo}^{-1}K_{oa}a \tag{A.5}$$

And the first block equation, can be re-written as:

$$K_{aa}a + K_{ao}o = p_a \tag{A.6}$$

$$K_{aa}a + K_{ao}K_{oo}^{-1}p_o - K_{ao}K_{oo}^{-1}K_{oa}a = p_a \tag{A.7}$$

$$\left[K_{aa} - K_{ao}K_{oo}^{-1}K_{oa}\right] a = p_a - K_{ao}K_{oo}^{-1}p_o \tag{A.8}$$

So given a system of equations $Ku = p$, if K_{oo} is invertible, it follows that the system is equivalent to:

$$\begin{cases} \left[K_{aa} - K_{ao}K_{oo}^{-1}K_{oa}\right] a = p_a - K_{ao}K_{oo}^{-1}p_o \\ K_{oo}o = p_o - K_{oa}a \end{cases} \tag{A.9}$$

A.2 Conditions for condensation

The static condensation of a system

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} a \\ o \end{bmatrix} = \begin{bmatrix} p_a \\ p_o \end{bmatrix}$$

is possible if K_{oo} is invertible.

Let the system represent a local model, where a are the DOFs at the interface and o are the internal DOFs.

If the local model is correctly defined, without internal mechanisms, then it has at most 6 dependent equations ($0 \leq dim(Ker(K) \leq 6$), corresponding to the 6 rigid body motions.

If fixing a constrains all 6 independent RBM, then the system $K_{oo}o = p_o - K_{oa}\bar{a}$ is solvable or equivalently K_{oo} is invertible.

Thus, if a local model becomes solvable when the interface nodes are constrained, it can be condensed.

Appendix B

Static aeroelastic formulation

B.1 Aerodynamic solution

The aerodynamic analysis considered in this work always assume an inviscid, incompressible, irrotational fluid, which can be solved with the potential theory. In particular the doublet lattice method is used [76], which solves the discretised equation:

$$[AIC]\ \Gamma = b_c \tag{B.1}$$

where $[AIC]$ is the matrix of aerodynamic influence coefficients, b_c is the vector of aerodynamic boundary conditions and Γ is the vector of vortex strength.

B.2 Fluid structure coupling

The aeroelastic problem is the result of the coupling of aerodynamic fluid solution and structural elastic deformation:

$$\begin{cases} Ku & = f(\Gamma) \\ [AIC]\ \Gamma & = b_c(u) \end{cases}$$

two linear system of equations, one for the static analysis and one for the aerodynamic analysis, whose solution affects the RHS of the other system.

B.3 Reformulation of the coupled problem

Instead of iterating between solving the structural and the aerodynamic equation, the problem can be rewritten in the form:

$$Ku = f([AIC]^{-1}\, b_c(u)) \tag{B.2}$$

$$Ku = p + f^{A}_{rigid} + Cu \tag{B.3}$$

$$\tag{B.4}$$

The first step is to divide $f(\Gamma)$ in two components:

$$f(\Gamma) = p + f^{A}(\Gamma) \tag{B.5}$$

where f_{Ae} represents the aerodynamic loads and p all the other (static) loads.

The aerodynamic forces depend on the difference in pressure coefficients:

$$f_{Ae}(\Gamma) = T_{SA} \cdot q \cdot A \cdot \Delta C_P(\Gamma) \tag{B.6}$$

The dependency on Γ is changed into a dependency on u using $[AIC]\ \Gamma = b_c(u)$:

$$\begin{aligned} \Delta C_{Pi} &= \frac{2}{c_i} \cdot \Gamma_i && \text{(B.7)} \\ &= \frac{2}{c_i} \cdot \left[AIC^{-1}\right]_{ij} \cdot b_{cj} && \text{(B.8)} \\ &= [AIC_{C_P}]_{ij} \cdot b_{cj} && \text{(B.9)} \end{aligned}$$

$$\Delta C_P(\Gamma) = [AIC_{C_P}] \cdot b_c(u) \tag{B.10}$$

So the aerodynamic loads become:

$$f_{Ae}(u) = T_{SA} \cdot q \cdot A \cdot [AIC_{C_P}] \cdot b_c(u) \tag{B.11}$$

With this, (eq. B.5) now becomes:

$$f(u) = p + f^{A}(u) \tag{B.12}$$

And the coupled problem can be rewritten in a single equation as:

$$Ku = p + f^A(u) \tag{B.13}$$

This formulation can be further manipulated by decomposing $b_c(u)$ as:

$$\begin{aligned} b_c(u) &= b_{c_{state}} + b_{c_{shape}} + b_{c_{elastic}}(u) && \text{(B.14)} \\ &= b_{c_{state}} + b_{c_{shape}} + T_{AS} \cdot u && \text{(B.15)} \\ & && \text{(B.16)} \end{aligned}$$

which leads to reformulating equation $(B.11)$ as:

$$\begin{aligned} f_{Ae}(u) &= T_{SA} \cdot q \cdot A \cdot AIC_{C_P} \cdot \left(b_{c_{state}} + b_{c_{shape}} + b_c(u)\right) && \text{(B.17)} \\ &= T_{SA} \cdot q \cdot A \cdot AIC_{C_P} \cdot \left(b_{c_{state}} + b_{c_{shape}}\right) + T_{SA} \cdot q \cdot A \cdot AIC_{C_P} T_{AS} \cdot u && \text{(B.18)} \\ &= f^A_{rigid} + C \cdot u && \text{(B.19)} \end{aligned}$$

where the aeroelastic stiffness matrix is defined as:

$$C = q \cdot T_{SA} \cdot A \cdot AIC_{C_P} \cdot T_{AS} \tag{B.20}$$

With this, $B.5$ becomes:

$$f(u) = p + f^A_{rigid} + C \cdot u \tag{B.21}$$

and the coupled problem yields the discretised equation used in this thesis:

$$Ku = p + f^A_{rigid} + C \cdot u \tag{B.22}$$

Appendix C

Methodology derivation

Given the global system of equations for static analysis:

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i \end{bmatrix} \tag{C.1}$$

the local condensed system of equation:

$$K^{\dagger}_{aa} i = p^{\dagger}_{a} \tag{C.2}$$

and the coupling

$$i = a \tag{C.3}$$

which can be written as:

$$\begin{bmatrix} z \\ i \\ a \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \cdot \\ \cdot & \mathbb{I} \\ \cdot & \mathbb{I} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} \tag{C.4}$$

It follows that:

$$\begin{bmatrix} K_{zz} & K_{zi} & \cdot \\ K_{iz} & K_{ii} & \cdot \\ \cdot & \cdot & K_{aa}^{\dagger} \end{bmatrix} \begin{bmatrix} z \\ i \\ a \end{bmatrix} = \begin{bmatrix} p_z \\ p_i \\ p_a^{\dagger} \end{bmatrix} \tag{C.5}$$

$$\begin{bmatrix} \mathbb{I} & \cdot & \cdot \\ \cdot & \mathbb{I} & \mathbb{I} \end{bmatrix} \begin{bmatrix} K_{zz} & K_{zi} & \cdot \\ K_{iz} & K_{ii} & \cdot \\ \cdot & \cdot & K_{aa}^{\dagger} \end{bmatrix} \begin{bmatrix} \mathbb{I} & \cdot \\ \cdot & \mathbb{I} \\ \cdot & \mathbb{I} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \cdot & \cdot \\ \cdot & \mathbb{I} & \mathbb{I} \end{bmatrix} \begin{bmatrix} p_z \\ p_i \\ p_a^{\dagger} \end{bmatrix} \tag{C.6}$$

$$\begin{bmatrix} K_{zz} & K_{zi} \\ K_{iz} & K_{ii} + K_{aa}^{\dagger} \end{bmatrix} \begin{bmatrix} z \\ i \end{bmatrix} = \begin{bmatrix} p_z \\ p_i + p_a^{\dagger} \end{bmatrix} \tag{C.7}$$

The same applies to the case of a global aeroelastic analysis.

Appendix D

Computational cost of computing $\frac{dP}{dx}$

The static part of the pseudo-load vector $\dfrac{dP^G}{dx^L}$ is computed according to eq. 4.31. The computational cost of computing $\dfrac{dK^{\dagger}_{aa}}{dx^L}i$ according to eq. 5.10 is:

$$\begin{aligned}
C_{\mathrm{dK}^*} &= O(n_o {n_a}^2)n_L && \left(\text{compute } \frac{dK_{oa}}{dx^L}i\right) && \text{(D.1)}\\
&+ O({n_o}^2)n_L && \left(\text{left multiply } \frac{dK_{oa}}{dx^L}i \text{ by } y \text{ (cached)}\right) && \text{(D.2)}\\
&+ O(n_o {n_a}^2) && (\text{compute } K_{oa}i \text{ (could be cached)}) && \text{(D.3)}\\
&+ O({n_o}^2) && \left(\text{left multiply } K_{oa}i \text{ by } K_{oo}^{-1}\right) && \text{(D.4)}\\
&+ O(n_a {n_o}^2)n_L && \left(\text{left multiply } K_{oo}^{-1}K_{oa}i \text{ by } \frac{dK_{ao}}{dx^L}\right) && \text{(D.5)}\\
&+ O({n_o}^3)n_L && \left(\text{left multiply } K_{oo}^{-1}K_{oa}i \text{ by } \frac{dK_{oo}}{dx^L}\right) && \text{(D.6)}\\
&+ O({n_o}^2)n_L && \left(\text{left multiply } \frac{dK_{oo}}{dx^L}K_{oo}^{-1}K_{oa}i \text{ by } y\right) && \text{(D.7)}
\end{aligned}$$

The computational cost of computing $\frac{dp_a^\dagger}{dx^L}$ is:

$$\begin{aligned}
C_{\text{dp}^*} &= O({n_o}^2) && \big(\text{compute } K_{oo}^{-1}p_o\big) && \text{(D.8)}\\
&+ O(n_a {n_o}^2)n_L && \left(\text{left multiply } K_{oo}^{-1}p_o \text{ by } \frac{dK_{ao}}{dx^L}\right) && \text{(D.9)}\\
&+ O({n_o}^3)n_L && \left(\text{left multiply } K_{oo}^{-1}p_o \text{ by } \frac{dK_{oo}}{dx^L}\right) && \text{(D.10)}\\
&+ O({n_o}^2)n_L && \left(\text{left multiply } \frac{dK_{oo}}{dx^L}K_{oo}^{-1}p_o \text{ by } y\right) && \text{(D.11)}\\
&+ O({n_o}^2)n_L && \left(\text{left multiply } \frac{dp_o}{dx^L} \text{ by } y\right) && \text{(D.12)}
\end{aligned}$$

Since $n_o \gg n_a$, the total computational cost of computing $\frac{dP}{dx}$ is therefore $O({n_o}^3)$, for each local model.

Appendix E

Rationale for choosing between direct and adjoint

In sensitivity analysis, objective and constraints are derived with respect to the design variables. These usually depend on the design variables directly and indirectly through the solution field.

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \frac{\partial g}{\partial u}\frac{du}{dx} \tag{E.1}$$

In the case of static analysis, when computing the implicit part of the gradient:

$$\begin{aligned} (\text{implicit}) &= \frac{\partial g}{\partial u}\frac{du}{dx} && \text{(E.2)} \\ &= \frac{\partial g}{\partial u}K^{-1}P^{\star} && \text{(E.3)} \end{aligned}$$

one has to essentially solve a product of three matrices. (The formula above, valid for static analysis is proven here). In this specific case the dimensions are:

$$(C \times DOF)(DOF \times DOF)(DOF \times DV) \tag{E.4}$$

where:

$$\begin{aligned} &C\text{: number of constraints} && \text{(E.5)} \\ &DOF\text{: number of degrees of freedom} && \text{(E.6)} \\ &DV\text{: number of design variables} && \text{(E.7)} \end{aligned}$$

The direct and adjoint method are different, in that they solve these products in a different order. In practice, one of the two products is obtained as a solution of a linear system of equations, for K^{-1} is not explicitly computed.

E.1 Direct method

In the direct method,

$$\frac{du}{dx} = K^{-1}P^{\star} \tag{E.8}$$

is actually obtained by solving:

$$K\frac{du}{dx} = P^{\star}$$

the linear system has a size of DOF and must be solved DV times.

Then the implicit part is computed as:

$$\text{(implicit)} = \frac{\partial g}{\partial u}\frac{du}{dx} \tag{E.9}$$

E.2 Adjoint method

Similarly, let a be:

$$a = \frac{\partial g}{\partial u}K^{-1} \tag{E.10}$$

by right multiplying both sides by K, the relation becomes:

$$aK = \frac{\partial g}{\partial u} \tag{E.11}$$

so a can be obtained by solving the adjoint system:

$$K^T a^T = \frac{\partial g}{\partial u}^T \tag{E.12}$$

which gives the method its name. The system has a size of DOF and must be solved C times. The implicit part is computed as:

$$\text{(implicit)} = aP^{\star} \tag{E.13}$$

E.3 Comparison of computational cost

Let $L(DOF)$ be the computational cost of solving a linear system of equations.

The product of two matrices of size $(a \times b)(b \times c)$ requires to compute b products for each entry of the resulting matrix. Thus an estimate of the computaional cost is: $n_P = abc$. In the direct method,

$$(C \times DOF)\Big[(DOF \times DOF)(DOF \times DV)\Big] \tag{E.14}$$

the computational cost is:

$$L(DOF) \cdot DV + C \cdot DOF \cdot DV \tag{E.15}$$

In the adjoint case,

$$\Big[(C \times DOF)(DOF \times DOF)\Big](DOF \times DV) \tag{E.16}$$

the computational cost is:

$$C \cdot L(DOF) + C \cdot DOF \cdot DV \tag{E.17}$$

Therefore, if $C > DV$, the direct method requires fewer operations, while if instead $DV > C$, the adjoint method is more convenient:

$$\begin{cases} C > DV \implies \text{choose } \textbf{direct} \\ DV > C \implies \text{choose } \textbf{adjoint} \end{cases} \tag{E.18}$$

The same argument can be made for disciplines other than static analysis.