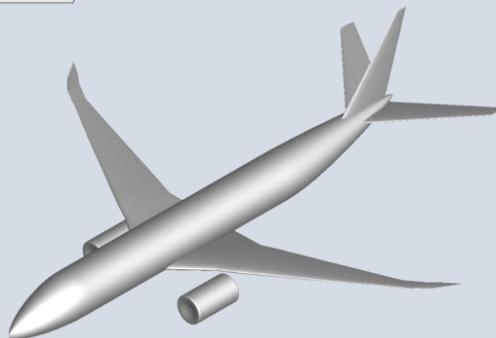
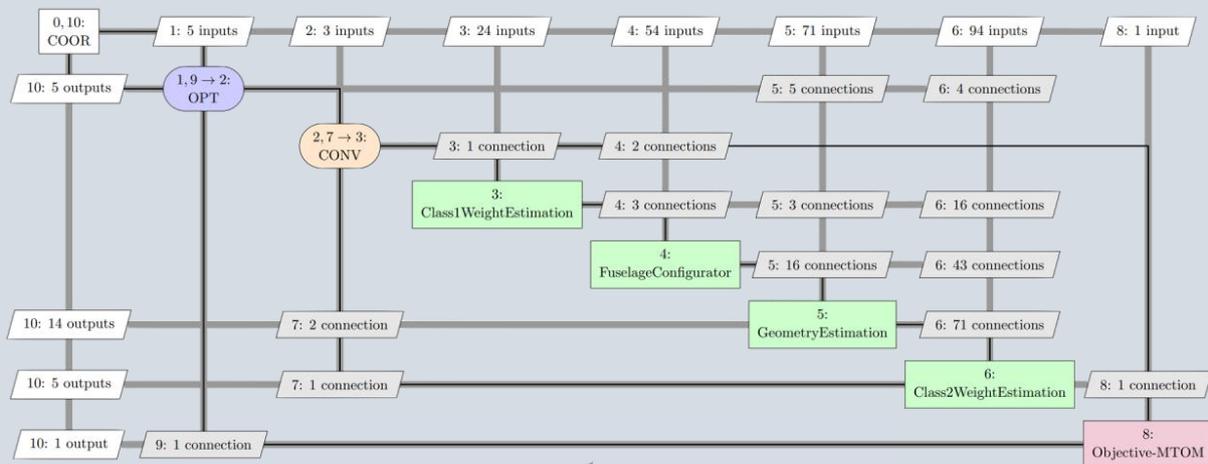
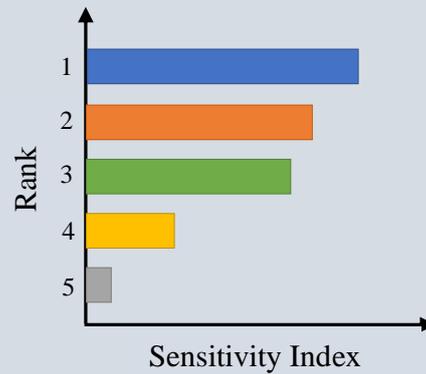


Adjustment of the MDAO Problem Formulation using Sensitivity Analysis to Reduce the Computational Cost within Aircraft Design

Max Rein

Technische Universiteit Delft



ADJUSTMENT OF THE MDAO PROBLEM FORMULATION USING SENSITIVITY ANALYSIS TO REDUCE THE COMPUTATIONAL COST WITHIN AIRCRAFT DESIGN

by

Max Rein

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Monday December 12, 2022 at 1:00 PM.

Thesis committee:	Dr.ir. G. la Rocca,	TU Delft, supervisor
	Dr.ir. M.B. Zaaijer,	TU Delft
	Dr.ir. M.E.M. Hoogreef,	TU Delft
	ir. A.M.R.M. Bruggeman,	TU Delft, daily supervisor

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

SUMMARY

An increasing demand to develop sustainable solutions and technologies within the aerospace industry and the extensive amount of data within each engineering discipline of aircraft design, has advanced the development of the Multidisciplinary Design Analysis and Optimization (MDAO) technology. Utilizing MDAO in the aircraft design process allows to obtain an improved design from a holistic point of view by capturing and including the interactions between different disciplines in the design problem. Currently, MDAO is not yet widely used in the aerospace industry, as there are several obstacles hindering its full implementation, especially the generation of considerable computational expense when dealing with large-scale problems. Therefore, this thesis investigated on how to adjust the MDAO problem formulation with the use of sensitivity analysis and thus to consequently alleviate the computational expense of MDAO problems. This thesis achieved to contribute a new approach to the existing literature on how to adjust the MDAO problem formulation by developing and subsequently incorporating a sensitivity analysis tool into the sequencing and decomposition algorithms of KADMOS (Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System), a software that is used within the MDAO formulation phase.

The two methodologies that were created within this thesis develop, evaluate, and incorporate a sensitivity analysis tool to improve the MDAO problem formulation with the goal to reduce the execution time of the MDAO problem. The aim of the first methodology is to identify and remove design variables that have a non-influential effect on the objective by utilizing global sensitivity methods, as these allow to characterize the input variables on the entire input space. Concurrently, using this method, it was also possible to determine the key design variables for an aircraft design problem. The second methodology uses local sensitivity methods, as they only characterize around a single point, to improve the existing sequencing and decomposition algorithms in KADMOS. The goal is to optimize the execution order and the distribution of disciplines over various partitions, as the incorporation of sensitivity information into the decomposition algorithms allows to gather disciplinary tools with low sensitivities in the one partition while collecting disciplinary tools with high sensitivity in another partition, resulting in reducing the number of necessary iterations. Subsequently, both methodologies were validated and verified with the use of mathematical problems. A variable complexity problem was used to evaluate that a feedback loop containing low sensitivities will lead to fewer iterations compared to a feedback loop with higher sensitivities, thus leading to a reduction in computational time.

The methodologies were tested on an aircraft design problem and efficiently demonstrated the identification and removal of low sensitivity design variables from the problem formulation, with minor impact on accuracy. However, it was determined that it did not speed up the whole computation time due to the large computational time associated with performing sensitivity analyses. Anyhow, it was evaluated that performing sensitivity analysis concurrently on more than one objective variable during the MDAO problem formulation allowed for time savings and for reduction of the computational cost. Overall, this thesis demonstrated that although incorporating sensitivity information in the MDAO problem formulation phase is advantageous, the computational effort to perform the sensitivity analysis may diminish the benefits. To make the adjustment of the MDAO formulation with sensitivity information a standardized and widely used procedure, its application has to be further advanced in future, respectively new methodologies could be developed.

CONTENTS

List of Figures	vii
List of Tables	ix
List of Symbols	xi
List of Abbreviations	xiii
1 Introduction	1
1.1 Research Question	3
1.2 Research Objective	4
2 State of the Art	5
2.1 DEFAINE Project	5
2.2 KADMOS	6
2.2.1 Execution Phase of the Problem Formulation	8
2.2.2 Sequencing and Decomposition	8
3 Literature Review	11
3.1 Introduction	11
3.2 Local and Global Sensitivity Analysis	12
3.3 One-At-a-Time and Many-At-a-Time Sensitivity Methods	13
3.4 Sampling Strategies	13
3.5 Review of Sensitivity Methods	13
3.5.1 Morris Method	14
3.5.2 Sobol Method	15
3.5.3 FAST and eFAST Method	16
3.5.4 Derivative-Based Local Sensitivities	17
3.6 Software for Performing Sensitivity Analysis	18
4 Methodology	19
4.1 Methodology 1: Identify Non-Influential Design Variables	19
4.1.1 Phase 1: Initial Formulation	20
4.1.2 Phase 2: Analyze Problem	20
4.1.3 Phase 3: Formulation Adjustment and Phase 4: Execution	21
4.2 Methodology 2: Sequencing and Decomposition with use of Sensitivity Information	21
4.2.1 Performing Sensitivity Analysis	22
4.2.2 Creating a Coupling Dictionary	23
5 Verification & Validation	25
5.1 Verification and Validation of the Sensitivity Methods	25
5.2 Verification and Validation of the Sequencing and Decomposition Process	26
6 Case Study 1	31
6.1 The Aircraft Design Initiator	31
6.2 The Conceptual Design of an Airbus A320-200	33
6.2.1 MDAO Problem Formulation	35
6.2.2 Sensitivity Analysis to Identify Non-Influential Design Variables	38
6.2.3 Optimization of the MDAO Problem Formulation	41
7 Case Study 2	45
7.1 Problem Formulation	45
7.2 Sensitivity Analysis	47
7.2.1 Minimizing the Fuel Mass	47

8	Case Study 3	49
9	Conclusion & Recommendations	55
A	Verification & Validation	57
A.1	XDSM of the Variable Complexity Problem	57
B	Initiator Sensitivity Analysis	59
B.1	Initiator Inputs for Sensitivity Analysis using the LHS Strategy	59
B.2	Initiator Input File	59
B.3	Initiator Output File.	64
C	Complete XDSM of Initiator Optimization	65
D	Case Study 2	67
D.1	Output File	67
	Bibliography	69

LIST OF FIGURES

1.1	Overview of the MDAO development process [13].	2
2.1	Overview of the framework for the DEFAINE project ⁴	5
2.2	Example of the data and process graph used in KADMOS [11].	6
2.3	Example of an eXtended Design Structure Matrix (XDSM).	7
2.4	Overview of the full MDAO development process.	7
2.5	Example to explain sequencing algorithms.	9
2.6	Example to illustrate the decomposition process.	10
3.1	Ranking, screening, and mapping of sensitivity analysis [25].	12
3.2	Example of the Morris plot.	15
3.3	Apportioning output variance for ANOVA methods.	15
3.4	Example of the visualization of the sensitivity indices for the ANOVA methods.	17
3.5	Finite-difference intervals [48].	17
4.1	Overview of the steps performed in the first methodology where non-influential design variables are identified.	19
4.2	Example problem consisting of four disciplines with the corresponding sensitivity information.	21
4.3	Comparison of the sequencing solution for the two different methods.	22
4.4	Overview of the workflow for implementing sensitivity analysis within the sequencing and decomposition algorithms.	22
4.5	Coupling dictionary for the example problem as shown in Figure 4.2.	23
4.6	Overview of the branch-and-bound algorithm applied to the example problem when using sensitivity information.	23
4.7	Final sequence obtained by the branch-and-bound algorithm using sensitivity information.	24
5.1	The repository connectivity graph of a variable complexity problem.	27
5.2	Left: FPG with the number minimum number of feedback connections, Right: FPG with sensitivity analysis.	28
5.3	RCG of the variable complexity problem with four disciplines used for the verification and validation.	28
5.4	FPGs of the sequencing using different algorithms.	29
5.5	FPGs of the decomposition using different methods.	30
6.1	Schematic process diagram of the Initiator [60].	32
6.2	The central data schema for the Initiator [14].	33
6.3	The repository connectivity graph for the design of an Airbus A320-200.	34
6.4	Gauss-Seidel convergence with one partition of the aircraft design problem when minimizing for MTOM.	36
6.5	Partition convergence with two partitions of the aircraft design problem when minimizing for MTOM.	37
6.6	Morris plot of the sensitivity results with five trajectory designs.	39
6.7	Morris plot of the sensitivity results with ten trajectory designs.	40
6.8	XDSM of the optimization for minimum MTOM using a MDF Gauss-Seidel architecture.	42
6.9	Wing thrust loading diagram for the problem with the removal of two design variables.	43
6.10	History of MTOM and the design variables for the minimization of the MTOM.	43
7.1	Problem formulation using the Gauss-Seidel convergence for the two objectives: MTOM and fuel mass.	46
7.2	Morris plot of the sensitivity results for minimizing fuel mass.	47

8.1	Comparison between the average number of iterations for the two approaches: when using sensitivity data and for minimum number of feedback for sequencing and decomposition algorithm.	50
8.2	Comparison of the FPGs for test case 4 for the two different approaches.	51
8.3	Comparison of the FPGs for test case 9 for the two different approaches.	52
8.4	Comparison between the average number of iterations when using sensitivity information and for minimum number of feedback in the sequencing and decomposition algorithm.	53
A.1	XDSM of the MDA to be converged for the variable complexity problem with the approach of minimizing the number of feedbacks.	57
A.2	XDSM of the MDA to be converged for the variable complexity problem with the approach of using sensitivity information.	58

LIST OF TABLES

3.1	Summary of sensitivity analysis methods covered in this thesis including whether they are global or local methods, detect interactions between factors, capability to handle non-linearities in the model, the computational cost, and aim(s).	14
4.1	Overview of the required number of problem evaluations for each sensitivity method, where k is the number of design variables, n is the minimum number of model evaluations for estimating one individual effect, N_s is the number of samples per parameter, N_R is the re-sampling size, and r is the number of trajectories.	20
4.2	Overview of the sampling method for each sensitivity method.	20
5.1	Comparison between the Sobol sensitivity indices and the analytical values.	25
5.2	Comparison between the FAST sensitivity indices and the analytical values.	25
5.3	Results of the Morris method for the Ishigami function.	26
5.4	Chosen parameters for the variable complexity problem.	26
5.5	Comparison between the analytical values and sensitivity values using the finite-difference method for the first discipline (D1).	27
5.6	Comparison of the results of the optimization between the minimum number of feedback implementation and the implementation with sensitivity information.	28
5.7	The sum of sensitivities in the feedback for all possible sequences.	29
6.1	Overview of the design variables and their bounds for the aircraft design problem of case study 1.	33
6.2	Overview of the constraints for the aircraft design problem of case study 1.	35
6.3	Airbus A320-200 top level requirements	35
6.4	Execution time of each Initiator module.	38
6.5	Convergence results of the two different architectures: Gauss-Seidel convergence with one partition and a partition convergence with two partitions for the aircraft design problem.	38
6.6	Number of model evaluations required for each global sensitivity method.	38
6.7	Results of the sensitivity analysis using the Morris method.	39
6.8	Results of the sensitivity analysis using the Morris method with ten trajectory designs.	40
6.9	Results of the optimization for the original problem and the problem with the removal of two non-influential design variables.	41
6.10	Evaluation time of the optimization and sensitivity analysis.	44
7.1	Results of the sensitivity analysis using the Morris method for minimizing fuel mass.	47
8.1	Comparison of the number of iterations for each test case for the two approaches: the minimum number of feedback implementation and the implementation with sensitivity data.	50
8.2	Comparison of the convergence history for test case 4.	51
8.3	Comparison of the convergence history for test case 9.	52
8.4	Average number of iterations and confidence level.	53

LIST OF SYMBOLS

η_{eng}	=	Non-dimensional engine spanwise position
η_{kink}	=	Non-dimensional wing kink position
μ^*	=	Mean of the absolute value of elementary effect
σ	=	Standard deviation of elementary effect
AR	=	Aspect ratio
EE	=	Elementary effect
f	=	Objective value
k	=	Number of design variables
n_{iter}	=	Number of iterations
S_i	=	First-order sensitivity index
S_{tot}	=	Total-order sensitivity index
t_{iter}	=	Execution time per iteration
t_{tot}	=	Total execution time
TR	=	Taper ratio
T/W	=	Thrust-to-weight ratio
V	=	Output variance
W/S	=	Wing loading
\mathbf{c}	=	Constraint vector
\mathbf{x}	=	Design vector
\mathbf{y}	=	Coupling vector

LIST OF ABBREVIATIONS

<i>ANOVA</i>	=	ANalysis of VAriance
<i>AVL</i>	=	Athena Vortex Lattice
<i>CMDOWS</i>	=	Common MDO Workflow Schema
<i>CPACS</i>	=	Common Parametric Aircraft Configuration Schema
<i>DEFaine</i>	=	Design Exploration Framework based on AI for front-loaded Engineering
<i>DoE</i>	=	Design of Experiment
<i>FAST</i>	=	Fourier Amplitude Sensitivity Test
<i>FPG</i>	=	Fundamental Problem Graph
<i>KADMOS</i>	=	Knowledge and graph-based Agile Design for Multidisciplinary Optimization System
<i>LHS</i>	=	Latin Hypercube Sampling
<i>MAT</i>	=	Many-At-a-Time
<i>MDA</i>	=	Multidisciplinary Analysis
<i>MDAO</i>	=	Multidisciplinary Design Analysis and Optimization
<i>MDF</i>	=	Multidisciplinary Feasible
<i>MDG</i>	=	MDAO Data Graph
<i>MDK</i>	=	Metis-based Decomposition of KADMOS graphs
<i>MPG</i>	=	MDAO Process Graph
<i>MTOM</i>	=	Maximum Take-Off Mass
<i>OAT</i>	=	One-At-a-Time
<i>OEM</i>	=	Operational Empty Mass
<i>RCB</i>	=	Runtime Coupling Balance
<i>RCE</i>	=	Remote Component Environment
<i>RCG</i>	=	Repository Connectivity Graph
<i>VCP</i>	=	Variable Complexity Problem
<i>XDSM</i>	=	eXtended Design Structure Matrix

1

INTRODUCTION

The aerospace industry is facing extensive challenges, as the growing importance of sustainability considerations and the requested reduction of the environmental impact of aviation need to be incorporated into the design process of novel aircraft configurations. Especially to be emphasized is the planned goal of reducing carbon dioxide (CO_2) by 75%, nitrogen oxide emissions (NO_x) by 90%, noise by 65% until 2050, and to increase fuel efficiency by 2% annually until 2050 [1, 2]. Therefore, significant technological innovations in engine and airframe design are needed and can only be achieved through substantial technical advancements, like the development of cutting-edge future aircraft configurations and technologies. Subsequently, aircraft design will become even more complex in the future. Already, current aircraft design represents a highly intricate and interdisciplinary process where numerous discipline domains need to be considered and incorporated simultaneously. The engineering domains, like structure, propulsion, flight control, performance, weights, and aerodynamics, are mutually interdependent and their integration to obtain a particular aircraft configuration is a highly iterative process. It requires the continuous optimization by a multidisciplinary system approach and involves compromises in each discipline. Consequently, the design goal is, for instance, to identify all relevant future aircraft configurations and technologies, respective to the projected objective, and to achieve the most suitable integration of all aerospace engineering disciplines and aircraft components into, for example, an efficient and sustainable aircraft that complies with all the requirements.

Continuous digitalization of aircraft design has generated an extensive amount of data for each engineering discipline which need to be collected, analyzed, and evaluated, creating a considerable technical complexity of the aircraft design process. Furthermore, the integration of all relevant data, respective to the objective, into the design product results in an extreme time- and cost-consuming expense. Therefore, the aircraft design process has to be made more cost-effective and efficient. The ultimate goal is to shorten the time to market for new products, variants, or technologies. Consequently, particular emphasis has been focused on the development of new strategies to generalize and facilitate the aircraft design process. A successful approach is using the innovative technology of Multidisciplinary Design Analysis and Optimization (MDAO) [3–5]. MDAO enables the efficient analysis and optimization of a design that consists of a complex system where multiple coupled disciplines interact. Additionally, a further goal of the application of MDAO is the modeling of MDAO workflows which consequently allow the repetitive design and optimization of an aircraft design system, respectively to comparatively assess different aircraft configurations. MDAO is discussed below and in Chapter 2. The research project AGILE (Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts) was established to further enhance the use of MDAO for aircraft design [6, 7]¹. AGILE propagates a collaborative MDAO platform which has been created by experts and organizations from several countries. Consequently, this facilitates the aircraft design process by, for example, establishing the formulation of an automated collaborative MDAO workflow. This was further enhanced by the development of a novel design methodology within MDAO, the “AGILE Paradigm” and the development of the AGILE Framework. The AGILE Paradigm consists of the knowledge architecture, which supports the formulation of the design problem and the selection of the most suitable solution strategy, and the collaborative architecture, which focuses on cross-national and cross-organizational integration of design competencies into MDAO workflows [8]. One of the software developed within the AGILE Framework is KADMOS

¹<https://www.agile-project.eu/>, accessed: 31.08.2022

(Knowledge-and graph-based Agile Design for Multidisciplinary Optimization System), using graph-based analysis for the formulation of an MDAO problem and the MDAO solution strategy. AGILE lead to a 40%-time reduction for setup and solving of an MDAO problem [8, 9]. The subsequent AGILE 4.0 project automates the collaborative MDAO workflow further in also digitalizing and incorporating data from various levels of the aeronautical supply chain, like design, production, certification, and manufacturing². It thus alleviates the optimization of the entire life cycle of an airplane. The additional project DEFAINE (Design Exploration Framework based on AI for front-loaded Engineering) provides a technological platform for an advanced design exploration framework, based on principles of artificial intelligence and machine learning³. The purpose of DEFAINE is to improve early stages of design solutions and subsequently reduce lead time and costs of aircraft design. DEFAINE is discussed later in Section 2.1. This thesis aims to contribute to DEFAINE and AGILE 4.0 in introducing a novel approach to consider sensitivity data, obtained from sensitivity analysis, into the sequencing and decomposition algorithms of KADMOS and also, identify non-influential design variables, respective to the objective, during the MDAO problem formulation procedure. The purpose of this research is to determine if this consideration of sensitivity data during the MDAO formulation phase will ensure a further reduction of MDAO execution time and consequently in lead time and costs of aircraft design.

Despite the benefits of MDAO and the before discussed applications, it is not yet widely utilized in the aerospace industry, as there are numerous obstacles that need to be resolved [10, 11]. In particular, due to the increasing complexity of aeronautical systems, MDAO problems tend to become large, have a long setup time, and a significant computational expense [12].

For a better understanding of the MDAO problem formulation and how it is set up into the MDAO development process, refer to Figure 1.1. There are two main phases of the MDAO development process: MDAO formulation, depicted on the left side of Figure 1.1, and MDAO execution phase, depicted on the right side of Figure 1.1.

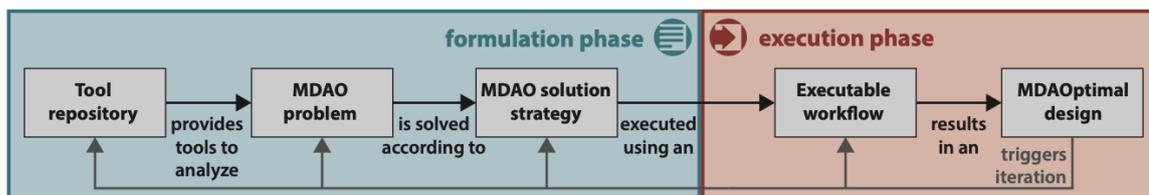


Figure 1.1: Overview of the MDAO development process [13].

To obtain an efficient and easy implementable MDAO problem, MDAO problem formulation needs to be carefully evaluated, as the formulation plays a key role in reducing the computational time. The first step of the MDAO problem formulation is to establish a tool repository which contains the tools, like for example the disciplinary analyses, which are needed to analyze the design. The next step of the formulation is the definition of the MDAO problem - the relevant design problem, the design variables and constraints, the respective objective, and the quantities of interest need to be specified. Then, for example, for each disciplinary tool an input and an output CPACS (Common Parametric Aircraft Configuration Schema) file is formulated which together constitute the database and subsequently is used to analyze the MDAO problem. The database of the tool repository is processed using KADMOS (Knowledge-and graph-based Agile Design for Multidisciplinary Optimization System) [11]. Consequently, an MDAO solution strategy, is defined and derives the formal MDAO model, meaning that it is not executable, of the MDAO system. The solution strategy is iterated over the MDAO problem until the optimal solution strategy, or MDAO architecture, is derived. The MDAO architecture determines how the execution process is coordinated in specifying the interaction between tools and other elements, like optimizers and convergers.

For this work, the software package KADMOS is utilized for the formulation phase of the MDAO development process. A more detailed review of KADMOS and how it is incorporated into this work is outlined in the next chapter, in Section 2.2 and Figure 2.4. The formulation phase includes sequencing and decomposition of the MDAO system and affects the execution time of the MDAO system. Sequencing is defined as the determination of the execution order, whereas decomposition is defined as the distribution of the disciplines over various partitions which can be executed successively or in parallel. Currently, various sequencing and

²<https://www.agile4.eu/>, accessed: 31.08.2022

³<https://www.defaine.eu/>, accessed: 17.08.2022

decomposition algorithms are implemented in KADMOS [14]. The algorithms that are used in KADMOS are extensively discussed in Section 2.2.2.

This thesis strives to contribute to KADMOS by integrating sensitivity analysis into the MDAO formulation process. For this purpose, this thesis uses two different approaches to use sensitivity data as a tool to enhance the optimization of the MDAO formulation. Firstly, the use of global sensitivity data will help to identify design variables of low influence and consequently will speed up the execution time of MDAO by only considering variables that are influential in respect to the requested objective during the MDAO formulation. Secondly, the consideration of local sensitivity data permits an optimization of the solution strategy, as the sequencing and partitioning can be improved. Currently, the algorithms that are used in KADMOS, do not use sensitivity information. Therefore, a focus of this thesis is to enhance these algorithms by implementing sensitivity information into them in order to adjust the MDAO problem formulation. For sequencing, the consideration of local sensitivity information permits to establish a sequence where the variables with low sensitivity are placed in the feedback, which may lead to fewer iterations, thus, reducing the computational time. For decomposition, the incorporation of local sensitivity information helps to collect disciplinary tools with low sensitivities in the same partition and disciplinary tools with high sensitivities in a different partition which will result in reducing the number of necessary iterations. Detailed information of how sensitivity analysis can improve sequencing and decomposition is explained in Section 4.2. Consequently, this thesis provides a contribution to current scientific literature, research, and enhancement of MDAO as it propagates a novel approach to include sensitivity information into the algorithms of KADMOS during the MDAO formulation. Hence, this thesis supports the progressive advancement of MDAO for its application within the aerospace industry and the automation of the development of a collaborative MDAO workflow.

The formulated problem is then passed onto the execution phase (right part of Figure 1.1), where the system is solved to find the desired result, for example, the optimal design or an DoE.

1.1. RESEARCH QUESTION

The main research question of this thesis is:

To what extent can the MDAO problem formulation be adjusted, in order to reduce the MDAO execution time, by utilizing sensitivity data to refine the sequencing and decomposition algorithms of KADMOS and to determine any non-influential variables, in respect to the requested objective?

The main research question is split into two sub-questions:

1. To what extent can the existing sequencing and decomposition algorithms of KADMOS be refined by using sensitivity analysis in order to improve the MDAO problem formulation?

A goal of this thesis is to improve the sequencing and decomposition algorithms of KADMOS by implementing sensitivity information, in order to optimize the execution order and the partitions. Sensitivity information will be used to investigate if a feedback with low sensitivities will lead to fewer iterations until a converged MDAO solution is reached, compared to a feedback with high sensitivities. In addition, for the decomposition one could gather disciplinary tools with strong sensitivities in the same partition and at the same time to collect disciplinary tools with low sensitivities in a different partition. This may minimize the dependency between the various partitions and can lead to a reduction in iterations.

2. To what extent could the number of design variables be reduced to decrease the computational expense while still obtaining accurate results with the use of sensitivity analysis?

As the overall MDAO optimization cost strongly depends on the number of design variables, it is favorable to reduce them in the MDAO problem formulation. Sensitivity data, derived from sensitivity analysis, reveal how the design variables have an influence on the objective function and the level of coupling between systems. Consequently, this thesis investigates the application of sensitivity analysis as a method for the reduction of design space and thus control of the dimensionality of the MDAO problem and strives to identify criteria when a variable can be removed.

1.2. RESEARCH OBJECTIVE

The main research objective of this thesis is:

To advise and enable adjustments of the MDAO problem formulation by developing and implementing a sensitivity analysis tool that generates, evaluates, and incorporates sensitivity information, derived from local and global sensitivity analysis.

For the execution of the research objective several sub-goals are formulated:

1. Develop:

- A sensitivity analysis tool that will provide the user advice on how to adjust the MDAO problem formulation
- Two independent methodologies to use sensitivity information during the MDAO problem formulation.
 - Methodology 1 will be utilizing global sensitivity methods to identify non-influential design variables.
 - Methodology 2 will be utilizing local sensitivity methods to improve the sequencing and decomposition algorithms of KADMOS.
- A methodology to assess the influence of the adjusted MDAO problem formulation on the MDAO execution time and on the accuracy of the obtained results.

2. Test and verify the implemented sensitivity analysis tool:

- Test the full MDAO system formulation and execution process on a theoretical mathematical problem and on a realistic aircraft design problem.

3. Answer these questions:

- To what extent does the incorporation of sensitivity information into the sequencing and decomposition algorithms of KADMOS benefit MDAO formulation?
- To what extent can the MDAO problem formulation be adjusted when non-influential design variables, in respect to the requested objective, are removed?
- How much will both approaches reduce the MDAO execution time?
- To what extent is integrating sensitivity analysis an efficient tool to reduce convergence time of MDAO?

This thesis will follow the subsequent outline: After the previous introduction chapter, Chapter 2 will outline the State of the Art for DEFAINE and KADMOS, Chapter 3 will provide a literature review about sensitivity analysis, Chapter 4 the methodology, Chapter 5 the verification and validation, three case studies will be evaluated in Chapter 6, 7, and 8, and the thesis finishes with Chapter 9 with the conclusion and recommendations.

2

STATE OF THE ART

A description of the DEFAINE project and how this thesis is connected to the initiatives of the project is given in Section 2.1. Afterwards, Section 2.2. gives an overview of how the software package KADMOS formulates the MDAO system based on a repository of tools.

2.1. DEFAINE PROJECT

The DEFAINE (Design Exploration Framework based on AI for front-loaded Engineering) project provides an advanced framework for large-scale exploration of aircraft designs, data analyses, and artificial intelligence/machine learning capabilities, and consists of the following innovations⁴:

- The development of a knowledge-based engineering (KBE) methodology in order to model knowledge, accelerate design solutions, and provide adjustments during a design study.
- A methodology for the automatic formulation of multidisciplinary design analysis and optimization workflows.
- A smart, scalable, and cost-efficient computing infrastructure.
- A methodology based on automated data analysis techniques and machine learning algorithms to identify trends and relations in large sets of design data and support machine-driven knowledge discovery.

Figure 2.1 gives an overview of the DEFAINE project.

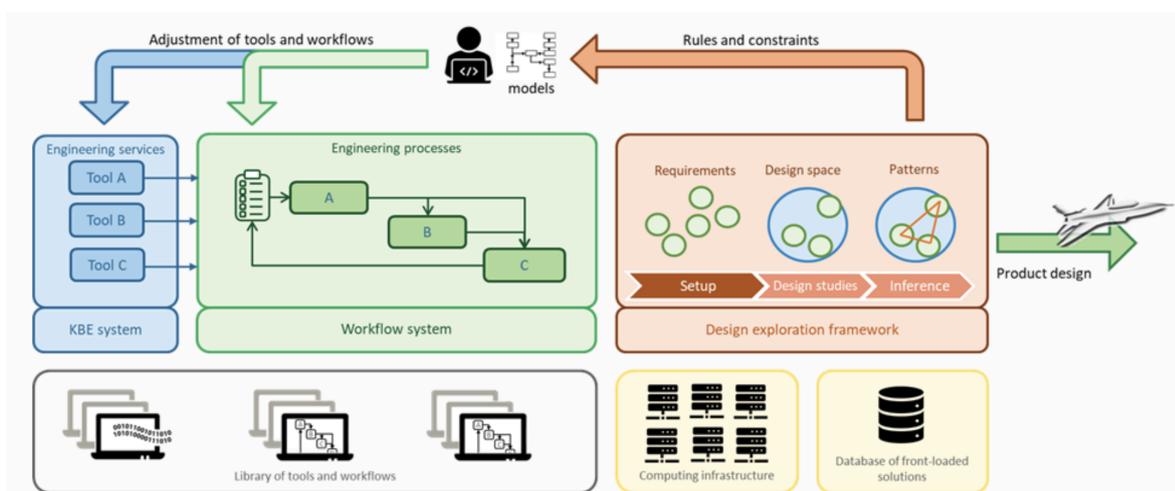


Figure 2.1: Overview of the framework for the DEFAINE project⁴.

³<https://www.defaine.eu/>, accessed: 17.08.2022

DEFAINE is being developed to address the demands of the aerospace market, such as to reduce recurring aircraft design costs by 10% and lead times for design updates by 50%³. For example, employing the KADMOS package within DEFAINE for the automatic reformulation of workflows reduces the set-up/execution time of design formulations. Simultaneously, increasing the number of advisory capabilities, like, for instance, the adaptation of MDAO problem formulations based on insights obtained from sensitivity analysis, as discussed later on in this thesis, will further enhance the set-up/execution time.

This thesis strives to contribute to the DEFAINE project in evaluating a new approach to include sensitivity information into the workflow formulation, for instance in reducing the number of design variables and in optimizing the sequencing and decomposition algorithms of KADMOS. This is explained in the introduction and Chapter 4. Referring to Figure 2.1, the contribution of this thesis is an approach to enhance the workflow system in the middle of Figure 2.1.

2.2. KADMOS

As this thesis strives to enhance the MDAO formulation process by using sensitivity data to identify non-influential design variables, but also to incorporate them into the sequencing and decomposition algorithms of KADMOS, KADMOS is extensively discussed in the subsequent chapter. KADMOS is an open-source Python software package. It allows to automate the generation of MDAO problem formulations through a graph based methodological approach [11]. Consequently, this enables to assemble, analyze, adjust, and reconfigure MDAO computational models easily and flexibly. An indepth and comprehensive formulation of the model is essential for the system, especially when the model contains a large number of coupled disciplines. Furthermore, in order to find the optimal design, the MDAO system requires a formal specification, then the solution strategy is determined, subsequently this formulation is translated into an executable workflow, and afterwards executed, such as described in the sections below.

The subsequent section provides a definition of the graph syntax and main graph classes utilized in KADMOS. A graph G contains a set of vertices V and a set of edges E : $G = (V, E)$, where an edge represents a connection between two vertices [11]. All the graphs in the KADMOS methodology are directed graphs. In directed graphs, the edge consists of a set of an ordered pair to indicate the edge direction. For undirected graphs, the edges show the connections between the nodes, but the starting and ending points of the edges are not defined. KADMOS uses two different types of directed graphs, as illustrated in Figure 2.2, data graphs and process graphs. A data graph, Figure 2.2a, only consists of function and variable nodes. Function nodes are executable blocks and variable nodes represent elements from the data schema. In addition, in the data graph the direction of the edges is depicted. Figure 2.2b shows the process graph consisting of only the function nodes and their execution order.

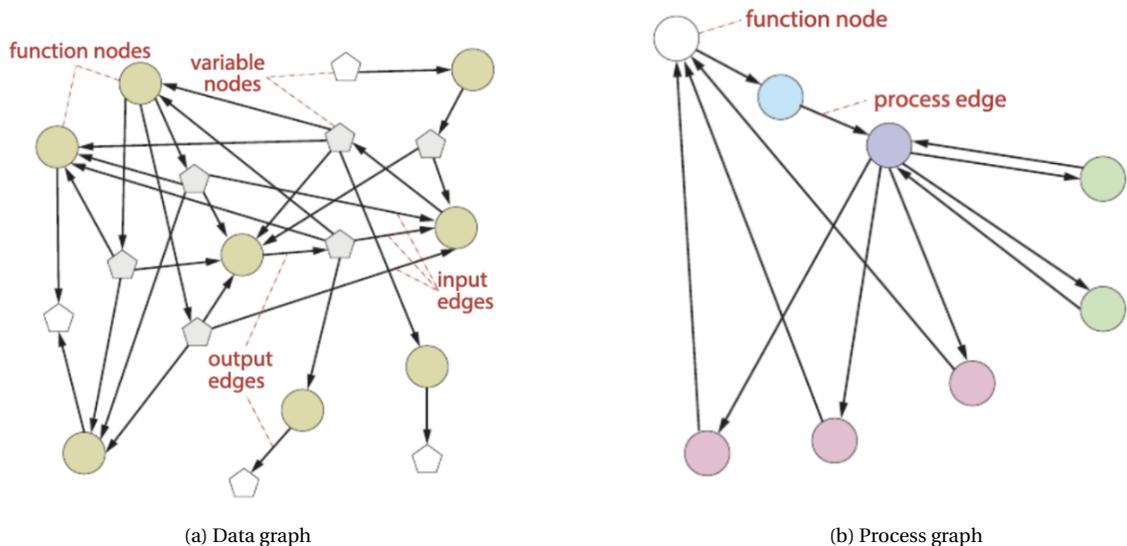


Figure 2.2: Example of the data and process graph used in KADMOS [11].

KADMOS enables the combination of data and process graphs to define an MDAO system and to facili-

tate a convenient visualization of the assembled MDAO system in the form of an eXtended Design Structure Matrix (XDSM) [15]. An example of an XDSM generated in KADMOS is shown in Figure 2.3. The executable components of the MDAO system are placed along the diagonal. Iterative components such as the optimizer and converger have an elliptical shape. Non-iterative components such as disciplinary tools and mathematical relations are represented as rectangles. The grey-colored parallelograms show couplings between components and white colored parallelograms show system inputs/outputs. The data flow is represented by the thick grey line, whereas the thin black line and the sequence numbers on each block indicate the process flow.

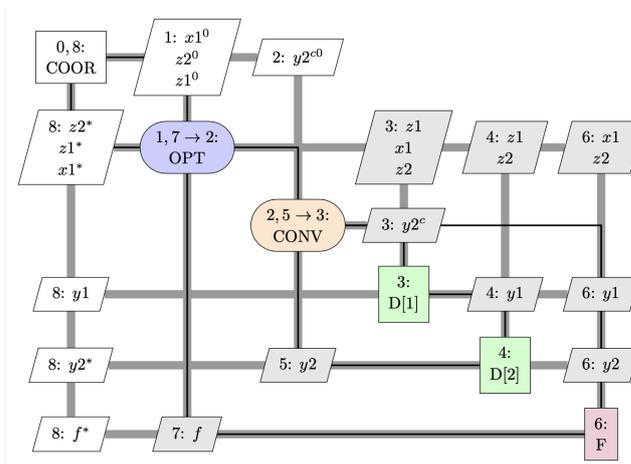


Figure 2.3: Example of an eXtended Design Structure Matrix (XDSM).

An overview of the full MDAO development process using KADMOS is illustrated in Figure 2.4. The first step is to create a tool repository where for each disciplinary tool an input and an output file is formulated according to a standard XML data schema. For example, the Common Parametric Aircraft Configuration Schema (CPACS) [16] can be used as an XML data schema definition for the description of an aircraft and its properties. CPACS represents a standard dictionary consisting of input and output data of disciplinary analysis tools in conceptual and preliminary aircraft design. Each disciplinary tool in the MDAO system uses a CPACS file as input and the resulting output is recorded in a new CPACS file. All accumulated corresponding CPACS files constitute the database that is subsequently used. This created database is then passed on to KADMOS using a CMDOWS file. CMDOWS is an XML schema and stores the MDAO workflow in a neutral format so that it can be used by the different applications, such as, for instance, MDAO problem formulation and MDAO solution strategy [17]. Subsequently, KADMOS evaluates the CPACS input and CPACS output files – if CPACS was used - and creates a Repository Connectivity Graph (RCG). The RCG is a directed graph and links all the inputs/outputs of the different tools.

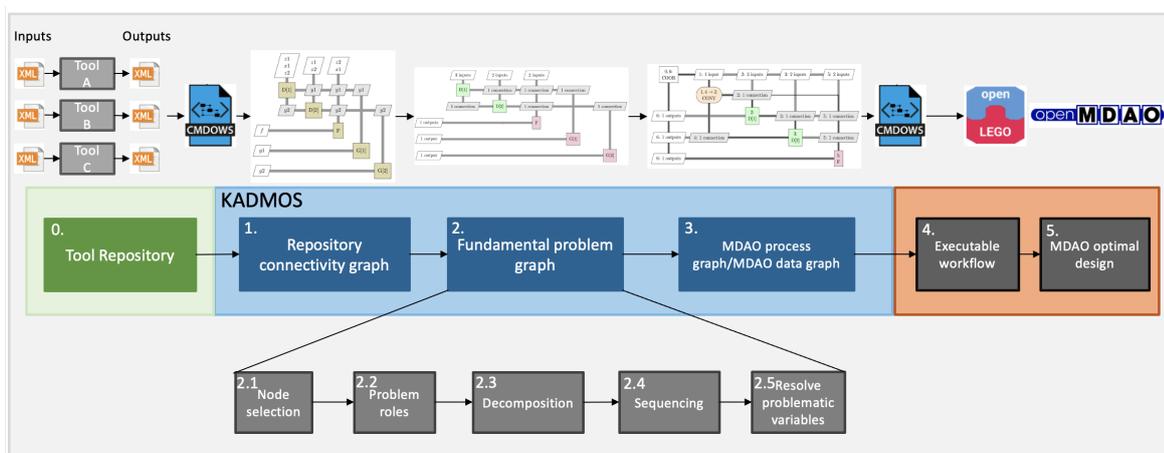


Figure 2.4: Overview of the full MDAO development process.

Using the RCG, a Fundamental Problem Graph (FPG) is generated. Various steps are performed to create the FPG, which are illustrated at the bottom of Figure 2.4. The FPG only contains necessary tools and variables required to solve the MDAO problem. Therefore, the first step of the FPG (2.1 in Figure 2.4) is to remove any redundant tools and variables. In the next step (2.2), each variable node is assigned to a specific problem role. The possible problem roles are design variable, constraint, objective, or quantity of interest. All outputs of the graph need to be assigned to a problem role, otherwise they are considered irrelevant outputs. In addition, the function nodes need to be categorized into pre-coupling, coupled, and post-coupling functions. Pre-coupling functions are the first functions in the analysis order and do not incorporate any data feedback, but only forward data. Coupled functions are defined as a group of functions that consist of having at minimum one circular dependency. Post-coupling functions are defined as functions that utilize the outputs of the pre-coupling and coupling functions in order to compute the MDAO model output.

The next steps of the FPG (2.3. and 2.4. in Figure 2.4) are sequencing and decomposition in order to formulate the execution process of the disciplines. An extensive review of the decomposition and sequencing process is provided in Section 2.2.2. In the last step of the FPG (2.5 in Figure 2.4) the graph validity is checked based on strict conditions and all problematic nodes are solved. If nodes in the FPG exhibit issues related to collisions and circularity, they are called problematic nodes and need to be resolved. Collision is defined when multiple disciplines have the same output variables. Circularity occurs when the same variable is used as an input and output within a discipline. Resolving collisions and circularity is achieved in three different ways: edge removal, node splitting, and node removal [11].

In the last step of KADMOS (step 3 of Figure 2.4), an MDAO architecture is imposed on the FPG and KADMOS creates an MDAO Data Graph (MDG) and an MDAO Process Graph (MPG). An XDSM is used to visualize the two graphs and they are also saved as a CMDOWS file. Also, the MDG depicts how the data flow among the different tools and contains all architectural elements, such as optimizers and convergers. The MPG consists of the execution sequence of all blocks in the architecture. There are various MDAO architectures that can be applied such as Individual Discipline Feasible (IDF), Multidisciplinary Feasible (MDF), Multidisciplinary Analysis (MDA), etc.

2.2.1. EXECUTION PHASE OF THE PROBLEM FORMULATION

Once the desired MDAO solution strategy, for instance, an MDF architecture, has been decided, the problem formulation needs to be executed by creating a workflow. The CMDOWS file of the MDG and MPG is then communicated to a workflow software which translates it to an executable model as depicted in Figure 2.4 (step 4). In this thesis, the software package OpenMDAO [18] is used to execute the workflow and to solve the optimization's problems. OpenMDAO is an open-source platform for system analysis and multidisciplinary design optimization written in Python. OpenMDAO is not able to directly process the CMDOWS file from KADMOS, thus the software package OpenLEGO (Open-source Link between AGILE and OpenMDAO) [19] is used as a link between KADMOS and OpenMDAO.

Another possible software is Remote Component Environment (RCE) [20] which translates the CMDOWS file through a native plugin. Bruggeman [14] performed a software comparison between OpenMDAO and RCE on a mathematical problem and concluded that the execution time in RCE is substantially greater than in OpenMDAO. In addition, the results demonstrated that both software derived the same objective function for all four architectures that were used.

2.2.2. SEQUENCING AND DECOMPOSITION

As mentioned before, during the generation of the FPG the execution process of the disciplines needs to be formulated. In the MDAO problem formulation phase, the sequencing and decomposition of the disciplines play a critical role in order to reduce the convergence time of MDAO systems, as the execution order determines the number of feedback connections. There are two properties that need to be considered: the execution time of a single iteration and the number of iterations. Subsequently, the total execution time t_{tot} can be calculated as shown in Equation 2.1:

$$t_{tot} = n_{iter} * t_{iter} \quad (2.1)$$

where n_{iter} is the number of iterations and t_{iter} is the execution time for a single iteration. Decreasing n_{iter} can be achieved by reducing the number of feedback connections, whereas t_{iter} can be reduced with decomposition.

SEQUENCING

Sequencing is defined as the determination of the execution order. For a small system containing a few disciplines and couplings, determining the execution order is a straightforward task. However, for large systems having numerous connections, finding the best function order is not obvious. The objective of the sequencing algorithms is to find the sequence that has the lowest number of feedback connections together with the shortest possible execution time of a single iteration for that number of feedback connections [14].

Figure 2.5 illustrates an example which contains four disciplines, where each discipline contains one output variable and the execution time for each discipline is also depicted in the figure. In Figure 2.5a an initial order is shown and contains three feedback variables. Disciplines A, B, and D can be executed simultaneously. Then C is started after discipline A provided the required input. The execution time of one iteration is equal to four seconds. Thus, the total time for this order is 12 seconds, as the execution time for one iteration is four seconds and assuming that the number of iterations is equal to three, where the 3 is taken as a numerical example. Figure 2.5b shows the order when the number of feedback connections is minimized. The system contains only a single feedback variable. Disciplines D and C are executed in parallel, followed by B and A. For a single iteration, the execution time is equal to nine seconds. Thus, the total time is nine seconds ($n_{iter} = 1$, $t_{iter} = 9$), assuming that now it only takes one iteration, where the 1 is taken as a numerical example. The order with minimum feedback reduces the total time by three seconds compared to the initial order.

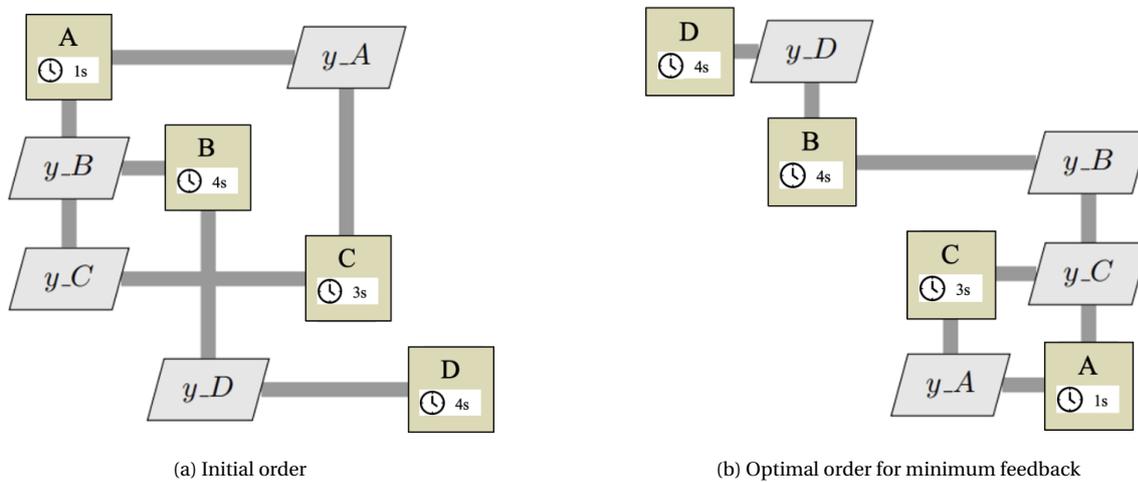


Figure 2.5: Example to explain sequencing algorithms.

Currently, there are various sequencing algorithms implemented within KADMOS [14]:

- *Branch-and-bound*: systematically searches the solution space by repeatedly dividing it into smaller subspaces using a tree structure.
- *Swapping*: is an iterative improvement method and aims to improve a given solution by searching for better solutions in the neighborhood. There are three different swapping methods:
 - Single-swap*: consists of moving a single node to another place in the execution order.
 - Two-swap*: consists of swapping the position of two nodes.
 - Hybrid-swap*: first uses the two-swap method until converged and then uses single-swaps to refine.
- *Brute-force*: consists of determining all possible sequences and inspects them separately.
- *Genetic algorithm*: different orders are created and evaluated based on their fitness, where better orders have a higher fitness. Solutions with higher fitness are more likely to produce offspring.

A more indepth explanation and validation/verification of all the algorithms is provided by Bruggeman [14].

DECOMPOSITION

In addition to the sequencing algorithms, a decomposition algorithm is also implemented in KADMOS. Decomposition is defined as the distribution of the disciplines over various partitions. The goal is to obtain the best decomposition of the system that balances the required coordination between partitions by minimizing the couplings between partitions and the execution time of each partition [14].

Figure 2.6 shows an example with three disciplines to illustrate decomposition. Figure 2.6a shows one partition, where all the disciplines are executed in sequence. If D1, D2, and D3 have an execution time of 4, 4, and 9 seconds, respectively, then the total execution time for a single iteration is 17 seconds. Figure 2.6b shows the example with two partitions. D1 and D2 form one partition, while D3 forms the second partition. As the two partitions can be performed in parallel, the total execution time for a single iteration is 9 seconds. The example demonstrates the importance and benefit of implementing decomposition. However, one also needs to consider the consequences of partitioning, as an increase in the number of partitions will cause a reduction in execution time per iteration but the complexity increases which will lead to a greater number of iterations. Thus, there is a trade-off between the complexity and the number of partitions.

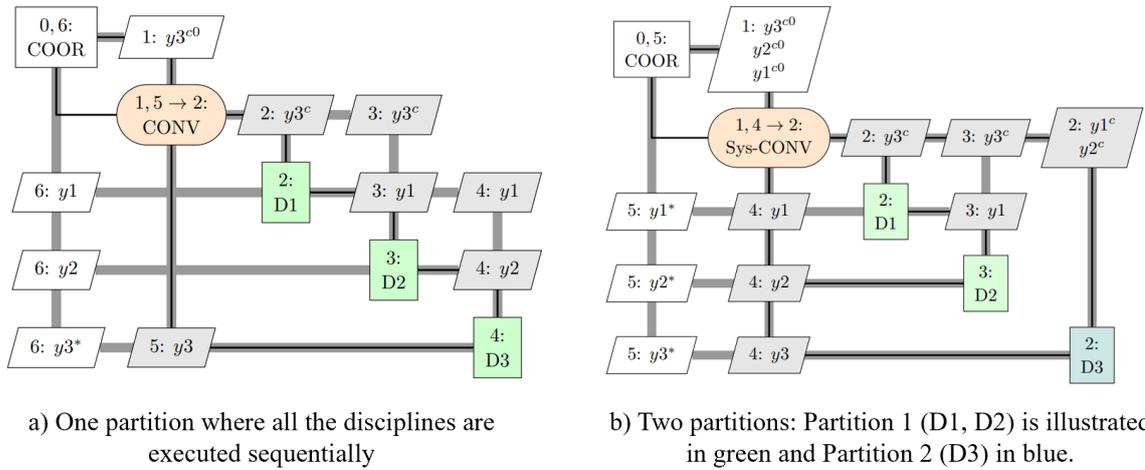


Figure 2.6: Example to illustrate the decomposition process.

KADMOS employs a decomposition method called the Metis-based Decomposition of KADMOS graphs (MDK). This method uses the Metis graph partitioning algorithm developed by Karypis and Kumar [21]. A detailed description of the steps for the MDK method is provided by Bruggeman [14].

In this chapter an overview of the DEFAINE project and the software package KADMOS was provided. The next chapter consists of a literature review about sensitivity analysis.

3

LITERATURE REVIEW

Chapter 3 provides a literature review and outlines the current state of research about sensitivity analysis. After the introduction section 3.1, which explains the basic definitions and concepts to use sensitivity analysis, the subsequent sections provide a detailed review of various sensitivity methods. Section 3.2 describes local and global sensitivity analysis, 3.3 One-At-a-Time and Many-At-a-Time sensitivity methods, 3.4 sampling strategies to obtain samples which contain values that are subsequently utilized in sensitivity analysis, 3.5 outlines the various sensitivity analysis methods which are used in this thesis, while 3.6 describes the software used for performing sensitivity analysis. The current research status about sensitivity analysis and its applications within the MDAO problem formulation in aircraft design is rather limited, available scientific literature is discussed in the subchapters.

3.1. INTRODUCTION

Designing an aerospace vehicle involves the consideration of an extensive number of input variables, derived from the mutual interference and/or interaction of multiple coupled disciplines which consequently results in a considerable computational burden. In order to reduce this computational cost, the most important variables of the formulated aircraft design problem can be pre-screened before a specific MDAO strategy is applied. This can be achieved with the use of sensitivity analysis methods. In this subsection, the basic definitions and concepts of sensitivity analysis are provided.

Sensitivity analysis allows to identify which parameters, or a set of parameters have the most influence on the model output. Saltelli [22] listed a set of reasons to use sensitivity analysis, as it allows to determine:

- The design variables that contribute the most to the output
- If and which design variables interact with each other
- Non-influential design variables

In addition, it is also important to understand what sensitivity analysis can achieve. In literature, the aim of applying sensitivity analysis is as follows:

- *Ranking*, also referred to as factor prioritization, comprises ordering the inputs in terms of their influence on the model output [23]. The inputs ranked the highest are the most influential parameters of the model. Information from this type of analysis may be essential to model improvement, as the most influential parameters become the focus of future evaluations. Figure 3.1a illustrates the effects of three variables (X_1 , X_2 , X_3) on the variance of output ($V(Y)$). $V(E(Y|X_i))$ shows the variance of Y when factor X_i is left to vary freely, and all other factors are fixed at their nominal values. In this case, factor X_2 contributes the most to the output Y , hence, should be prioritized. In addition, the sensitivity indices of the parameters can be ranked from highest to lowest as illustrated in the right plot of Figure 3.1a.
- *Screening*, also referred to as factor fixing, determines which model inputs have little or no influence on the model output. This enables to set these inputs to a fixed value. For aircraft design problems which

are comprised of computationally expensive models that have a large number of input parameters, screening methods can be applied to identify which of these input parameters have the lowest influence on the outputs [24]. Considering Figure 3.1a again, X_1 has the smallest effect to the variability of output Y , indicating that its value could be fixed in subsequent model executions. Eliminating such non-influential factors or fixing them to a constant value may aid to reduce model complexity and reduce unnecessary computational work in subsequent model runs.

- *Mapping* observes the effect of the inputs on the output. In Figure 3.1b, domain B of the output space Y represents the group of behavioral model outcomes. Domain \bar{B} represents the group of non-behavioral model outcomes resulting from the entire input space X . Mapping is concerned with the process of tracking which factor values in the input space X produce behavioral model outcomes in the output space [25].

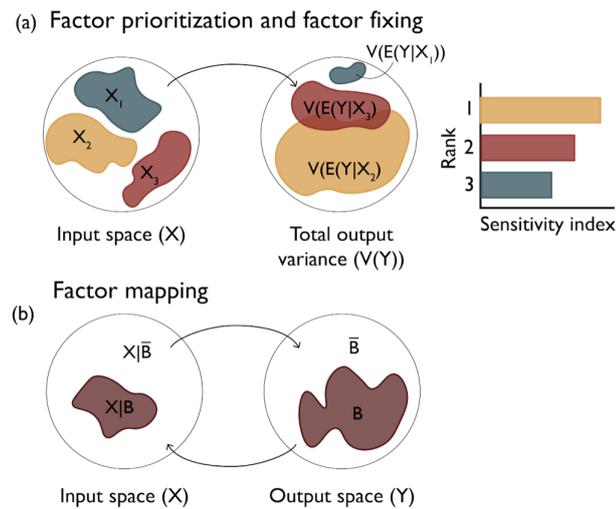


Figure 3.1: Ranking, screening, and mapping of sensitivity analysis [25].

A careful selection of the sensitivity analysis method must occur, as there is no suitable method comprising all of the three above aims. In this thesis, ranking and screening methods will be used. The next section explains two broad categories of sensitivity analysis: local and global sensitivity analysis methods.

3.2. LOCAL AND GLOBAL SENSITIVITY ANALYSIS

Sensitivity analysis methods are classified into two main groups [23]:

- *Local sensitivity analysis* methods evaluate how the outputs of a mathematical model or system change with respect to variations in a single input variable, thus, around one design point. The input variables are changed one by one at small increments and the effect of this individual input perturbation on the model output is determined by using local sensitivity indices.
- *Global sensitivity analysis* methods evaluate how the model output changes when all input variables are varied simultaneously over the entire input space. This allows to evaluate the contribution of each input variable to the model output and additionally the interactions between the inputs.

Due to the low computational cost and ease of implementation of local sensitivity methods, they are widely used in literature. However, they are only useful if the model is linear. Global methods are more suited for non-linear models, and they also give indications of each input's interactions with other input(s).

This distinction between local and global sensitivity methods is relevant for this thesis, as their different properties and areas of application permit their use for different purposes within this thesis. Global methods are used to adjust the MDAO problem formulation through the removal of non-influential design variables. Whereas local methods are applied to change the MDAO problem formulation by using sensitivity information within the sequencing and decomposition algorithms.

3.3. ONE-AT-A-TIME AND MANY-AT-A-TIME SENSITIVITY METHODS

Each sensitivity method requires to evaluate the model using different sets of input values. Thus, sensitivity methods can also be classified by counting how many input values' changes occur for each successive model simulation. The two categories, which can be either local or global sensitivity analysis methods, as outlined at the end of this chapter, are as follows:

- *One-At-a-Time (OAT)* methods change only one input per model evaluation.
- *Many-At-a-Time (MAT)* methods change two or more inputs simultaneously per model evaluation.

Most OAT methods assume a baseline set of input values for which the model is known to converge. Therefore, altering the value of an input is less likely to cause model evaluation to fail, for example due to instability or numerical errors. However, the OAT method has some serious limitations, most notably the difficulty of analyzing non-linear models [26]. Although widely used, OAT methods examine only a small fraction of the possible parameter space, especially with models containing large number of parameters [27]. For example, for a 12-parameter model, the OAT method examines less than one thousandth of the created parameter space [28]. Therefore, the simplicity of OAT methods comes at the expense of a more thorough understanding of the parameter space.

Local methods can only be OAT, whereas global methods can be either OAT or MAT. MAT-based methods are often more computationally expensive but are preferred for the analysis of non-linear functions because they explore a larger proportion of parameter space [26].

3.4. SAMPLING STRATEGIES

Often, sensitivity information cannot be computed analytically. Alternately, numerical approximations can be calculated, which require the generation of input samples and are called sampling. This section provides a short overview of common sampling strategies that are necessary for sampling-based methods. Sampling-based methods depend on the relationship between the sampled inputs and their corresponding outputs.

- *Random Sampling* obtains the values for the inputs with the use of a pseudorandom number generator. Random sampling does not store the history of previous points, so points may accumulate in some regions [29]. The problem with obtaining clustered samples is that the model approximation becomes less reliable in regions further away from this cluster. Therefore, it is recommended to utilize a sampling strategy that fills the factor space more evenly, such as Latin Hypercube Sampling (LHS).
- The *Latin Hypercube Sampling (LHS)* method is defined as dividing the input space into a number of equiprobable regions and then picking random samples from each region [30]. For instance, with a sample size of four, LHS would ensure that one point is in the 0 to 25th percentile range, one point between the 25th to 50th percentiles, one between the 50th to 75th percentiles, and one in the 75th to 100th percentile range. This ensures that the full range of values gets coverage and avoids random clumping. In addition, LHS also leads to a faster convergence than random sampling [30].
- The *Sobol Sequence* is a quasi-random method that generally results in more stable estimates compared to random sampling [31]. Quasi-random sequences guarantee that the samples fill the space more evenly than uncorrelated random samples [32]. As a result, quasi-random sequences need fewer model evaluations for convergence compared to random sampling [33].

3.5. REVIEW OF SENSITIVITY METHODS

In this section, a detailed review of the sensitivity analysis methods that are used in this thesis is provided. Table 3.1 shows a summary of the implemented methods in this thesis and provides information including whether they are global or local methods, detect interactions between factors, their capability to handle non-linearities in the model, their computational cost, and aim(s).

Table 3.1: Summary of sensitivity analysis methods covered in this thesis including whether they are global or local methods, detect interactions between factors, capability to handle non-linearities in the model, the computational cost, and aim(s).

Method	Type	Interactions	Non-linearities	Cost	Evaluation	Aim
Morris	Global	Yes	Yes	Medium	OAT	Screen
Sobol	Global	Yes	Yes	High	MAT	Rank, Screen
FAST/eFAST	Global	Yes	Yes	High	MAT	Rank, Screen
Local-derivative	Local	No	No	Low	OAT	Rank

3.5.1. MORRIS METHOD

The Morris method [34] is a global sensitivity method, screening method and is based on a repetition of a set of randomized OAT design experiments. The Morris method investigates how the output responds to a change in the inputs by varying one input at a time.

The steps for the Morris method are:

- The first step is to sample a set of starting values with a defined range for all input variables and then calculate the model output.
- The next step is to change the value of one input variable while all the other input variables remain at the starting values and then determine the resulting change in the model output compared to the model output that was obtained in the first step.
- Subsequently, the value of another input variable is changed while the previous input is kept at its previously changed value and the other input variables are kept at the starting values. This process is repeated until all input variables are changed.
- The Morris method approximates the derivative by using finite difference. With this method, each input is associated to an elementary effect EE_i , that is defined as the ratio between the variation in the model output and the variation in the model input. The elementary effect, EE_i of the i th input is as follows:

$$EE_i = \frac{f(X_1, X_2, \dots, X_i + \Delta, \dots, X_n) - f(X_1, X_2, \dots, X_i, \dots, X_n)}{\Delta} \quad (3.1)$$

where X_1, X_2, \dots, X_i are the model inputs and Δ is the step-size. Morris [34] recommended a value of $\Delta = k/(2(k-1))$, where k is the number of inputs, which ensures a good coverage of the input space with few trajectories.

- This procedure is repeated using different trajectory design r times where at each time a different set of starting value is used. Generally, r is chosen to be between 4 and 10 [35]. The number of model evaluations is determined by $r(k+1)$.

After obtaining the elementary effects, the sensitivity indices are determined as follows:

$$\mu_i^* = \frac{\sum_{k=1}^r |EE_i^k|}{r} \quad (3.2)$$

$$\sigma_i = \left[\frac{\sum_{k=1}^r (EE_i^k - \mu_i^*)^2}{r-1} \right]^{1/2} \quad (3.3)$$

where μ_j is the mean of the absolute value of the elementary effects and σ_j is the standard deviation of the elementary effects. The larger the value of μ_i^* , the more the i th input influences the model output. On the other hand, the higher the σ_i value, the more non-linear the input is or the more it interacts with other inputs, while a low σ_i indicates a linear, additive input.

The main advantage of the Morris method is the low computational expense as it requires only about one model evaluation for each elementary effect per replication. However, a disadvantage is that the Morris method is not able to distinguish between the effects caused by non-linearity and by the interactions with other input factors.

In literature, Brevault et al. [24] used the Morris method on a multidisciplinary design of a space launch vehicle and were able to identify non-influential design parameters. In addition, it was concluded that the Morris method is more robust to sample sizing and has a lower computational cost than compared to the Sobol method.

VISUALIZATION OF THE MORRIS METHOD

The Morris method has its own visualization. The Morris plot is the mean absolute value of the base effect μ^* plotted against the standard deviation σ of the base effect. This provides a visual representation of an input's ranking, which is shown on the μ axis. The interaction with other variables is shown on the σ axis. If a point is at the lower left corner of the plot, then that indicates that it has little effect on the model output and few interactions with other factors. Whereas, if a point is at the upper right corner of the plot, then it shows a large influence on the output and has presence of interactions with other factors. Figure 3.2 shows an example of the Morris plot for four inputs (X_1, X_2, X_3, X_4).

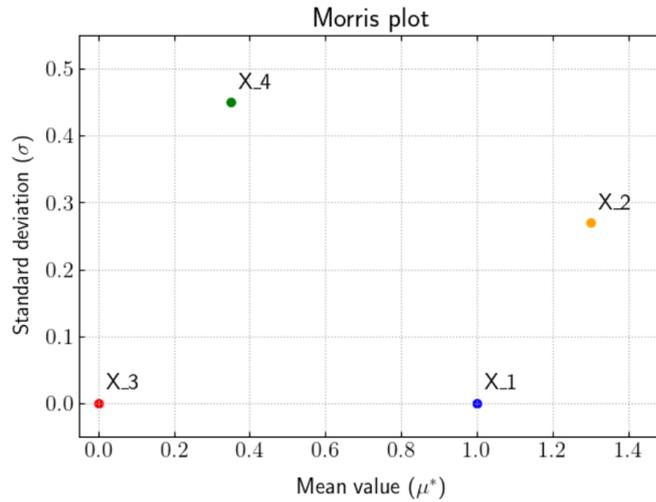


Figure 3.2: Example of the Morris plot.

3.5.2. SOBEL METHOD

In the following two sections, two commonly used ANalysis Of VAriance (ANOVA) methods are discussed, the Sobol and the FAST method. The ANOVA method is a global sensitivity method and is a variance decomposition method that consists in the decomposition of the variance of the output into a sum of contributions due to the input factors and their interactions [24], as shown in Figure 3.3. The results of the sensitivity analysis allow to rank the model inputs (factors), where the objective is to determine which factors are non-influential and may be set to a constant value without substantially affecting the results.

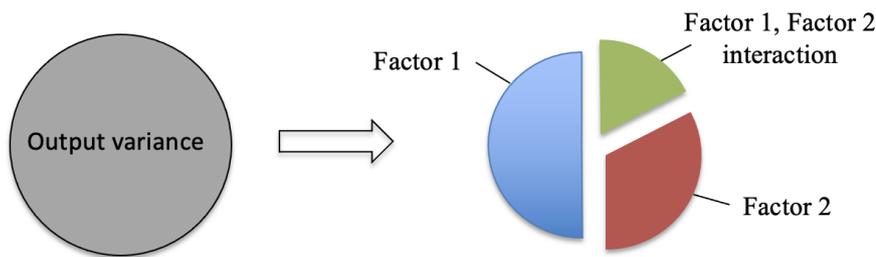


Figure 3.3: Apportioning output variance for ANOVA methods.

The first of these ANOVA methods, which is discussed subsequently, is the Sobol method [36, 37]. The Sobol sensitivity indices are described in terms of conditional variances [38]. For example, first-order index S_i measures the direct contribution of each input factor to the output variance and is defined as:

$$S_i = \frac{V_{X_i}(E_{X_{-i}}(Y|X_i))}{V(Y)} \quad (3.4)$$

where V is the variance, X_i are all model inputs, E is the expected value, and Y is the model output. Homma and Saltelli [39] introduced the total-order indices S_{Ti} , or known as total effects which allow to mea-

sure the part of output variance explained by all the effects in which it plays a role. The equation for S_{Ti} is:

$$S_{Ti} = \frac{E_{X_{\sim i}}(V_{X_i}(Y|X_{\sim i}))}{V(Y)} \quad (3.5)$$

An input is non-influential when the total-order index of that input is zero. Thus, S_{Ti} is used for factor screening. First-order indices are typically used for ranking, particularly when the interaction with other inputs does not have a significant effect on the output variance.

There are two major steps in order to compute the Sobol indices. The first step is to apply one of the sampling strategies discussed in Section 2.5.3. It was determined that quasi-random methods such as the Sobol Sequence are more efficient for calculating the total-order Sobol indices compared to random sampling or LHS [40]. The second step is to compute the Sobol indices, S_{Ti} and S_i by using the samples created from the sampling strategy. There exist several different algorithms to calculate these indices, however the Sobol's approach is most commonly used [40].

The Sobol method is used throughout literature and in the aerospace industry. Duler et al. [41] performed a sensitivity analysis based on the Sobol indices on a hybrid-electric aircraft powertrain and successfully identified non-influential performance parameters. Tremolet et al. [42] used the Sobol method on a multidisciplinary optimization of a blended wing body configurations and were able to determine the most influential design variables with regard to minimizing the fuel weight.

3.5.3. FAST AND eFAST METHOD

The Fourier Amplitude Sensitivity Test (FAST) is another ANOVA method. The theory of FAST was first introduced by Cukier et al. [43]. FAST uses a periodic sampling approach and Fourier series expansions to decompose the variance of a model output into partial variances, contributed by different model parameters [44]. FAST is more efficient to calculate the sensitivities compared to other variance-based global sensitivity analysis methods, such as the Sobol method, as it computes all indices in one set of model evaluations [26].

An improved method developed from FAST is the extended-Fourier Amplitude Sensitivity Test (e-FAST) [45]. The advantage of e-FAST over the classical FAST is the ability to compute both the first order and the total order sensitivity of each input parameter.

The main advantage of variance-based sensitivity analyses, such as FAST and Sobol methods, is that they are easy to implement and interpret. However, the computational cost to evaluate all the sensitivity indices is excessively high, as the number of indices grows in an exponential way with the number of input variables (e.g. $2^k - 1$, where k is the number of input variables) [46]. This leads to a drawback that in order to obtain sufficient accuracy of the sensitivity indices, a large number of model evaluations is required. In the case of complex models, such as an aircraft design, where the number of inputs is large, using quantitative methods could become unfeasible. An alternative method to overcome this problem are the screening methods, such as the Morris method.

Park et al. [47] used the e-FAST method to determine the global sensitivities on an aero-structural design optimization of a light-jet aircraft wing. The MDAO problem was formulated with two disciplines: an aerodynamic solver using the panel method and a structural solver using a finite element ANSYS model and the objective function was to maximize the lift-to-drag ratio. Utilizing the e-FAST method successfully reduced the number of design variables which lead to a reduction of the number of iterations by 33.55%, compared to with the case when considering all design variables.

VISUALIZATION OF THE ANOVA METHODS

The sensitivity indices for the ANOVA methods are commonly visualized with a barplot but could also be presented in a table. Figure 3.4 shows an example of a double barplot for the total-order (S_{Ti}) and first-order (S_i) Sobol sensitivity indices.

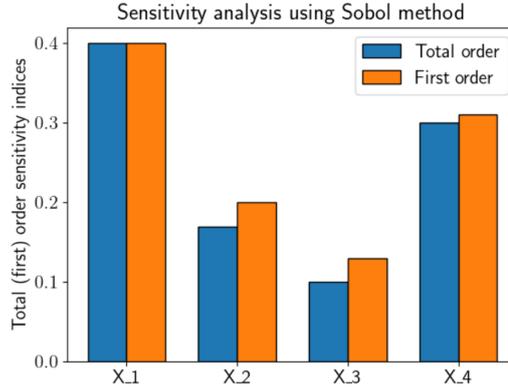


Figure 3.4: Example of the visualization of the sensitivity indices for the ANOVA methods.

3.5.4. DERIVATIVE-BASED LOCAL SENSITIVITIES

The second category of sensitivity analyses are the local sensitivity methods. As stated before, these methods evaluate how the model output changes relative to the change of a single parameter value. The simplest approach to compute sensitivities is with the finite-difference method. The equation for the finite difference is derived by truncating a Taylor Series expanded over a given point x . For instance, for a function f with one variable x (Figure 3.5), the forward-difference approximation of the first derivative is derived by expanding $f(x+h)$,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \dots \quad (3.6)$$

Solving for the first derivative (f'), the finite difference equation is obtained

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \vartheta(h) \quad (3.7)$$

where h is the finite-difference interval and $\vartheta(h)$ is the truncation error. The second order approximation is obtained by using the expansion of $f(x-h)$,

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \dots \quad (3.8)$$

and then subtracting it from Equation 3.6. The resulting derivative of f' , results in the central-difference formula,

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \vartheta(h^2) \quad (3.9)$$

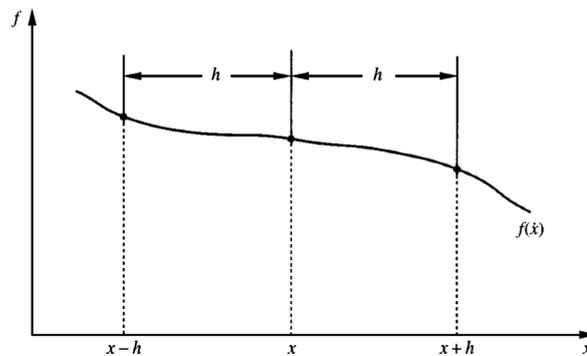


Figure 3.5: Finite-difference intervals [48].

When the function has several variables, x becomes a vector, and the application of Equation 3.7 will require the calculation of each component of the gradient $\nabla f(x)$ by perturbing the corresponding variable. This leads to a high cost for calculating the sensitivities using the finite-difference method for large scale problems, as it is proportional to the number of design variables and f needs to be computed for each perturbation of x_i [49].

Another main disadvantage of the finite-difference method is the level of accuracy of its sensitivity results which is influenced by the step size. The choice of the step size contributes to three different types of errors: truncation error, condition error, and subtractive cancellation [50]. The truncation error is defined as the difference between the exact result and the computed result of a perturbed function. The truncation error is the neglected terms in the finite-difference equation ($\vartheta(h)$) and amplifies with increased step size. The condition error is related to numerical noise and generally decreases with increased step size. With decreasing values of the finite-difference interval h , the perturbed value will match the reference value and will lead to subtractive cancellation. For very small values of h , all the digits of the perturbed value will be equal to the reference value, leading to a derivative of zero. The choice of step size encounters the dilemma that it is favorable to select a small step size to minimize the truncation error but at the same time avoiding using such a small step size that the subtractive cancellation becomes dominant.

Currently, finite-difference methods are widely used due to ease of implementation and as they allow to calculate the sensitivities when the computational model is a black box. The main limitation is that they evaluate the parameters one at a time and do not allow for the evaluation of simultaneous changes in all model parameters. Furthermore, the interaction between parameters cannot be evaluated using a local sensitivity analysis. Thus, local sensitivity analysis is more suited to determine the sensitivities on a single discipline. Therefore, for MDAO systems with various coupled disciplines global sensitivity methods should be used.

3.6. SOFTWARE FOR PERFORMING SENSITIVITY ANALYSIS

In conclusion, three global sensitivity methods were implemented in this thesis: Sobol, FAST, and Morris methods and one local sensitivity method: the finite-difference method. The three global methods are well established in literature; thus, an external software package called SALib (Sensitivity Analysis Library in Python) [51, 52] was utilized. SALib contains Python implementations of commonly used sensitivity analysis methods, including Sobol, FAST, and Morris methods. SALib allows to have a decoupled workflow, as SALib does not directly interface with the computational model. A typical sensitivity analysis using SALib consists of the following steps:

1. Determine the inputs of the model and their bounds.
2. Run a sampling strategy to generate model inputs.
3. Evaluate the model using the generated input and save the model output.
4. Run a sensitivity method on the outputs to calculate the sensitivity indices.

4

METHODOLOGY

The main goal of this thesis is to implement sensitivity information, obtained by conducting sensitivity analysis, into the MDAO problem formulation which subsequently is adjusted and optimized. Furthermore, it will be evaluated if considering sensitivity information during MDAO formulation leads to a time reduction for MDAO execution. Two independent methodologies were developed to perform these investigations. The aim of the first methodology is to identify design variables that have a non-influential effect on the objective by utilizing global sensitivity methods, as they allow to characterize the input variables on the entire input space. The goal of the second methodology is to use a local sensitivity method to improve the sequencing and decomposition algorithms because local sensitivity methods only characterize around a single point.

4.1. METHODOLOGY 1: IDENTIFY NON-INFLUENTIAL DESIGN VARIABLES

An overview of the steps performed in the first methodology is illustrated in Figure 4.1. The MDAO workflow is divided into four main phases: initial formulation, formulation analysis, formulation adjustment, and execution. A detailed review of each phase is provided in the subsequent sub-sections. The development of these steps does not provide an indicative time breakdown of the whole process, respectively some time estimation for certain steps/phases. These will be obtained later on during the case studies when a specific design is being evaluated.

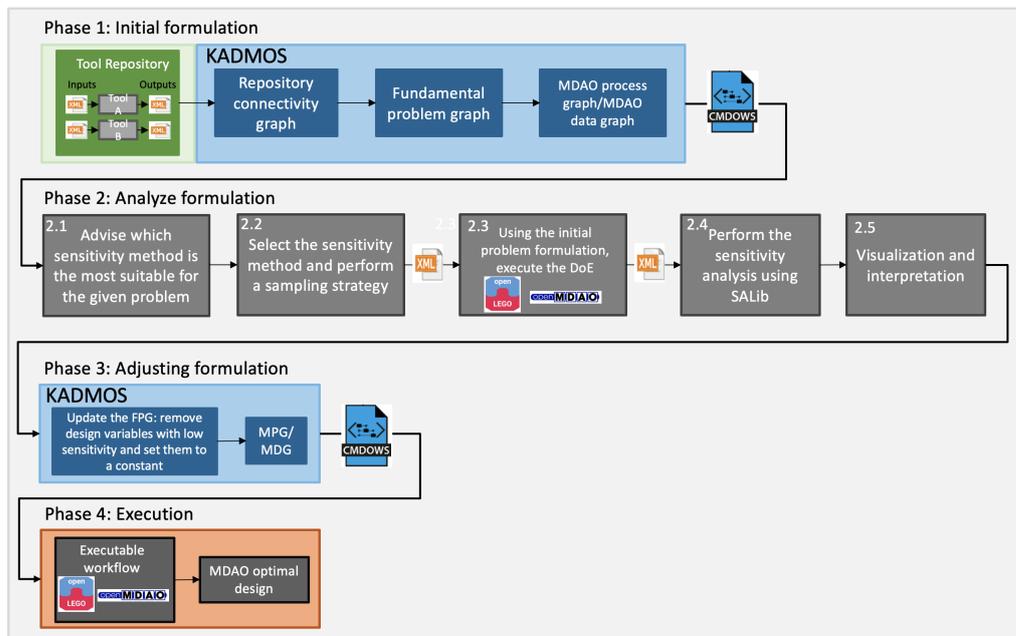


Figure 4.1: Overview of the steps performed in the first methodology where non-influential design variables are identified.

4.1.1. PHASE 1: INITIAL FORMULATION

The first phase consists of creating an initial MDAO problem formulation using KADMOS. The generation of the MDAO formulation follows the same steps as described in Section 2.2 and creates the KADMOS graphs: RCG, FPG, and MDG/MPG. For the MDG/MPG an MDA architecture is applied, as later in the second phase the problem needs to be solved numerous times in order to perform the sensitivity analysis.

4.1.2. PHASE 2: ANALYZE PROBLEM

The second phase consists of analyzing the problem with the use of global sensitivity analysis. In order to achieve this, a sensitivity analysis tool was developed in Python which uses the library SALib for the Morris, Sobol, and FAST sensitivity methods. In Figure 4.1, the first step of the sensitivity analysis tool (2.1.) is to advise the user which sensitivity method is the most suitable for the given problem. The tool contains functions that print a detailed description on how each method works and what requirements are necessary. In addition, each method has different characteristics in terms of computational cost and accuracy. Thus, an advisory capability is implemented which calculates the number of required problem evaluations. Table 4.1 shows an overview of how the required number of runs is calculated and the accuracy of each method.

Table 4.1: Overview of the required number of problem evaluations for each sensitivity method, where k is the number of design variables, n is the minimum number of model evaluations for estimating one individual effect, N_s is the number of samples per parameter, N_R is the re-sampling size, and r is the number of trajectories.

Method	Required number of problem evaluations [34, 53, 54]	Accuracy
Sobol	$n(2k + 2)$	high
FAST	$N_s * k * N_R$	highest
Morris	$r(k + 1)$	least

In Table 4.1 k is the number of design variables. The smallest sample size for the Sobol method depends on the minimum number of model evaluations for estimating one individual effect (n) and takes the value of 16, 32, or 64 [53]. The number of model evaluations for the FAST method depends on the number of samples per parameter (N_s) and the re-sampling size (N_R). The minimum value for N_s is 65 [54]. The smallest sample size for the Morris method depends on the number of trajectories (r) and is usually taken between 5 and 15 [34]. For instance, a problem with 10 design variables would require 352 model evaluations using the Sobol method ($n = 16$, $k = 10$). Hence, if a single model evaluation takes a long time, then obtaining the sensitivity information is a costly process.

For each method there is a trade-off between computational cost and accuracy. The FAST method is the most accurate, however, it requires significant more numbers of problem evaluations. The Morris method is the least computational expensive, however, it also has the least accuracy. In general, for all sensitivity methods the number of model evaluations increases with the number of design variables. The developed sensitivity analysis tool outputs such a table with the computed number of required problem evaluations for the given problem. However, the user still needs to choose which sensitivity method to use, as it is very problem dependent.

After the user determines which sensitivity method to use, the next step in the workflow (2.2. in Figure 4.1) is to apply a sampling strategy. All implemented sensitivity methods are sampling-based methods, which rely on the relationship between the sampled inputs and their corresponding outputs. Each sensitivity technique requires the computation of the quantity of interest of the model using different sets of input values. An overview of the sampling strategies for each method is shown in Table 4.2 and was described in Section 3.4.

Table 4.2: Overview of the sampling method for each sensitivity method.

Method	Sampling method
Sobol	Sobol Sequence
FAST	Latin Hypercube Sampling (LHS)
Morris	Latin Hypercube Sampling (LHS)

The computation of the Sobol sensitivity indices needs the input sample generated by the Sobol sequence [37]. Helton et al. [55] compared the sensitivity analysis results obtained by different sampling strategies and

they recommended the LHS compared to the random sampling for the analysis of complex systems with small sample sizes. Thus, both the FAST and Morris methods will use the LHS.

A requirement for the sampling methods is the number of design variables and their lower and upper bounds which are received from the MDG/MPG CMDOWS file in the first phase of the workflow. The sensitivity analysis tool then automatically creates the sampling and generates an XML file containing the input values for every problem evaluation.

The next step (2.3. in Figure 4.1) is to execute and solve the problem. Utilizing the OpenLEGO and OpenMDAO, the problem is solved as a Multidisciplinary Analysis (MDA) with the generated input files and the outputs saved to an XML file.

Using the set of inputs and the corresponding outputs allows to perform the sensitivity analysis and calculate the sensitivity indices (2.4. in Figure 4.1). The sensitivity analysis is performed automatically and does not require any input from the user.

The last step (2.5. in Figure 4.1) is to generate visualizations of the results in the form of a Morris plot (Section 3.5.1) or barplot (Section 3.5.3) and to interpret the results. For the Sobol and FAST methods it applies that if a design variable has a total sensitivity index of less than 1% or 5% of the variance in the output, then that variable is non-influential, according to advice from literature [56]. For the Morris method there is no fixed threshold value; however, a threshold is usually set as 5-10% of the maximum μ^* value [56].

4.1.3. PHASE 3: FORMULATION ADJUSTMENT AND PHASE 4: EXECUTION

In the third phase of the first methodology, the MDAO problem formulation is adjusted according to the sensitivity information obtained in the previous phase. The FPG is updated such that the problem role of the non-influential design variables is removed and set to a constant value. In phase 4 the workflow is executed, and a full optimization is performed with OpenMDAO.

4.2. METHODOLOGY 2: SEQUENCING AND DECOMPOSITION WITH USE OF SENSITIVITY INFORMATION

As stated before, the second methodology was developed with the aim to improve the existing sequencing and decomposition algorithms in KADMOS. As mentioned in Section 2.2.2, the original goal of sequencing was to find the lowest number of feedback variables and the shortest possible execution time of a single iteration for that number of feedback variables. Instead of using the number of feedback variables, local sensitivity information can be used. The hypothesis is that if the sensitivity in the feedback is lower, then the expectation is that the number of iterations is reduced. Fewer iterations will lead to a lower execution time. For instance, a sequence with two feedback variables that both have a low sensitivity can have a lower number of iterations than a sequence with one variable that has a high sensitivity. Thus, the goal is to determine a sequence where the variables with a low sensitivity are placed in the feedback. In order to further explain this concept, consider the example illustrated in Figure 4.2. The example consists of four disciplines (A, B, C, D) where each discipline has a single output, and the corresponding sensitivity information is illustrated.

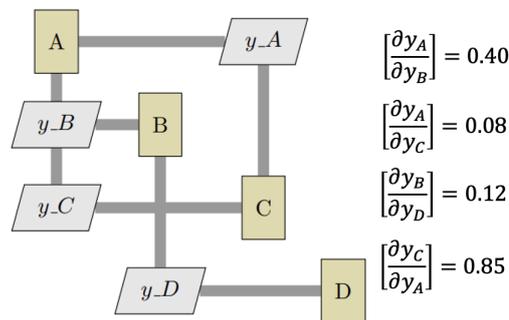


Figure 4.2: Example problem consisting of four disciplines with the corresponding sensitivity information.

The left-hand side of Figure 4.3 shows the solution for the minimum number of feedback and has only a single feedback connection (y_A). The right-hand side of Figure 4.3 depicts the sequence where the variables that have the lowest sensitivity are placed as feedback connections ($\frac{\partial y_A}{\partial y_C}$, $\frac{\partial y_B}{\partial y_D}$). $\frac{\partial y_C}{\partial y_A}$, has the highest sensitivity

and becomes a feedforward variable. The improved sequence may require less iterations, thus be a more computational efficient sequence, even though it now has two feedback variables instead of one. A downside is that the figure on the left will have a shorter execution time per iteration than the figure on the right. A shorter runtime can lead to a shorter execution time even when the number of iterations is increased.

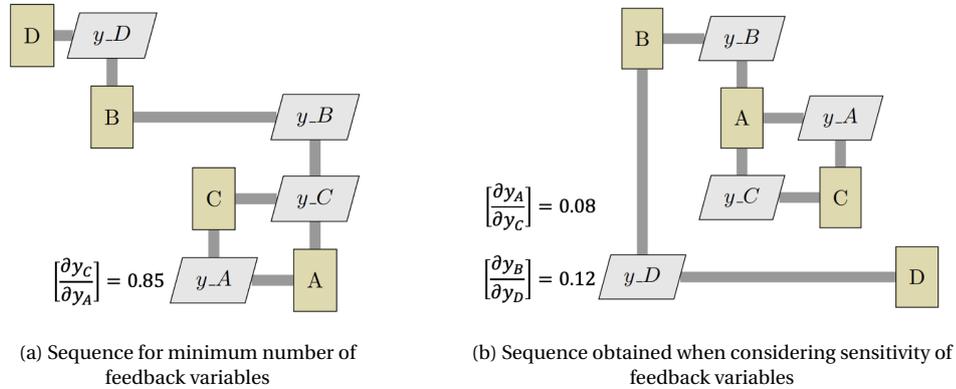


Figure 4.3: Comparison of the sequencing solution for the two different methods.

Incorporating sensitivity information can also be beneficial in the decomposition process. The goal is to gather disciplinary tools with strong sensitivities in the same partition and at the same time to collect disciplinary tools with low sensitivities in a different partition. This allows to minimize the dependency between the various partitions which again can lead to a reduction in iterations.

An overview of methodology 2 is illustrated in Figure 4.4. When creating the FPG two new steps are introduced: 2.3: perform local sensitivity analysis and 2.4: create a coupling dictionary. A detailed description of the two new steps is given in the following sub-sections.

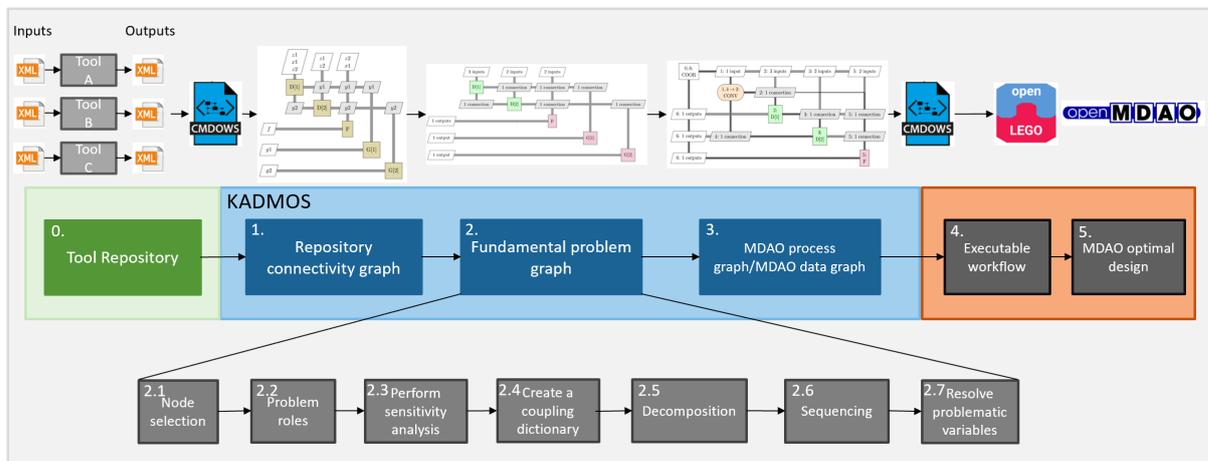


Figure 4.4: Overview of the workflow for implementing sensitivity analysis within the sequencing and decomposition algorithms.

4.2.1. PERFORMING SENSITIVITY ANALYSIS

The main difference to the previously described methodology is that the sensitivity analysis is performed on a local level. Instead of investigating how influential the design variables are on the whole system, here a sensitivity analysis on each tool is performed. This allows to obtain information to what extend each input of the tool affects the output of the tool. A finite-difference sensitivity method is used to generate the sensitivity information and allows to calculate the partial derivative of the output with respect to the input. The finite-difference method was chosen due to the ease of implementation and the ability to compute the sensitivities when the disciplinary tool is a black box.

4.2.2. CREATING A COUPLING DICTIONARY

The existing sequencing and decomposition algorithms in KADMOS require a coupling dictionary as an input to determine the number of feedback variables or to determine the sensitivity of the feedback connections. The coupling dictionary indicates from which function nodes the input and the number of variables is obtained. Using the example problem (Figure 4.2), Figure 4.5a shows the original coupling dictionary. For instance, discipline A has one coupling variable from discipline B and one from discipline C. Instead of using the number of coupling variables, the modified coupling dictionary contains the sensitivity information and is shown in Figure 4.5b.

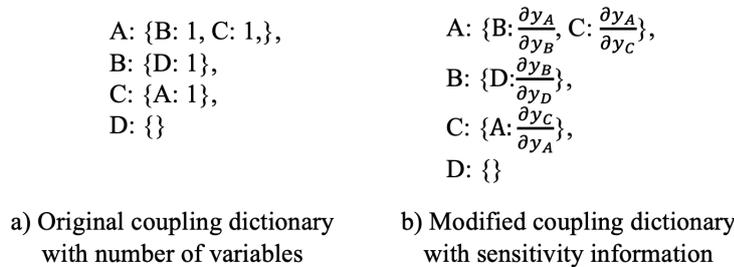


Figure 4.5: Coupling dictionary for the example problem as shown in Figure 4.2.

The modified coupling dictionary is then passed as an input to the existing sequencing and decomposition algorithms. As discussed in Section 2.2.2 there are various sequencing algorithms. To further illustrate how the sensitivity information is used, a branch-and-bound algorithm is applied to the example problem (Figure 4.2) and is shown in Figure 4.6. This algorithm systematically explores the solution space by repeatedly dividing it into smaller subspaces using a tree structure. In the first step, the solution space is split into four branches by fixing the first node of the sequence. For each branch, the bound is calculated by determining the sensitivity value and the execution time. For instance, discipline A obtains an input from discipline B and C whose sensitivities are 0.40 and 0.08 respectively. If A is placed first, the sensitivities are added together and the bound becomes (0.48, 1). The 1 in (0.48, 1) stands for the execution time of discipline A. After computing the branches, it was determined that the branch with D placed first is the most promising, as it has the lowest sensitivity value. Therefore, this branch is explored further, and the bounds are again calculated. The most promising branch is selected, and the process is repeated until a global optimum is determined.

After performing the algorithm, it was determined that the sequence D, B, A, and C has the lowest sensitivity in the feedback. The sequence is shown in Figure 4.7. The solution contains only a single feedback with the lowest sensitivity.

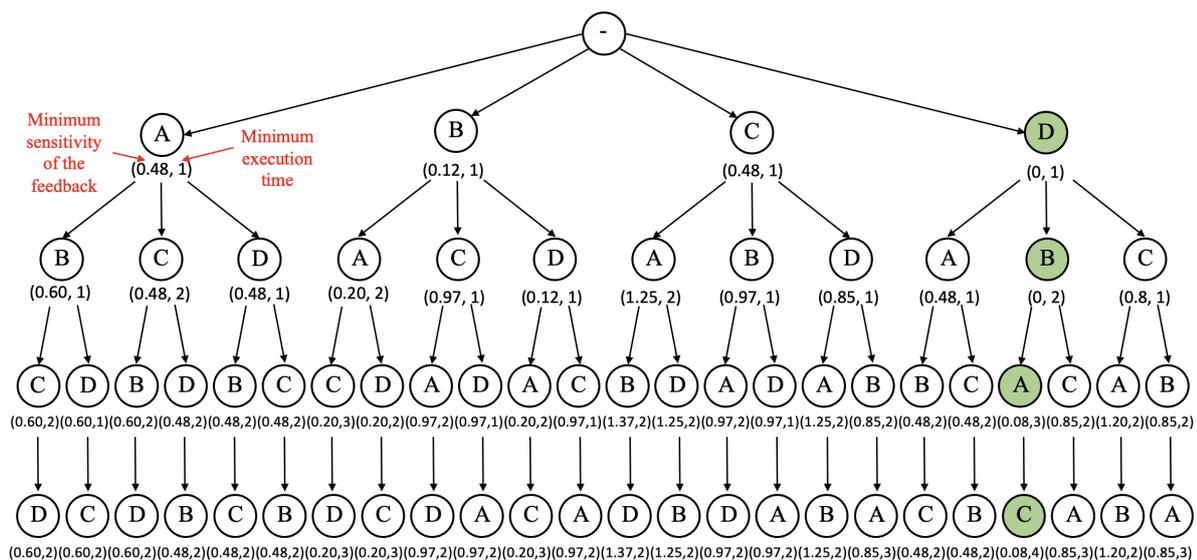


Figure 4.6: Overview of the branch-and-bound algorithm applied to the example problem when using sensitivity information.

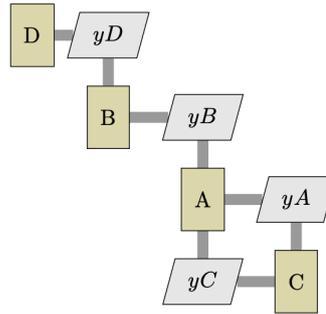


Figure 4.7: Final sequence obtained by the branch-and-bound algorithm using sensitivity information.

In this chapter, a detailed description of the two implemented methodologies was given. In the next chapter, the methodologies are verified and validated with the use of mathematical problems.

5

VERIFICATION & VALIDATION

After the implementation of the two methodologies, they were verified and validated. Verification is a quality control process and used to evaluate whether the developed methodologies are suitable for the research topic; while validation is a quality assurance process and proves that the outcomes are true and are based on strong evidence.

5.1. VERIFICATION AND VALIDATION OF THE SENSITIVITY METHODS

The implemented global sensitivity methods were verified and validated on a mathematical function. The Ishigami function is commonly used as a test function to benchmark, as it exhibits strong non-linearity and non-monotonicity [57]. The function is defined as follows:

$$f(x) = \sin(x_1) + a \sin^2(x_2) + b x_3^4 \sin(x_1) \quad (5.1)$$

where x_j are the input factors that are uniformly distributed: $-\pi \leq x_j \leq \pi$, $j = 1, 2, 3$. The variables a and b are set to constant values of 7.0 and 0.1 respectively. The first-order and total-order sensitivity indices can be calculated analytically. Table 5.1 and 5.2 show the comparison between the analytical values and the calculated sensitivity indices from the Sobol and FAST method, respectively. The main peculiarity of the function is the dependence on x_3 , as the first-order sensitivity index is equal to zero, but the total-order cannot be neglected. All of the obtained results agree with the analytical values. The greatest percentage difference is obtained for the total-order index of x_3 for both methods. Overall, the FAST method is more accurate than the Sobol method. In addition, the Morris method was also tested on the Ishigami function, and the results are shown in Table 5.3. As the Morris method is a screening method and not a ranking method, the results cannot be compared directly to the analytical values. However, μ^* and the total-order index quantify the same information as both are used for factor screening. Comparing the results of the Morris method and the analytical total-order indices, the parameters (x_1 , x_2 , and x_3) are ordered in the same way.

Table 5.1: Comparison between the Sobol sensitivity indices and the analytical values.

Parameter	First-order index analytical	First-order Sobol index	Total-order index analytical	Total-order Sobol index
x_1	0.3138	0.3104 (1.08%)	0.5574	0.5556 (2.67%)
x_2	0.4424	0.4437 (0.29%)	0.4424	0.4535 (2.47%)
x_3	0.0	0.0122 (-%)	0.2436	0.2506 (2.83%)

Table 5.2: Comparison between the FAST sensitivity indices and the analytical values.

Parameter	First-order index analytical	First-order FAST index	Total-order index analytical	Total-order FAST index
x_1	0.3138	0.3168 (0.95%)	0.5574	0.5558 (0.29%)
x_2	0.4424	0.4426 (0.05%)	0.4424	0.4419 (0.11%)
x_3	0.0	0.0 (0%)	0.2436	0.2392 (1.82%)

Table 5.3: Results of the Morris method for the Ishigami function.

Parameter	μ^*	σ
x_1	7.8750	7.8747
x_2	7.6703	6.2404
x_3	6.2406	8.8306

5.2. VERIFICATION AND VALIDATION OF THE SEQUENCING AND DECOMPOSITION PROCESS

In order to test the utilization of sensitivity information for the sequencing and decomposition process, it was tested on a scalable mathematical problem, called the variable complexity problem [58]. The variable complexity problem allows to generate numerous MDAO problems with arbitrary complexity by specifying its changeable parameters. The changeable parameters in the variable complexity problem are [58]:

- the number of disciplines
- the number of global variables
- the number of local variables
- the number of coupling variables
- the number of local constraints
- the number of global constraints
- the execution time of the disciplines

The optimization problem of the variable complexity problem is defined as follows:

$$\begin{aligned}
 & \text{minimize} && f(\mathbf{x}_0, \mathbf{x}, \mathbf{y}) \\
 & \text{with respect to} && \mathbf{x}_0, \mathbf{x} \\
 & \text{subject to} && c(\mathbf{x}_0, \mathbf{x}, \mathbf{y}) \leq 0
 \end{aligned} \tag{5.2}$$

where f is the objective that is minimized, \mathbf{x}_0 are the global design variables, \mathbf{x} are the local design variables, \mathbf{y} are the coupling variables, and \mathbf{c} are the constraint values. An equation is created to represent the analysis of discipline i in the variable complexity problem as follows:

$$\mathbf{B}_{ii}\mathbf{y}_i = -\mathbf{C}_i\mathbf{x}_0 - \mathbf{D}_i\mathbf{x}_i - \sum_{j=1, j \neq i}^N \mathbf{B}_{ij}\mathbf{y}_j \tag{5.3}$$

where \mathbf{B}_{ij} , \mathbf{C}_i , and \mathbf{D}_i are a coefficient matrix. The \mathbf{B}_{ij} matrix shows how each coupling variable influences the other coupling variables. The \mathbf{C}_i and \mathbf{D}_i matrices show the influence of the global and local design variables on the coupling variables. The values for the \mathbf{B}_{ij} , \mathbf{C}_i , and \mathbf{D}_i matrices are generated randomly.

The objective of the variable complexity problem is defined as:

$$f = \left(\sum_{i=0}^{n_{x0}} \mathbf{x}_{0i} + \sum_{i=1}^N \sum_{j=1}^{n_{xi}} \mathbf{x}_{ij} + \sum_{i=1}^N \sum_{j=1}^{n_{yi}} \mathbf{y}_{ij} \right)^3 \tag{5.4}$$

The second developed methodology (Section 4.2) was tested on a variable complexity problem with the chosen parameters shown in Table 5.4. In addition, Figure 5.1 shows the repository connectivity graph that contains all the connections for the given problem.

Table 5.4: Chosen parameters for the variable complexity problem.

Number of disciplines	5
Number of global design variables	1
Number of local design variables	[1, 2, 2, 1, 3]
Number of coupling variables	[2, 1, 3, 1, 4]

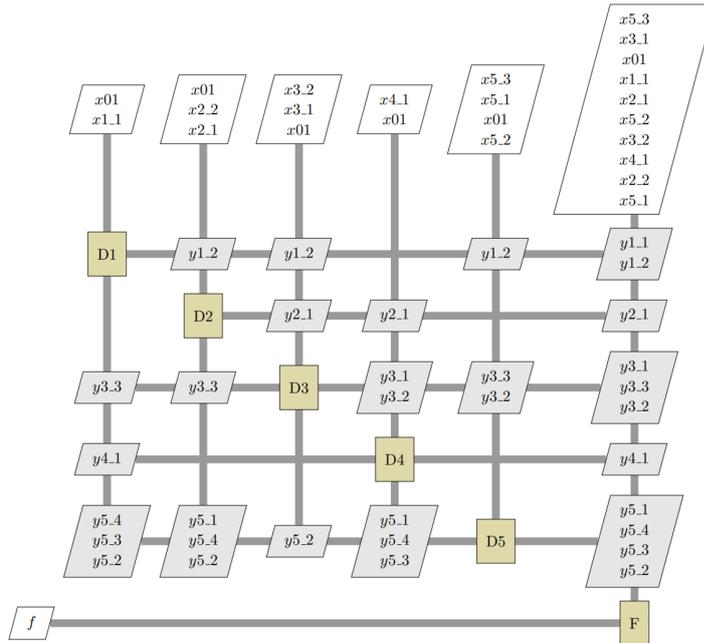


Figure 5.1: The repository connectivity graph of a variable complexity problem.

A local sensitivity analysis using the finite-difference method was performed on the problem and Table 5.5 shows the sensitivity values for the first discipline D1. As the disciplines only contain mathematical equations, the derivatives were also computed analytically to check that the finite-difference method is working correctly. From the sensitivity values a coupling dictionary was created. For example, for D1 it is as follows:

$$D1 : \{D3 : 0.2409, D4 : 0.1945, D5 = -0.1499\}$$

Table 5.5: Comparison between the analytical values and sensitivity values using the finite-difference method for the first discipline (D1).

Variable	Analytical value	Sensitivity value
$\frac{\partial y_{1,1}}{\partial y_{3,3}}$	0.15	0.15
$\frac{\partial y_{1,1}}{\partial y_{4,1}}$	0.15	0.15
$\frac{\partial y_{1,1}}{\partial y_{5,2}}$	-0.10	-0.10
$\frac{\partial y_{1,1}}{\partial y_{5,3}}$	-0.25	-0.25
$\frac{\partial y_{1,1}}{\partial y_{5,4}}$	0.20	0.20
$\frac{\partial y_{1,2}}{\partial y_{3,3}}$	0.0909	0.0909
$\frac{\partial y_{1,2}}{\partial y_{4,1}}$	0.0454	0.0454
$\frac{\partial y_{1,2}}{\partial y_{5,2}}$	0.0454	0.0454
$\frac{\partial y_{1,2}}{\partial y_{5,3}}$	-0.2272	-0.2272
$\frac{\partial y_{1,2}}{\partial y_{5,4}}$	0.1818	0.1818

After applying a branch-and-bound sequencing algorithm and the MDK decomposition algorithm with two partitions on the problem, the fundamental problem graph is created. The generated graph is shown on the right side of Figure 5.2. In addition, the left side of Figure 5.2 shows the original implementation of the sequencing and decomposition process with the goal to minimize the number of feedback connections. Both FPGs have the same partitioning, as the disciplines D2 and D3 are grouped together and the disciplines D1, D5, and D4 are placed together. The main difference is the sequence of the second partition, as the order of the disciplinary tools is D5, D4, and D1 compared to D1, D5, and D4.

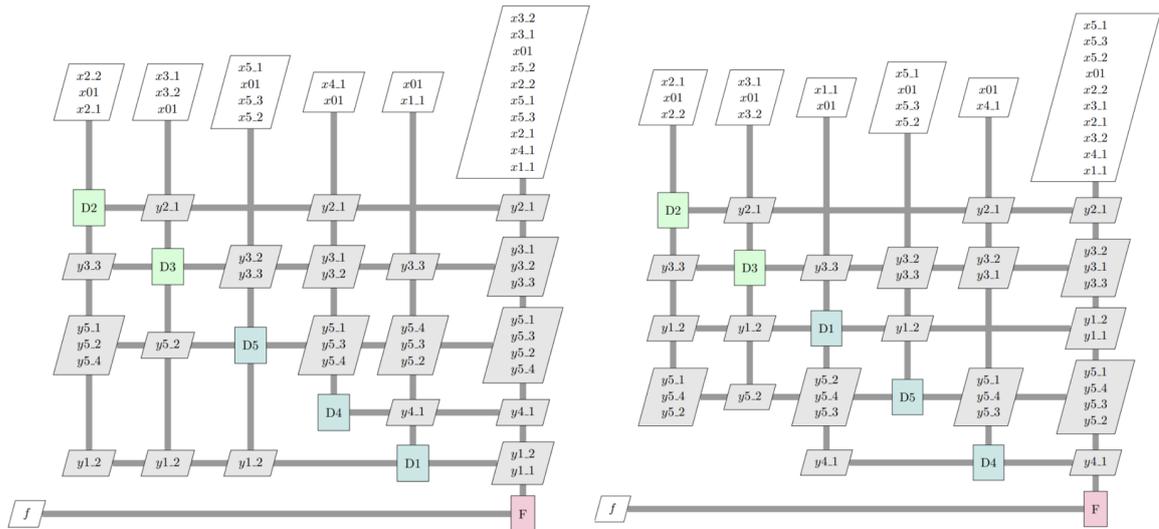


Figure 5.2: Left: FPG with the number minimum number of feedback connections, Right: FPG with sensitivity analysis.

An MDA to be converged is applied to both problem formulations and the XDSMs are shown in the Appendix A.1 and A.2. The created problem formulation is then passed onto the workflow software OpenMDAO in order to solve the problem and the results are shown in Table 5.6. For the implementation with the minimum number of feedback, solving the problem took 10 iterations, whereas the implementation with the sensitivity information took 9 iterations. The developed methodology reduced the number of iterations from 10 to 9 for the given problem.

Table 5.6: Comparison of the results of the optimization between the minimum number of feedback implementation and the implementation with sensitivity information.

	Minimum number of feedback	Implementation with sensitivity information
Number of coupling variables in feedback	8	11
Minimum objective	0.11206	0.11206
Number of iterations	10	9

After testing the developed methodology, the next step is to perform a verification and validation by comparing the results to analytical values and using a brute-force algorithm. As mentioned in Section 2.2.2, the brute-force algorithms determine all possible sequences and partitions. For the verification and validation, the variable complexity problem is again used with different inputs. The problem now consists of four disciplines with seven coupling variables. An overview of the RCG for the problem is given in Figure 5.3.

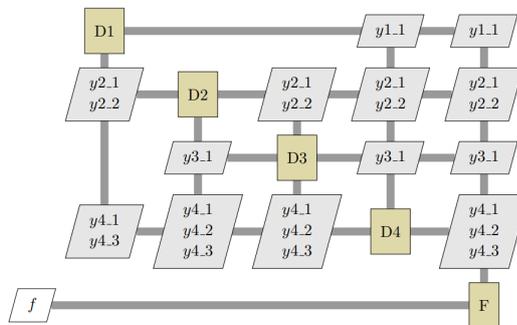


Figure 5.3: RCG of the variable complexity problem with four disciplines used for the verification and validation.

First, the sequencing algorithms were verified and validated. The problem consists of four disciplines, thus, there are 24 different possible sequences. For each of the possible sequences, the sensitivity values in the feedback were evaluated and results are shown in Table 5.7. As mentioned before, the goal of the improved sequencing algorithm is to find a sequence where the variables with a low sensitivity are placed in the feedback. After evaluating the results, two optimal sequences are found: D2, D4, D1, D3 and D2, D4, D3, D1 as they have the lowest sum of sensitivities in the feedback.

Table 5.7: The sum of sensitivities in the feedback for all possible sequences.

Sequence of disciplines	Sum of sensitivities in the feedback
D1, D2, D3, D4	-0.662
D1, D2, D4, D3	-0.867
D1, D3, D2, D4	-0.420
D1, D3, D4, D2	0.560
D1, D4, D2, D3	0.113
D2, D1, D3, D4	-0.412
D2, D1, D4, D3	-0.617
D2, D3, D1, D4	-0.412
D2, D3, D4, D1	-0.755
D2, D4, D1, D3	-0.960
D2, D4, D3, D1	-0.960
D3, D1, D2, D4	-0.420
D3, D1, D4, D2	0.560
D3, D2, D1, D4	-0.170
D3, D2, D4, D1	-0.513
D3, D4, D1, D1	0.217
D3, D4, D2, D1	0.467
D4, D1, D2, D3	-0.230
D4, D1, D3, D2	0.012
D4, D2, D1, D3	0.020
D4, D2, D3, D1	0.020
D4, D3, D1, D2	0.012
D4, D3, D2, D1	0.262

Next, a branch-and-bound sequencing algorithm and a brute-force algorithm were applied and the resulting FPGs are shown in Figure 5.4. Both algorithms found the same optimal sequence: D2, D4, D1, and D3 and they also agreed with the analytical results.

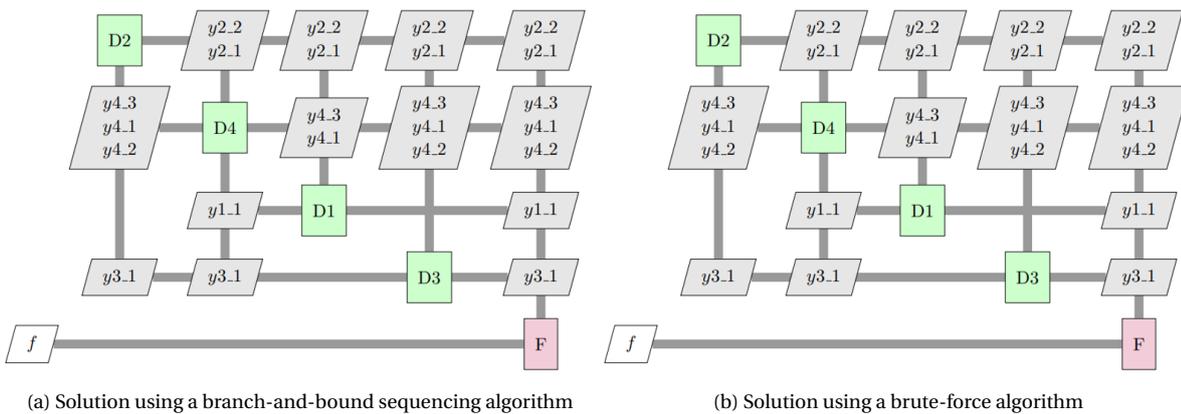


Figure 5.4: FPGs of the sequencing using different algorithms.

The MDK decomposition algorithm was verified analytically by solving for all possible combinations. The resulting FPGs for the two algorithms are shown in Figure 5.5. The exact same partitioning is obtained, as disciplines D4 and D1 create one partition and disciplines D2 and D3 create the second partition.

6

CASE STUDY 1

The developed methodologies were tested to generate the MDAO problem of a realistic aircraft design problem. The main purpose of case study 1 was to evaluate if integrating sensitivity information into the sequencing and decomposition algorithms of KADMOS in order to optimize them, respectively to use it additionally to screen for non-influential design variables during the MDAO problem formulation, would lead to a shorter MDAO execution time. Furthermore, it should be analyzed if this consideration of sensitivity data would lead to a reduction of the whole computation time, which consists of the time needed to execute the MDAO problem and the sensitivity analysis. Conclusively, the feasibility to include sensitivity analysis into MDAO formulation according to the methodologies used during case study 1 for the application within aircraft design should be discussed.

Section 6.1 provides an overview of the Aircraft Design Initiator tool that is used for the conceptual design and analysis of aircrafts, and which will be used for case study 1 and 2. Section 6.2 describes a design optimization study of an Airbus A320-200, including the MDAO problem formulation, sensitivity analysis, MDAO optimization, and MDAO execution.

Subsequently, a short overview of the steps performed for case study 1:

1. The first step is to establish the problem description of the aircraft design process for case study 1, where the objective, design variables, constraints, and top-level requirements are defined. The objective of case study 1 is to minimize the Maximum Take-Off Mass (MTOM) of an Airbus A320-200.
2. Next, the initial MDAO problem formulation is created with the use of methodology #2, where sensitivity information, derived from a local sensitivity method, is used within the sequencing and decomposition algorithms of KADMOS.
3. Afterwards, methodology #1 is utilized to identify non-influential design variables with the use of a global sensitivity analysis method.
4. The MDAO problem formulation is subsequently adjusted by removing the non-influential design variables.
5. The last step consist of performing the MDAO optimization and evaluating the results.

6.1. THE AIRCRAFT DESIGN INITIATOR

Case study 1 and also subsequently case study 2 were performed using the TU Delft Aircraft Design Initiator, which is an automated conceptual aircraft design and analysis tool developed at the Faculty of Aerospace Engineering at the Delft University of Technology [59, 60]. The Initiator is able to generate feasible aircraft designs that are based on a limited number of user-specified Top Level Aircraft Requirements (TLARs). In addition, the aircraft design complies with the regulations and requirements of the Federal Aviation Regulations and the European Aviation Safety Agency. Thus, the Initiator is suited to test the developed methodologies on an aircraft design.

The Initiator tool consists of various sizing, analyses, and design modules to perform a stepwise design synthesis. Figure 6.1 shows the schematic process diagram of the Initiator. The design process, implementing

the Initiator, is initiated with the specification of TLARs, such as the number of passengers, cargo weight, range requirements, airport runway length, cruise speed, etc. Next, the preliminary sizing of the airplane starts with a Class I Weight Estimation, using classical textbook sizing and weight estimation methods [61] which are used for the first iteration of the conceptual design loop.

These design routines are then augmented with more high-fidelity analysis methods, such as the numerical Class II.V EMWET wing weight estimation by Elham et al. [62], the fuselage configurator module, the aerodynamic analysis module, etc. The following modules are used during the case studies:

- *Database*: consists of reference data for different aircrafts and these data are utilized by other disciplines to empirically determine initial values for key parameters.
- *Empirical OEM*: calculates the Operational Empty Mass (OEM) based on information of the Database module.
- *Class 1 Weight Estimation*: calculates the aircraft mass breakdown.
- *Fuselage Configurator*: determines the fuselage geometry and the cabin layout.
- *Geometry Estimation*: determines the geometry of the wing, engine, and the horizontal and vertical stabilizers.
- *Athena Vortex Lattice (AVL)*: estimates the aerodynamic loads on the lifting surfaces and drag polar.
- *Wing Thrust Loading*: chooses a design point based on requirements that constraints the design space.
- *Class 2 Weight Estimation*: specifies a more detailed mass breakdown compared to the Class 1 weight estimation and consists of component masses and center of gravity locations.

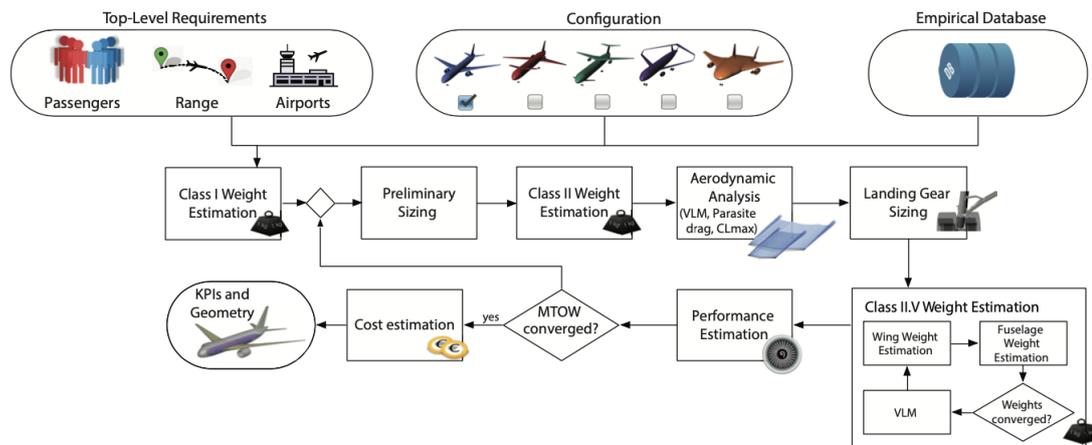


Figure 6.1: Schematic process diagram of the Initiator [60].

Several developments and changes were made in order to implement the Initiator in the MDAO development process using KADMOS [14]:

- *Creation of stand-alone modules*: all the modules are now independent and take a Controller object as an input.
- *Creation of the general data schema*: a new central data scheme was created that is able to be interpreted by the Initiator modules, KADMOS, and OpenLEGO. Figure 6.2 shows the central data schema. The central data schema starts with an Initiator element which is further divided into aircraft sub-elements. These sub-elements consist of missions, aircraft parts, performance parameters, results, etc.
- *Creation of the input and output files*: for each module in the Initiator, a list is generated with all the variables that are required to execute the module. In addition, a list is also made for all the output variables in each module.

- *Creation of a Python wrapper:* The Initiator modules are all written in MATLAB, whereas OpenLEGO and OpenMDAO are written in Python. Thus, a Python wrapper is used to execute the modules in OpenLEGO and OpenMDAO.

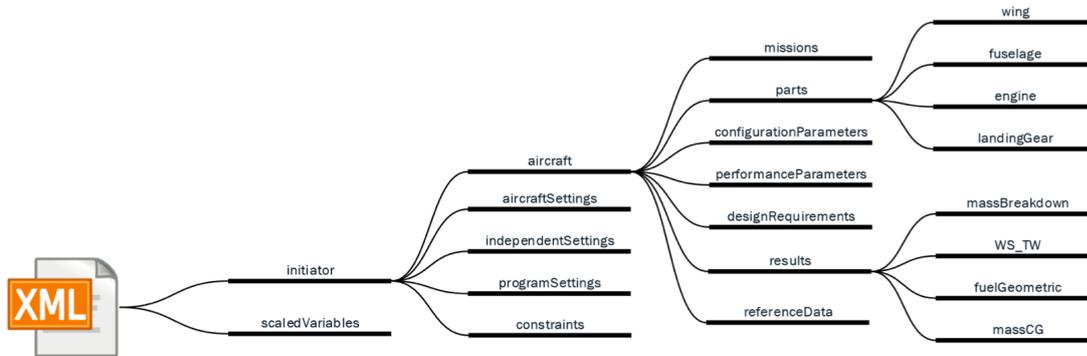


Figure 6.2: The central data schema for the Initiator [14].

6.2. THE CONCEPTUAL DESIGN OF AN AIRBUS A320-200

The case study was performed on a conceptual design of an Airbus A320-200 in which the objective is to minimize the Maximum Take-Off Mass (MTOM). The optimization problem is formulated as:

$$\begin{aligned}
 &\text{minimize: } W_{MTOM}(\mathbf{x}) \\
 &\text{with respect to: } \mathbf{x} \\
 &\text{subject to: } \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \\
 &\quad \mathbf{c}(\mathbf{x}) \leq 0
 \end{aligned} \tag{6.1}$$

where \mathbf{x} is the design vector, \mathbf{x}_l and \mathbf{x}_u are the upper and lower bounds of the design vector, and \mathbf{c} is the constraint vector. Seven parameters were selected as design variables and their upper and lower bounds are shown in Table 6.1. The design variables were chosen from the inputs of the RCG (Figure 6.3) and are related to the wing and empennage geometry. Commonly used design variables for minimizing the MTOM are using fuselage properties such as length and diameter. However, the Initiator uses an inside-out approach to determine the fuselage geometry; thus, the fuselage length or diameter cannot be used as a design variable. An overview of the constraints for the aircraft design problem is shown in Table 6.2. The top-level requirements that are used for this problem are depicted in Table 6.3.

Table 6.1: Overview of the design variables and their bounds for the aircraft design problem of case study 1.

Design variables	Bounds
Wing aspect ratio	$8 < AR < 11$
Wing kink position	$0.35 < \eta_{kink} < 0.40$
Horizontal tail taper ratio	$0.30 < TR_{HT} < 0.35$
Vertical tail taper ratio	$0.30 < TR_{VT} < 0.35$
Engine spanwise position	$0.30 < \eta_{eng} < 0.40$
Wing loading [N/m^2]	$2000 < W/S < 7000$
Thrust-to-weight ratio	$0.3 < T/W < 0.6$

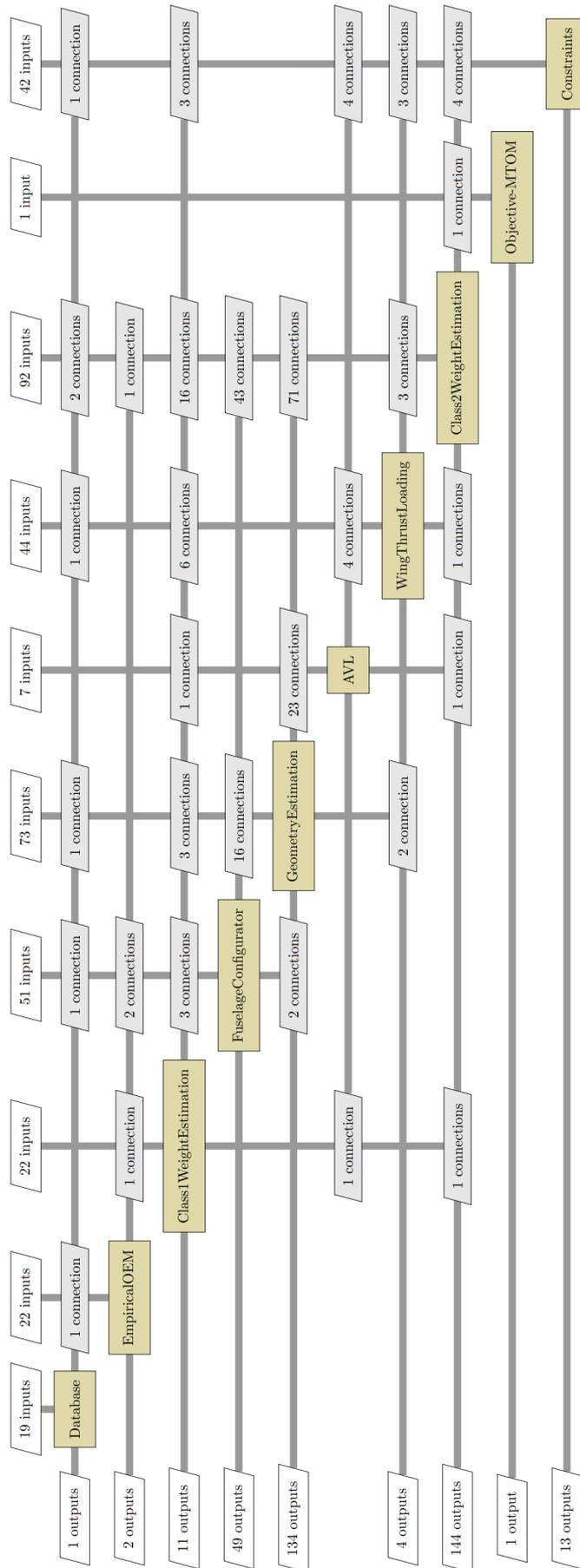


Figure 6.3: The repository connectivity graph for the design of an Airbus A320-200.

Table 6.2: Overview of the constraints for the aircraft design problem of case study 1.

Constraints
Balked landing climb AEO
Balked landing climb OEI
Cruise speed
En route climb
Fuel tank volume
Initial climb
Landing distance
Max cruise lift coefficient
Second segment climb
Take-off distance
Time to climb
Transition climb

Table 6.3: Airbus A320-200 top level requirements

Pax.	Payload Mass	Cruise Mach	Range	Altitude
156	20.5t	0.78	4000km	11.3km

The two developed methodologies, as described in Section 4.1 and 4.2, were created such that they are independent. Thus, as a next step the execution order of these methodologies needs to be determined to best fulfill the requirements of the case studies. For this case study, methodology #2 will be used first to generate the initial MDAO problem formulation, where a local sensitivity method is used to determine the sensitivity information and then this sensitivity information will be used within the sequencing and decomposition algorithms of KADMOS. Then, methodology #1 will be utilized to identify any non-influential design variables with the use of a global sensitivity method. Afterwards, the final MDAO problem formulation is generated and is used for the optimization.

6.2.1. MDAO PROBLEM FORMULATION

The first stage of the MDAO development process was formulated and a Repository Connectivity Graph (RCG) was generated and was loaded into KADMOS as a CMDOWS file. Figure 6.3 illustrates the RCG and shows a complete overview of the problem with all connections and the different Initiator modules.

SENSITIVITY ANALYSIS WITHIN THE SEQUENCING AND DECOMPOSITION ALGORITHMS

Next, the Fundamental Problem Graph (FPG) is generated with the use of methodology #2. A local sensitivity analysis using the finite-difference method was performed for each module and the obtained sensitivity values were used to create the coupling dictionary. The coupling dictionary was then passed on to the existing branch-and-bound sequencing algorithm and the MDK decomposition algorithm. Next, an MDA convergence architecture is imposed on the FPG to generate the MDG and MPG. Two different architectures were applied to investigate how many partitions should be used to obtain the lowest number of iterations until convergence is reached and thus subsequently to most efficiently optimize the MDAO execution in order to obtain the shortest execution time. These two architectures are a Gauss-Seidel convergence with one partition and the partition convergence with two partitions. The resulting architectures are shown in Figure 6.4 and 6.5, respectively.

For the partition convergence architecture with two partitions, a Runtime Coupling Balance (RCB) value of 0.5 was set, such that the execution time and coordination complexity were of equal importance. An execution time for each of the Initiator modules was specified and is summarized in Table 6.4. These execution times are an estimation, as at the beginning of the convergence the execution time is longer compared to when the solution is almost converged. For instance, the Fuselage Configuration module began with an execution time of around 12 seconds but once the solution was almost converged, the execution time decreased to around 7 seconds.

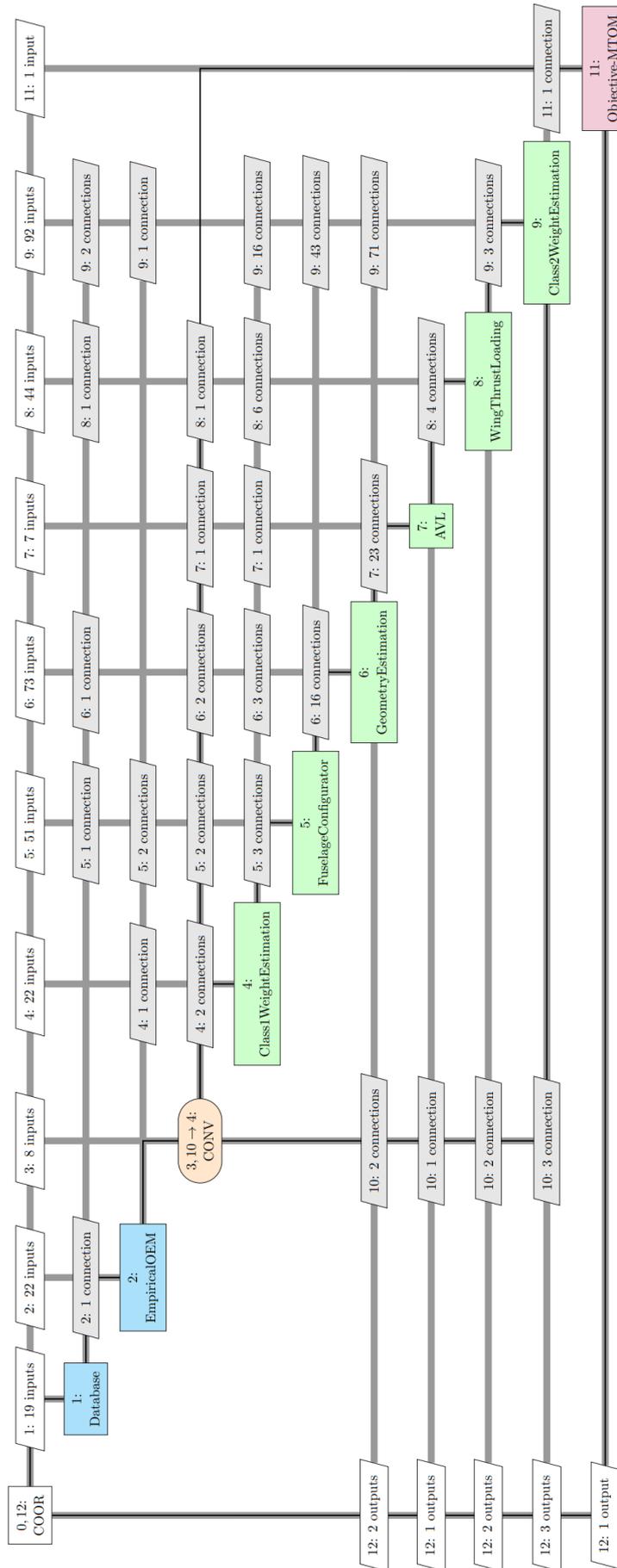


Figure 6.4: Gauss-Seidel convergence with one partition of the aircraft design problem when minimizing for MTOM.

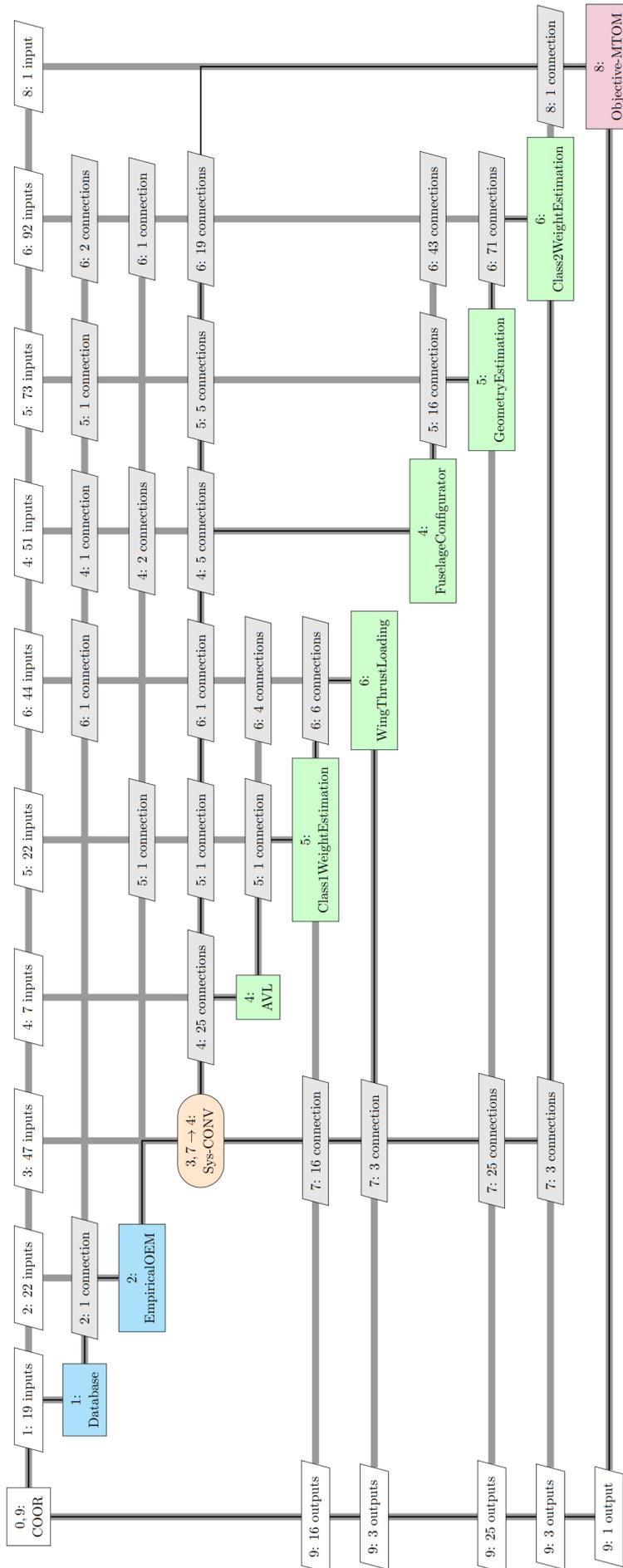


Figure 6.5: Partition convergence with two partitions of the aircraft design problem when minimizing for MTOM.

Table 6.4: Execution time of each Initiator module.

Module	Execution time [s]
Database	4
Empirical OEM	4
Class 1 Weight Estimation	3
Fuselage Configurator	7
Geometry Estimation	7
AVL	15
Wing Thrust Loading	5
Class 2 Weight Estimation	12

The architecture with two partitions (Figure 6.5) consists of one partition containing the AVL, Class 1 Weight Estimation, and the Wing Thrust Loading modules and has a total execution time of 23 seconds. The second partition contains the Fuselage Configurator, Geometry Estimation, and the Class 2 Weight Estimation modules and has a total execution time of 26 seconds.

The convergence results for the two different architectures are shown in Table 6.5. A convergence tolerance of 10^{-4} was used and was executed using 1 CPU. The two different architectures converge to the same MTOM value. For the two partitions, the number of iterations increased because more disciplines are executed in parallel, thus, the coordination complexity increased. The execution time of the Gauss-Seidel convergence was faster compared to the partitioned convergence with two partitions because the Initiator modules were not able to be executed in parallel within OpenMDAO. Hence, for the following steps the Gauss-Seidel convergence architecture with one partition will be used.

Table 6.5: Convergence results of the two different architectures: Gauss-Seidel convergence with one partition and a partition convergence with two partitions for the aircraft design problem.

	Gauss-Seidel convergence with one partition	Partition convergence with two partitions
Number of iterations	7	9
Execution time [s]	341	672
MTOM [t]	75485	75479

6.2.2. SENSITIVITY ANALYSIS TO IDENTIFY NON-INFLUENTIAL DESIGN VARIABLES

After the initial MDAO problem formulation is formed, the next step is to use methodology #1 to identify any non-influential design variables in the aircraft design problem. In order to perform sensitivity analysis, the developed sensitivity analysis tool was applied to the aircraft design problem and the steps are consequently outlined.

The first step is to select an appropriate sensitivity analysis method based on computational cost and accuracy of using each method. Table 6.6 shows the summary of the required runs to perform the sensitivity analysis on the aircraft design problem. A single model evaluation of the aircraft problem using the Gauss-Seidel convergence (Figure 6.4) takes around five minutes. Thus, using the Sobol or FAST method is not suitable due to the extensive computational cost as they would take around 21 respectively 38 hours to perform the sensitivity analysis.

Table 6.6: Number of model evaluations required for each global sensitivity method.

Sensitivity method	Number of model evaluations with 7 design variables
Sobol	256
FAST	455
Morris	40

The Morris method was chosen with five trajectory designs ($r=5$) [34]. As stated before, in Section 3.5.1, the Morris method outputs the standard deviations of the elementary effects (σ) and the mean of the absolute values of the elementary effects associated with each parameter (μ^*). The next step is to create the input values using the LHS sampling strategy. 40 different input sets were created such that each design variable has a specific input value. The input values are shown in Appendix B.1. Next, these input values were placed

in the Initiator input file. The Initiator input file uses the CPACS schema and an example is shown in Appendix B.2. The subsequent step is to execute the MDAO problem using the created inputs and the derived MDAO problem formulation (Figure 6.4) with the use of OpenLEGO and OpenMDAO. The resulting output values, the MTOM values, are stored in an XML file and are shown in Appendix B.3.

Subsequently, using the inputs and their corresponding outputs, the sensitivity analysis can be performed using the Morris method. The results of the sensitivity analysis are shown in Table 6.7 and of the Morris plot in Figure 6.6.

Table 6.7: Results of the sensitivity analysis using the Morris method.

Design variable	μ^*	σ
Wing aspect ratio	1491.4	1031.7
Wing kink position	15.3	24.0
Horizontal tail taper ratio	125.2	85.1
Vertical tail taper ratio	100.9	75.5
Engine spanwise position	20.3	31.2
Wing loading	216.0	89.2
Thrust-to-weight ratio	341.3	311.6

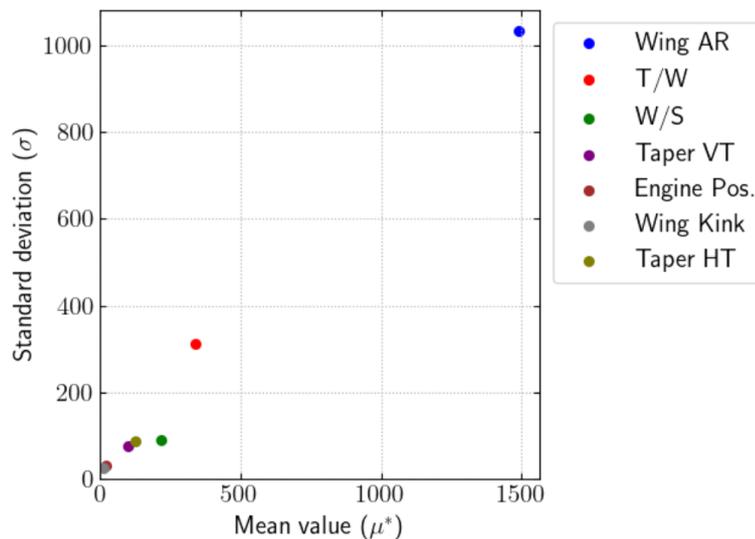


Figure 6.6: Morris plot of the sensitivity results with five trajectory designs.

Inspecting the μ^* values in Figure 6.6, the design variable that most affects the objective is the wing aspect ratio followed by the thrust-to-weight ratio, as they have the highest μ^* values. Whereas the engine spanwise position and the wing kink position are non-influential design variables as they are less than 5% of the maximum μ^* value. In order to evaluate whether the results of the sensitivity analysis are reasonable, it was analyzed how each of the design variables is used within the Initiator:

Wing aspect ratio: is an input of the Wing Thrust Loading module and the Geometry Estimation module. For the Wing Thrust Loading module the wing aspect ratio is used to compute the wing loading and the drag polar estimation. For the Geometry Estimation module, the wing aspect ratio is used for the calculation of various geometric properties of the wing such as the span and area. In the Class 2 Weight Estimation module the weight of the wing is calculated by using the Torenbeek method [61] and the wing weight is a function of the maximum zero fuel weight, span, wing area, semi-chord sweep angle, design ultimate load factor, and the maximum thickness of the wing root chord. The weight of the wing structure forms a large percentage of the aircraft's structural weight. Thus, altering the wing aspect ratio will have a significant effect on the MTOM.

Wing kink position: is an input of the Geometry Estimation module and is used to compute geometrical distances at the kink chord, centerline root chord, and tapers of the wing sections. However, the wing kink position is not used in determining the weight of the wing. Hence, it is reasonable that it is a non-influential design variable.

Horizontal tail taper ratio: is only used as an input for the Geometry Estimation module in order to calculate the horizontal root chord. The horizontal root chord is then used to compute the horizontal tail area. The horizontal tail area is passed to the Class 2 Weight Estimation module. The weight of the horizontal tail is calculated using the Torenbeek method where the inputs are the horizontal tail area, the design dive speed, and the horizontal tail semi-chord sweep angle. Thus, the horizontal taper ratio has a small indirect effect on the horizontal tail weight. Another reason why the horizontal taper ratio is a non-influential parameter is that the weight of the tail only accounts to about 2 to 3% of the MTOM.

Vertical tail taper ratio: uses the same approach as the horizontal tail taper ratio.

Engine spanwise position: is only an input of the geometry estimation module and is used to determine the position of the engines.

Wing loading: is an output of the Wing Thrust Loading module and an input of the Class 2 Weight Estimation module. The wing loading is used to calculate the weight of the wing.

Thrust-to-weight ratio: is an input of the Geometry Estimation module and the Class 2 Weight Estimation module and is used for the initial sizing. In addition, the thrust-to-weight ratio is used to calculate the thrust per engine which then is used to compute the propulsion system weight.

ROBUSTNESS TO SAMPLE SIZING OF THE MORRIS METHOD

A limitation of the Morris method is the level of accuracy when compared to the Sobol or FAST method. Having a larger sample size may lead to more accurate results but it involves a trade-off with the computational cost required to run the method on these extra samples. For the Morris method, the sample size is controlled by the number of trajectory designs (r).

The sensitivity analysis was repeated to investigate to what extent increasing the number of trajectory designs has an influence on the results of the sensitivity analysis. The number of trajectory designs was increased to ten which leads to a sample size of 80. The results of the sensitivity analysis are shown in Table 6.8 and Figure 6.7.

Table 6.8: Results of the sensitivity analysis using the Morris method with ten trajectory designs.

Design variable	μ^*	σ
Wing aspect ratio	1475.2 (1.09%)	1043.7 (1.16%)
Wing kink position	14.9 (2.63%)	24.8 (3.28%)
Horizontal tail taper ratio	127.3 (1.66%)	86.6 (1.74%)
Vertical tail taper ratio	103.9 (2.92%)	73.3 (2.95%)
Engine spanwise position	28.6 (1.40%)	30.9 (0.96%)
Wing loading	212.5 (1.63%)	88.7 (0.56%)
Thrust-to-weight ratio	335.5 (1.71%)	307.7 (1.26%)

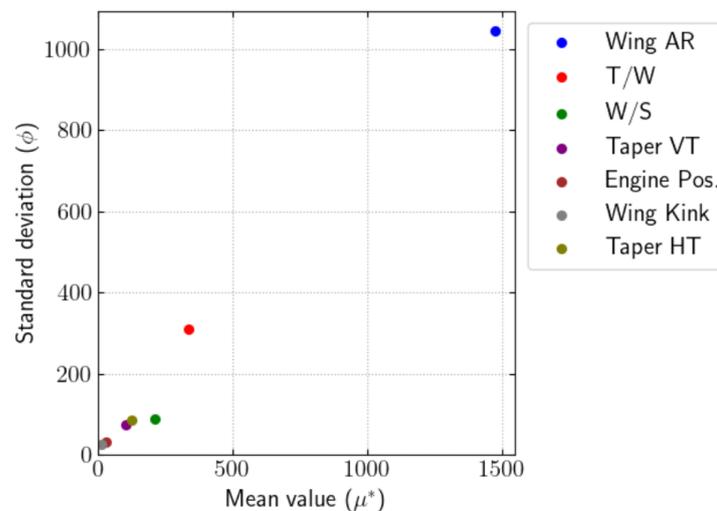


Figure 6.7: Morris plot of the sensitivity results with ten trajectory designs.

Table 6.8 also includes the percentage differences compared to the results of the previous sensitivity analysis with five trajectory designs. Both sensitivity analyses identified that the engine spanwise position and the wing kink position have a non-influential effect on the objective. Thus, it is concluded that using five trajectory designs with a sample size of 40 is sufficient and an increase in sample size will not lead to more accurate results. Furthermore, using ten trajectory designs doubled the time needed to perform the sensitivity analysis.

6.2.3. OPTIMIZATION OF THE MDAO PROBLEM FORMULATION

The MDAO problem formulation was adjusted according to the results of the above performed sensitivity analysis by updating the FPG. The two non-influential parameters, engine spanwise position and the wing kink position were removed as design variables and set to a constant value (middle of their bound: $\eta_{kink} = 0.375$, $\eta_{eng} = 0.35$).

An MDF Gauss-Seidel architecture was applied and the XDSM of the optimization for minimizing the MTOM is shown in Figure 6.8. The complete XDSM is shown in Appendix C. The results of the optimization are shown in Table 6.9 and Figure 6.9 shows the wing loading diagram. Figure 6.10 shows the history of MTOM values and the values for the design variables for each iteration. In addition, the optimization was also conducted on the problem without the removal of the two non-influential design variables in order to compare the results.

As shown in Table 6.9, the value of the objective (MTOM) is slightly different with a percent difference of 0.37% and increased by 296kg. The small difference in MTOM is due to the change of values for the two non-influential design variables. For instance, in the original problem the engine spanwise position had a value of 0.368, whereas for the problem with inclusion of the sensitivity analysis the value was set to 0.35. The values of the influential design variables for the original problem with seven design variables and the problem with the removal of the two design variables exhibited only a minor deviation.

Table 6.9: Results of the optimization for the original problem and the problem with the removal of two non-influential design variables.

	Original problem with 7 design variables	With sensitivity analysis and the removal of two non-influential design variables
<i>Objective</i>		
MTOM [kg]	80617	80913 (0.37%)
<i>Design variables</i>		
Wing aspect ratio	10.99	10.94
Wing kink position	0.383	-
Horizontal tail taper ratio	0.333	0.327
Vertical tail taper ratio	0.315	0.319
Engine spanwise position	0.368	-
Wing loading [N/m^2]	5141	5185
Thrust-to-weight ratio	0.272	0.267
<i>Constraints</i>		
Balked landing climb AEO	-0.6104	-0.6056
Balked landing climb OEI	-0.1068	-0.1098
Cruise speed	-0.2529	-0.2554
En route climb	-0.5371	-0.5374
Fuel tank volume	-0.4427	-0.4389
Initial climb	-0.1910	-0.1899
Landing distance	0.0015	0.0018
Max cruise lift coefficient	-0.2588	-0.2568
Second segment climb	-0.0513	0.0516
Take-off distance	0.0071	0.0074
Time to climb	-0.5555	-0.5558
Transition climb	-0.1410	-0.1406

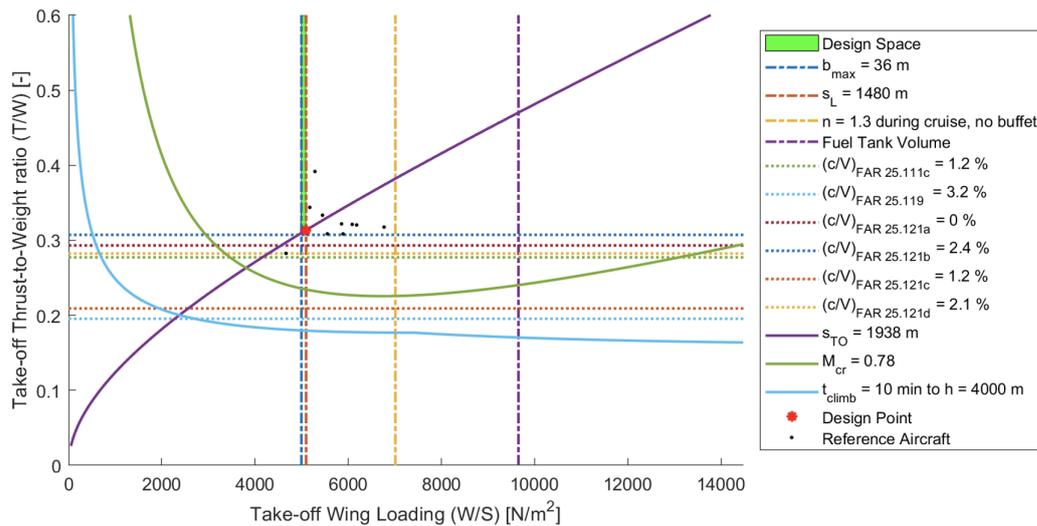


Figure 6.9: Wing thrust loading diagram for the problem with the removal of two design variables.

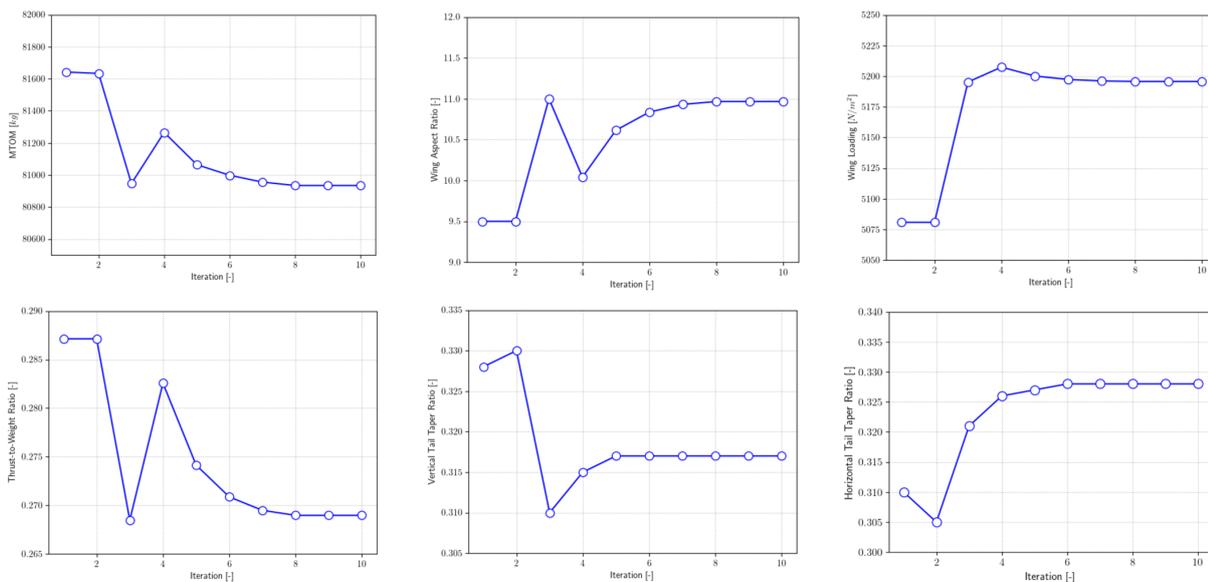


Figure 6.10: History of MTOM and the design variables for the minimization of the MTOM.

Table 6.10 shows the evaluation time of the optimization and the sensitivity analysis. The evaluation time of the optimization is reduced by 58.5 minutes when sensitivity analysis and the removal of two design variables is implemented. A main limitation of performing sensitivity analysis is that it takes a significant computational time. For this problem, it took 192.7 minutes to perform the sensitivity analysis as the problem had to be evaluated 40 times. Thus, the total time for the problem execution incorporating sensitivity analysis and the removal of two non-influential design variables increased by 134.2 minutes. However, for the next iteration of the conceptual design one could reuse the gathered sensitivity analysis information to prioritize tracking of these key design variables. Thus, the more optimizations that are performed the more beneficial the sensitivity analysis becomes in terms of time.

Table 6.10: Evaluation time of the optimization and sensitivity analysis.

	Original problem	With sensitivity analysis and the removal of two design variables
Evaluation time of the optimization [min]	143.3	84.8
Time for sensitivity analysis [min]	-	192.7
Total time [min]	143.3	277.5

7

CASE STUDY 2

The first case study demonstrated the benefits of using sensitivity analysis during the MDAO problem formulation phase, for instance, in being able to identify non-influential design variables and to use sensitivity information to optimize the sequencing and decomposition algorithms of KADMOS. For the subsequent MDAO execution the evaluation time is reduced when prior to the MDAO execution, the MDAO formulation is optimized by incorporating sensitivity information. However, it was concluded that the use of sensitivity analysis for MDAO of an aircraft design problem is limited given the computational time associated with performing the sensitivity analyses.

Aircraft preliminary design involves the optimization of several objectives. Thus, another case study was performed for the optimization of the Airbus A320-200 in which the objective is to minimize the fuel mass, additionally to minimizing the MTOM. The aim of the second case study is to analyze if performing sensitivity analysis concurrently on more than one objective variable during the MDAO problem formulation, will lead to time savings, consisting of the MDAO problem evaluation and the time needed to conduct the sensitivity analysis.

7.1. PROBLEM FORMULATION

The same problem formulation is used as described for the first case study (Figure 6.4). However, instead of only using MTOM as a quantity of interest, the fuel mass is also added as a quantity of interest. The resulting MDAO problem formulation which uses a Gauss-Seidel convergence architecture is shown in Figure 7.1. An MDA convergence architecture is used, as just the outputs from the given inputs are required, hence, an MDF optimization is not needed.

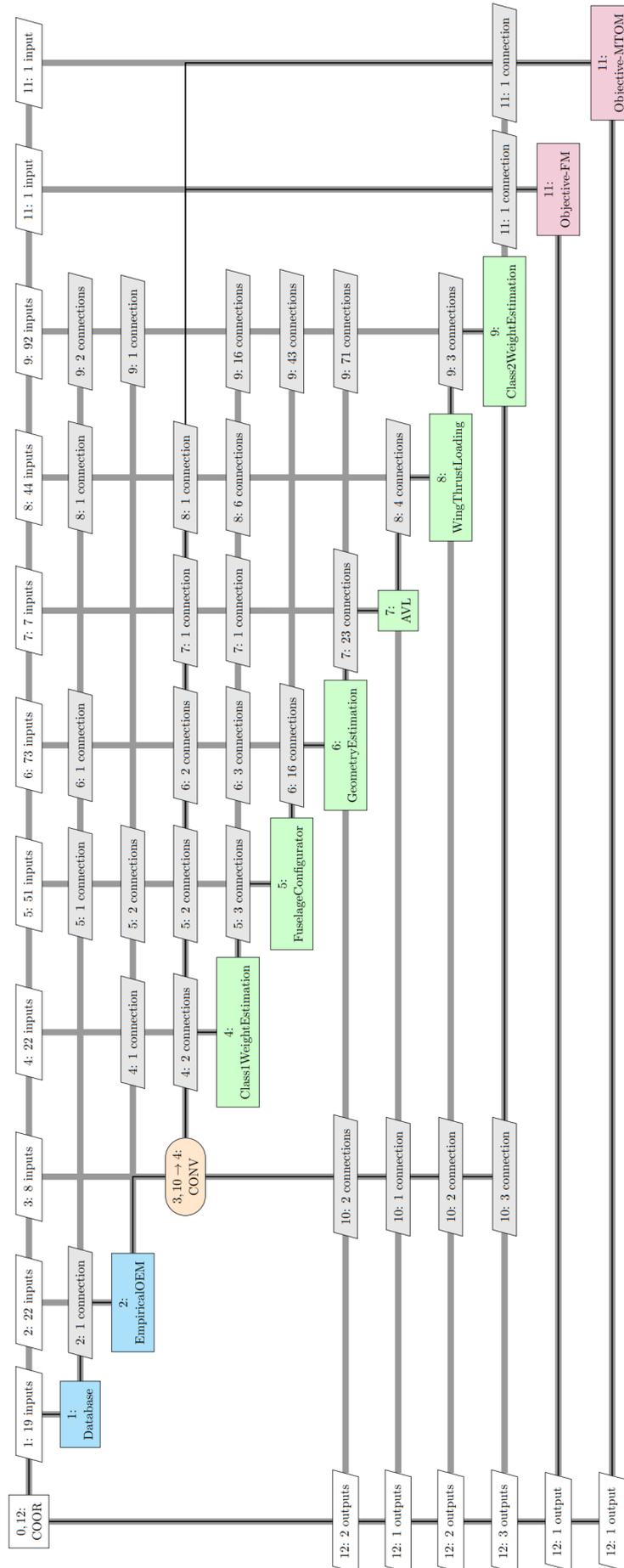


Figure 7.1: Problem formulation using the Gauss-Seidel convergence for the two objectives: MTOM and fuel mass.

7.2. SENSITIVITY ANALYSIS

For the sensitivity analysis the same seven design variables (Table 6.1) and sampling inputs (Appendix B.1) as for case study 1 are used. The above problem formulation, as depicted in Figure 7.1 is solved using OpenMDAO 40 times for every input set which results in the output values for the MTOM and fuel mass, shown in Appendix D. Using the input and output values, the Morris method is again used to perform the sensitivity analysis. The time to execute the sensitivity analysis took 183 minutes. Using this approach allows to save time as the sensitivity analysis does not need to be evaluated separately for each objective. Thus, applying sensitivity analysis to evaluate a second quantity of interest, increases the usefulness of executing the sensitivity analysis and integrating the results into the MDAO problem formulation. Performing the sensitivity analysis still involves computational expense and is time consuming, but in this case, can be applied for simultaneously evaluating MTOM and fuel mass as quantities of interest. An additional evaluation of more quantities of interest, while using the same approach, would implement an additional benefit to use sensitivity analysis.

7.2.1. MINIMIZING THE FUEL MASS

Table 7.1 and Figure 7.2 show the results of the sensitivity analysis for the design case when the objective is to minimize the fuel mass. The design variables that affect the objective the most are the thrust-to-weight ratio, the wing loading, and the wing aspect ratio. Only two design variables are identified as non-influential: the engine spanwise position and the position of the wing kink.

The results of the sensitivity analysis are different compared to the first case study when minimizing for the MTOM. The difference in the sensitivity analysis outcome is expected as global sensitivity methods evaluate the contribution of each input variables to the model output. However, the same two design variables are determined to be non-influential.

Table 7.1: Results of the sensitivity analysis using the Morris method for minimizing fuel mass.

Design variable	μ^*	σ
Wing aspect ratio	1544.9	3300.0
Wing kink position	76.7	104.8
Horizontal tail taper ratio	507.4	1100.3
Vertical tail taper ratio	486.3	1004.5
Engine spanwise position	110.3	132.9
Wing loading	1628.7	3628.4
Thrust-to-weight ratio	2429.5	4018.1

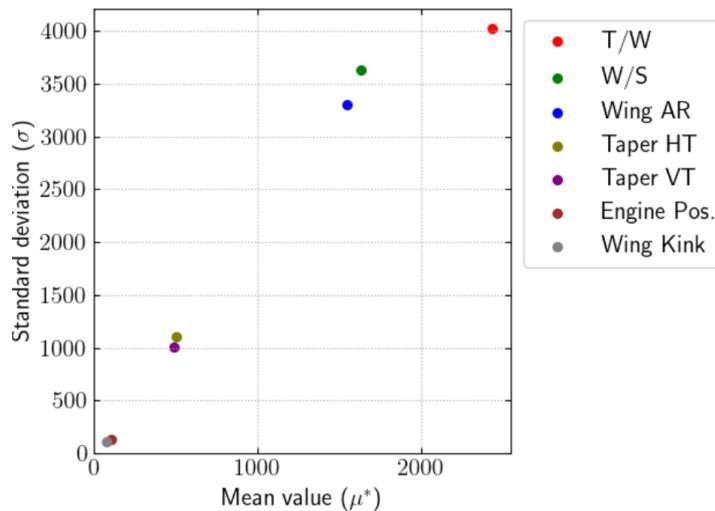


Figure 7.2: Morris plot of the sensitivity results for minimizing fuel mass.

8

CASE STUDY 3

The third case study investigates if the minimization of the sensitivities in the feedback loop will lead to fewer iterations until a converged solution is reached, compared to a feedback loop with higher sensitivities. Fewer iterations will lead to a decrease in computational time if the execution time per iteration does not increase too much. For the case study, the variable complexity problem (VCP) is again used (Section 5.2), as it is able to quickly generate numerous MDAO problems. The use of numerous MDAO problems allows to rule out any coincidences and obtain a good average. In addition, a confidence interval is used and is defined as a range around a measurement that conveys how precise the measurement is [63], in this case the equation for the confidence interval is as follows:

$$\left(\bar{x}_n - \frac{2z_{\alpha/2}s_n}{\sqrt{n}}, \bar{x}_n + \frac{2z_{\alpha/2}s_n}{\sqrt{n}} \right) \quad (8.1)$$

where \bar{x}_n is the average value, $z_{\alpha/2}$ is the value of the confidence level, s_n is the standard deviation, and n is the number of test cases. In order to obtain a confidence interval of 95%, $z_{\alpha/2}$ was set to 1.96.

100 randomly test cases were generated that contain five disciplines, five coupling variables per discipline, a coupling density of 0.1, and two partitions. The coefficient matrices (\mathbf{B}_{ij} , \mathbf{C}_i , \mathbf{D}_i) of the VCP changed for each test case and the values were generated randomly. Once again the finite-difference method was used to generate the sensitivity information and the procedure of the methodology #2 was applied, where the coupling dictionary was updated with the sensitivity data and then was passed to the sequencing and decomposition algorithms. An MDA Gauss-Seidel convergence architecture was applied on the problem, and the convergence tolerance was set to 10^{-10} . Subsequently, the test cases were solved. The number of iterations to achieve convergence for each test case is shown in Table 8.1 and the average number of iterations is shown in Figure 8.1. In addition, the same test cases were also solved using the original approach of minimum number of feedback connections in order to compare the results.

Table 8.1: Comparison of the number of iterations for each test case for the two approaches: the minimum number of feedback implementation and the implementation with sensitivity data.

Test case	Number of iterations for minimum feedback	Number of iterations with sensitivity data	Test case	Number of iterations for minimum feedback	Number of iterations with sensitivity data
1	7	5	51	6	6
2	5	5	52	5	5
3	8	7	53	10	9
4	9	6	54	9	8
5	10	9	55	6	6
6	6	6	56	10	8
7	7	7	57	7	6
8	10	9	58	7	7
9	7	7	59	8	7
10	7	6	60	6	6
11	7	7	61	9	8
12	11	8	62	6	6
13	7	7	63	8	7
14	10	9	64	7	7
15	7	7	65	7	7
16	8	7	66	6	6
17	9	9	67	7	7
18	9	8	68	8	8
19	12	9	69	12	9
20	7	6	70	8	7
21	10	9	71	7	7
22	7	7	72	9	8
23	9	8	73	7	7
24	7	7	74	10	7
25	6	6	75	6	6
26	6	6	76	7	6
27	7	6	77	7	6
28	10	8	78	8	7
29	5	5	79	8	6
30	9	8	80	8	8
31	7	6	81	6	6
32	9	8	82	8	7
33	5	5	83	7	7
34	4	4	84	7	7
35	10	9	85	9	8
36	4	4	86	8	8
37	11	9	87	7	6
38	6	6	88	9	8
39	12	10	89	7	7
40	10	7	90	7	7
41	7	7	91	7	7
42	5	5	92	8	7
43	6	6	93	6	6
44	7	6	94	8	7
45	6	6	95	6	6
46	5	5	96	7	7
47	6	6	97	6	6
48	8	7	98	9	8
49	10	8	99	10	7
50	8	8	100	8	8

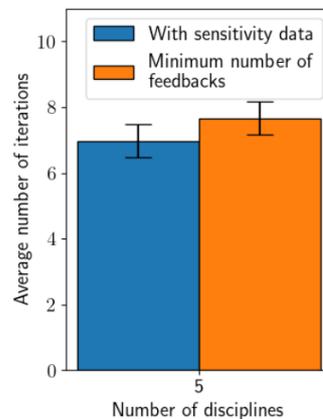
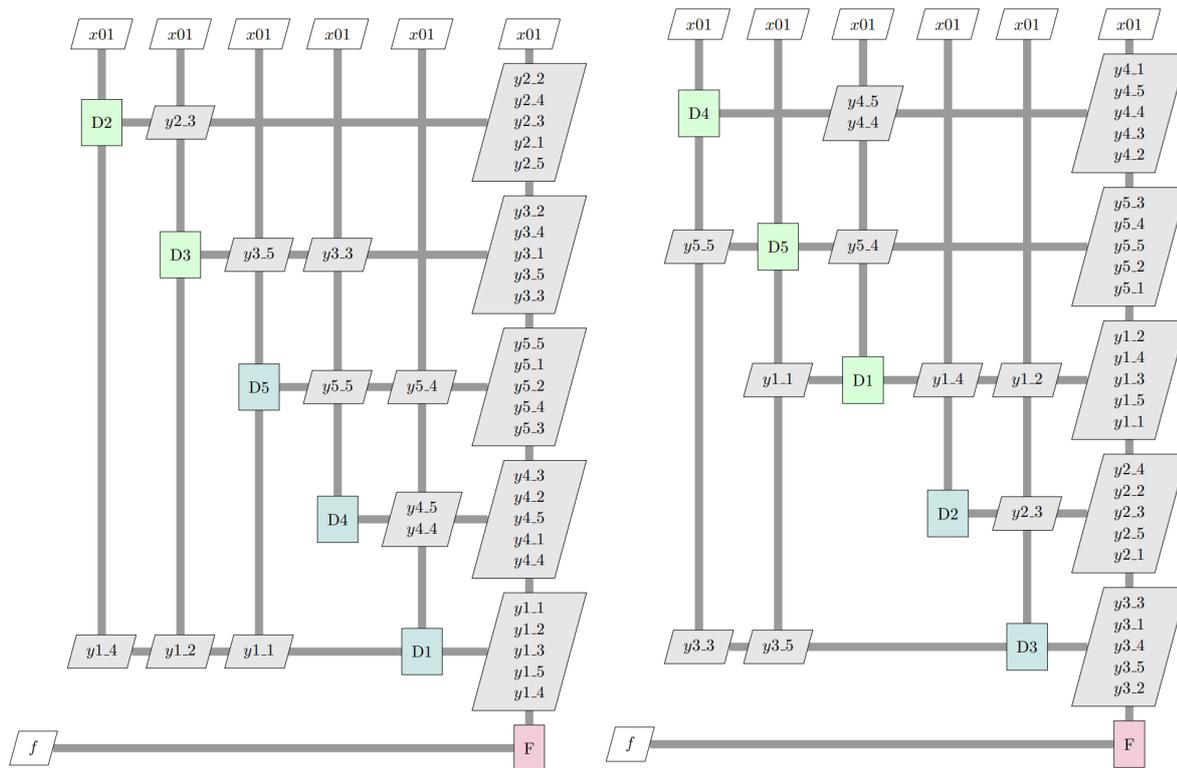


Figure 8.1: Comparison between the average number of iterations for the two approaches: when using sensitivity data and for minimum number of feedback for sequencing and decomposition algorithm.

Overall, for each test case the number of iterations decreased or remained the same with the utilization of sensitivity data, as will be outlined consequently. The average number of iterations for all 100 test cases is 6.97 when using sensitivity data and 7.66 when using the minimum number of feedback connections approach. To further evaluate the results, the convergence history for all the test cases was analyzed. Specifically, test cases number four and nine were evaluated, as test case four showed a large decrease in the number of iterations and test case nine had the same number of iterations. The FPGs for the fourth test case are compared in Figure 8.2 and the convergence history is shown in Table 8.2. Both problems had the same partitioning, as disciplines D2 and D3 are grouped together and disciplines D5, D4, and D1 formed another partition. However, the sequencing is different in both problems. With the utilization of the sensitivity data, the problem converged in six iterations. Whereas, when using the minimum number of feedback approach, the problem converged in nine iterations. Overall, it allowed to decrease the number of iterations by three and this signifies the

benefits of using sensitivity information within the sequencing and decomposition algorithms.



(a) Minimum number of feedback.

(b) Implementation with sensitivity data.

Figure 8.2: Comparison of the FPGs for test case 4 for the two different approaches.

Table 8.2: Comparison of the convergence history for test case 4.

Iteration	Convergence history for minimum feedback		Convergence history with sensitivity data	
1	6.20090294	1	5.68651852	1
2	1.72987797	0.278971947	1.55896226	0.274150564
3	0.150493292	0.0242695772	0.0567439262	0.00997867605
4	0.0181905039	0.00293352501	0.0024884629	0.000437607455
5	0.00277842969	0.000448068567	8.25818839e-05	1.45223978e-05
6	0.000427604298	6.89583924e-05	3.65264372e-06	6.42333918e-07
7	4.37851055e-05	7.06108544e-06	-	-
8	6.71355469e-06	1.08267373e-06	-	-
9	1.09620066e-06	1.76780812e-07	-	-

During the evaluation of the 100 test cases, some of the test cases did not show a reduction in the number of iterations. Such an example is test case number nine. The FPGs and the convergence history for both approaches are shown in Figure 8.3 and in Table 8.3, respectively. The FPGs show that both problems have the same partitioning and only a slight difference in sequencing, as disciplines D4 and D5 are switched. The convergence history also shows that the approach with sensitivity data converged slightly faster compared to the approach with minimum number of feedback. However, due to the fixed convergence tolerance, the number of iterations did not change. Overall, for all the cases which had the same number of iterations, the sequencing and partitioning only changed slightly.

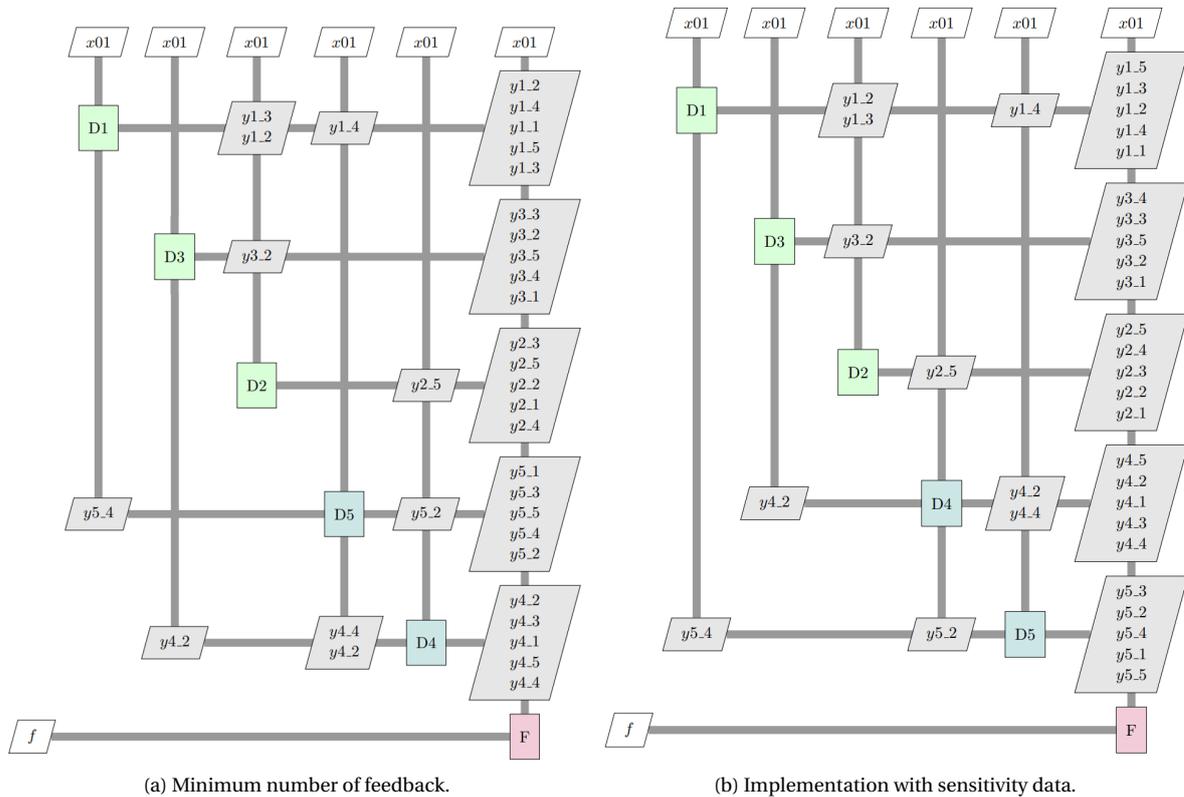


Figure 8.3: Comparison of the FPGs for test case 9 for the two different approaches.

Table 8.3: Comparison of the convergence history for test case 9.

Iteration	Convergence history for minimum feedback		Convergence history with sensitivity data	
	1	4.98660736	1	5.45412979
2	1.01684993	0.203916181	1.3243012	0.242807056
3	0.101155962	0.0202855277	0.100404439	0.0184088834
4	0.0115598517	0.00231817965	0.0102515098	0.00187958669
5	0.000747468644	0.000149895227	0.000156580793	2.87086664e-05
6	6.34026768e-06	1.27145917e-06	1.03201958e-05	1.89218009e-06
7	8.65216336e-07	1.73508013e-07	8.32730184e-07	1.52678835e-07

Furthermore, it was also investigated how a change in number of disciplines affects the number of iterations. 100 test cases were again generated for the variable complexity problem with seven disciplines and with nine disciplines. The test cases were solved using both approaches and the results are shown in Figure 8.4 and in Table 8.4. Figure 8.4 shows the average number of iterations for the different amount of disciplines and Table 8.4 also includes the values for the confidence interval. The results indicate that using sensitivity information within the sequencing and decomposition algorithms allows to reduce the number of iterations. An increase in number of disciplines leads to a similar result as discussed above.

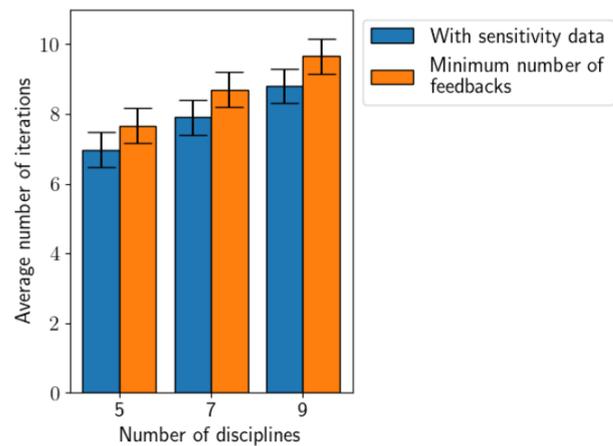


Figure 8.4: Comparison between the average number of iterations when using sensitivity information and for minimum number of feedback in the sequencing and decomposition algorithm.

Table 8.4: Average number of iterations and confidence level.

	With sensitivity analysis	Min. feedback connections	With sensitivity analysis	Min. feedback connections	With sensitivity analysis	Min. feedback connections
No. of disciplines	5	5	7	7	9	9
Average no. of iterations	6.97	7.66	7.90	8.71	8.85	9.65
Confidence interval	6.97 ± 0.432	7.66 ± 0.454	7.90 ± 0.441	8.71 ± 0.449	8.85 ± 0.421	9.65 ± 0.439

9

CONCLUSION & RECOMMENDATIONS

The increasing complexity of aeronautical systems and the prevailing focus to develop sustainable technological innovations for future aircraft configurations and technologies have emphasized a particular demand on advancing the Multidisciplinary Design Analysis and Optimization (MDAO) technique. MDAO perfectly addresses the complex synergy within aircraft design where multiple coupled disciplines and domains need to be efficiently analyzed and optimized. Considerable developments within MDAO, like the introduction of the software KADMOS (Knowledge-and graph-based Agile Design for Multidisciplinary Optimization System) have already generated a significant time reduction for setup and solving of an MDAO problem and consequently in lead time and costs of aircraft design.

The aims of this thesis are to further enhance the current application of MDAO for aircraft design and to alleviate the significant computational expense of MDAO problems through the consideration of sensitivity data during the MDAO formulation phase. Therefore, two approaches where the generation and evaluation of sensitivity information play a key role, were evaluated within this thesis, with the subsequent goal to shorten the MDAO execution time. Through the development and the following implementation of a sensitivity analysis tool into the sequencing and decomposition algorithms of KADMOS, this thesis demonstrated a new contribution to investigate on how to adjust the MDAO problem formulation. Furthermore, sensitivity information was used to identify the non-influential design variables respective to the objective during the MDAO problem formulation. Thus, during this thesis two independent methodologies were created in which the first methodology used a global sensitivity method to identify non-influential design variables, while the second methodology used a local sensitivity method to enhance the sequencing and decomposition algorithms of KADMOS.

Anyhow, performing sensitivity analysis, the evaluation of the corresponding sensitivity data, and the subsequent adaptation of the used design variables, respectively an adaptation of sequencing and decomposition algorithms will lead to an elevated set-up time for the MDAO problem formulation. Therefore, one additional goal of this thesis was to analyze if the consideration of sensitivity data will represent a beneficial task in terms of optimizing the MDAO problem formulation, respectively in terms of consequently reducing the time that is needed for MDAO execution. Furthermore, this thesis evaluated if integrating sensitivity analysis could provide an efficient tool to further enhance the MDAO development process. Based on the results of this thesis, several conclusions can be drawn:

- Firstly, non-influential design variables were able to be determined with the use of a global sensitivity method. This was demonstrated on an aircraft design problem where two non-influential design variables were identified using the Morris method. The two design variables were removed from the MDAO problem formulation and lead to a reduction in evaluation time of the optimization with only slightly impacting the accuracy of the results (the value of the objective had a percentage difference of 0.37%). However, due to the computational cost of the sensitivity analysis, the total computation time increased by 134 minutes.
- Secondly, performing sensitivity analysis concurrently on more than one objective variable during the MDAO problem formulation allowed for time savings, as the sensitivity analysis did not need to be repeated and could be applied simultaneously. The total computation time, consisting of the MDAO

problem evaluation and the time needed to conduct the sensitivity analysis, was reduced by approximately 180 minutes. This was demonstrated by performing the sensitivity analysis on an aircraft design problem where one objective was to minimize the MTOM and the other objective was to concurrently minimize the fuel mass.

- Thirdly, the sequencing and decomposition algorithms of KADMOS were successfully enhanced with the use of a local sensitivity method, the finite-difference method. Before this thesis, the goal of these algorithms was to minimize the number of feedback connections. This was altered such that the goal now is to determine a sequence where the variables with a low sensitivity are placed in the feedback. The improved algorithms were verified and validated on a variable complexity problem.
- Fourthly, it was shown that a feedback loop containing low sensitivities will lead to fewer iterations compared to a feedback loop with higher sensitivities, thus leading to a reduction in computational time. This was confirmed by testing it on numerous MDAO problems by utilizing a variable complexity problem. 100 different test cases, using the VCP, were evaluated and on average it led to a reduction of iterations of 0.69 iterations when compared to the original sequencing and decomposition algorithms.
- Fifthly, although the benefits of using sensitivity analysis are many, like to ascertain the key design variables for a specific design question, its use in practice is limited given the computational time associated with it. Nonetheless, the application of sensitivity analysis becomes a valuable procedure for the next iteration and optimization of a conceptual design problem with the same objective. Ultimately, with a subsequent number of optimizations the implementation of sensitivity analysis then would deliver a substantial reduction in time.

This thesis achieved to provide a new contribution to scientific literature as this thesis integrated sensitivity analysis into the sequencing and decomposition algorithms of KADMOS. In addition, the thesis demonstrated how to use sensitivity analysis to identify non-influential design variables which then additionally allowed to adjust the MDAO problem formulation. However, the following recommendations for future research were identified:

- *Implement a database that contains all the sensitivity information*
When applying the developed methodologies on the aircraft design problem, a great amount of sensitivity data is generated. Thus, further research could be the implementation of a database that stores all the sensitivity data into a storage repository. This repository can contain what the sensitivity data are and with what method they were computed. Having such a database available will help for future repetitions of a conceptual design and may reduce the need to repeat sensitivity analysis.
- *Investigate the approach of calculating the sensitivity information for a multidisciplinary problem based on disciplinary sensitivity analysis*
In this thesis, a global sensitivity analysis was performed on the entire system to identify non-influential design variables. However, a different approach could be to perform several disciplinary sensitivity analyses with fewer design variables as the number of design variables is a crucial aspect of the computational expense of sensitivity analysis. In addition, this could provide disciplinary sensitivity analysis information allowing the user to decide to refine disciplinary models with large influences on the system performances. However, there are several challenges with this approach. For example, if separated discipline sensitivity analyses are executed, then the sensitivity in the coupling variables between different disciplines are unknown. Another issue is when the disciplines share common design variables, the resulting coupling variables are not independent.
- *Implementation of constraint aggregation methods within the problem formulation*
In engineering problems, design constraints such as performance boundaries and safety limits, need to be incorporated and handled during the optimization process. However, the vast numbers of constraints have a significant effect on the optimization cost and convergence behavior. Thus, future research could be conducted to explore how constraint aggregation methods could be implemented in the problem formulation to reduce the computational cost.

A

VERIFICATION & VALIDATION

A.1. XDSM OF THE VARIABLE COMPLEXITY PROBLEM

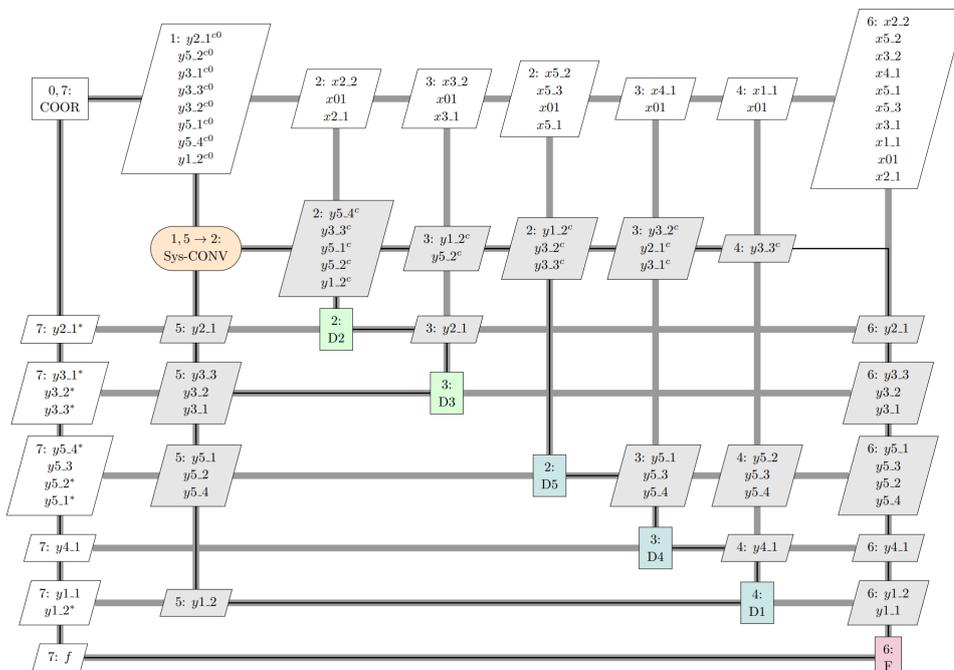


Figure A.1: XDSM of the MDA to be converged for the variable complexity problem with the approach of minimizing the number of feedbacks.

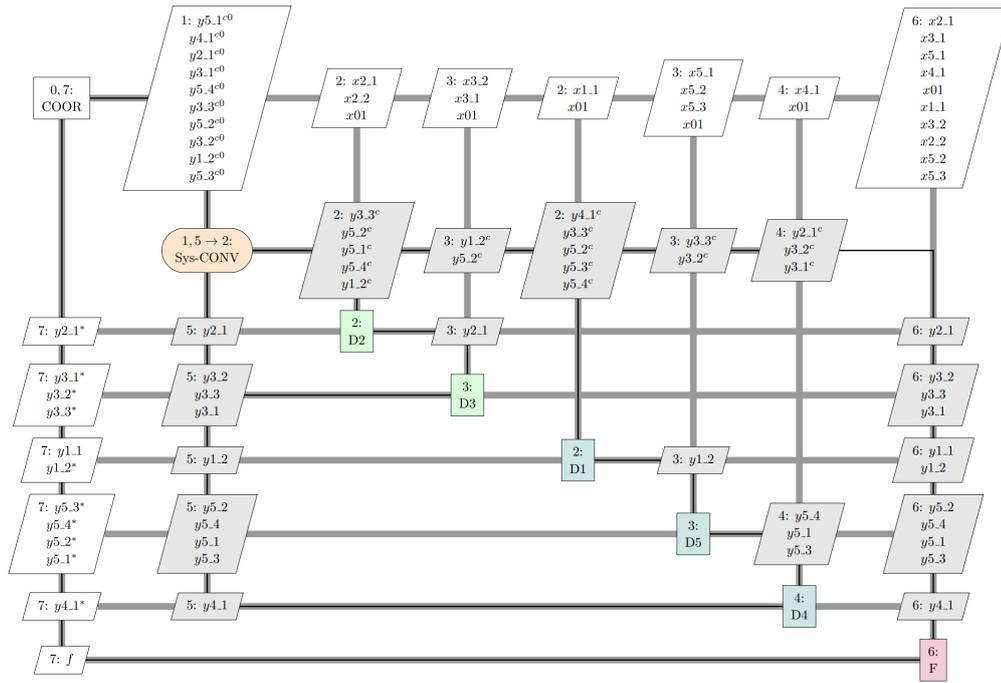


Figure A.2: XDSM of the MDA to be converged for the variable complexity problem with the approach of using sensitivity information.

B

INITIATOR SENSITIVITY ANALYSIS

B.1. INITIATOR INPUTS FOR SENSITIVITY ANALYSIS USING THE LHS STRATEGY

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <!-- ['WingAspectRatio', 'WS', 'TW', 'EnginePos', 'WingKink', 'TaperHT', 'TaperVT'] -->
3 <Input>
4 <Input0>[10.0, 3666.66666666667, 0.4, 0.3, 0.366666666666664, 0.35, 0.333333333333333]</Input0>
5 <Input1>[10.0, 3666.66666666667, 0.4, 0.3, 0.4, 0.35, 0.333333333333333]</Input1>
6 <Input2>[10.0, 7000.0, 0.6, 0.3, 0.4, 0.35, 0.333333333333333]</Input2>
7 <Input3>[10.0, 7000.0, 0.6, 0.3, 0.4, 0.35, 0.3]</Input3>
8 <Input4>[10.0, 7000.0, 0.6, 0.366666666666667, 0.4, 0.35, 0.3]</Input4>
9 <Input5>[8.0, 7000.0, 0.6, 0.366666666666667, 0.4, 0.35, 0.3]</Input5>
10 <Input6>[8.0, 7000.0, 0.6, 0.366666666666667, 0.4, 0.316666666666665, 0.3]</Input6>
11 <Input7>[10.0, 3666.66666666667, 0.4, 0.333333333333333, 0.366666666666664, 0.35, 0.3]</Input7>
12 <Input8>[10.0, 3666.66666666667, 0.4, 0.4, 0.366666666666664, 0.35, 0.3]</Input8>
13 <Input9>[8.0, 3666.66666666667, 0.4, 0.4, 0.366666666666664, 0.35, 0.3]</Input9>
14 <Input10>[8.0, 3666.66666666667, 0.4, 0.4, 0.366666666666664, 0.316666666666665, 0.3]</Input10>
15 <Input11>[8.0, 3666.66666666667, 0.6, 0.4, 0.366666666666664, 0.316666666666665, 0.3]</Input11>
16 <Input12>[8.0, 3666.66666666667, 0.6, 0.4, 0.366666666666664, 0.316666666666665, 0.333333333333333]</Input12>
17 <Input13>[8.0, 3666.66666666667, 0.6, 0.4, 0.4, 0.316666666666665, 0.333333333333333]</Input13>
18 <Input14>[8.0, 7000.0, 0.6, 0.4, 0.4, 0.316666666666665, 0.333333333333333]</Input14>
19 <Input15>[9.0, 5333.33333333333, 0.3, 0.333333333333333, 0.366666666666664, 0.333333333333333, 0.35]</Input15>
20 <Input16>[11.0, 5333.33333333333, 0.3, 0.333333333333333, 0.366666666666664, 0.333333333333333, 0.35]</Input16>
21 <Input17>[11.0, 5333.33333333333, 0.3, 0.333333333333333, 0.366666666666664, 0.3, 0.35]</Input17>
22 <Input18>[11.0, 5333.33333333333, 0.3, 0.333333333333333, 0.4, 0.3, 0.35]</Input18>
23 <Input19>[11.0, 5333.33333333333, 0.3, 0.333333333333333, 0.4, 0.3, 0.316666666666665]</Input19>
24 <Input20>[11.0, 2000.0, 0.3, 0.333333333333333, 0.4, 0.3, 0.316666666666665]</Input20>
25 <Input21>[11.0, 2000.0, 0.5, 0.333333333333333, 0.4, 0.3, 0.316666666666665]</Input21>
26 <Input22>[11.0, 2000.0, 0.5, 0.4, 0.4, 0.3, 0.316666666666665]</Input22>
27 <Input23>[10.0, 7000.0, 0.5, 0.333333333333333, 0.366666666666664, 0.35, 0.333333333333333]</Input23>
28 <Input24>[10.0, 3666.66666666667, 0.5, 0.333333333333333, 0.366666666666664, 0.35, 0.333333333333333]</Input24>
29 <Input25>[10.0, 3666.66666666667, 0.5, 0.333333333333333, 0.366666666666664, 0.316666666666665, 0.333333333333333]</Input25>
30 <Input26>[10.0, 3666.66666666667, 0.5, 0.333333333333333, 0.4, 0.316666666666665, 0.333333333333333]</Input26>
31 <Input27>[10.0, 3666.66666666667, 0.3, 0.333333333333333, 0.4, 0.316666666666665, 0.333333333333333]</Input27>
32 <Input28>[8.0, 3666.66666666667, 0.3, 0.333333333333333, 0.4, 0.316666666666665, 0.333333333333333]</Input28>
33 <Input29>[8.0, 3666.66666666667, 0.3, 0.4, 0.4, 0.316666666666665, 0.333333333333333]</Input29>
34 <Input30>[8.0, 3666.66666666667, 0.3, 0.4, 0.4, 0.316666666666665, 0.3]</Input30>
35 <Input31>[8.0, 7000.0, 0.5, 0.366666666666667, 0.35, 0.3, 0.333333333333333]</Input31>
36 <Input32>[8.0, 7000.0, 0.5, 0.366666666666667, 0.35, 0.333333333333333, 0.333333333333333]</Input32>
37 <Input33>[8.0, 7000.0, 0.5, 0.3, 0.35, 0.333333333333333, 0.333333333333333]</Input33>
38 <Input34>[8.0, 7000.0, 0.5, 0.3, 0.383333333333333, 0.333333333333333, 0.333333333333333]</Input34>
39 <Input35>[10.0, 7000.0, 0.5, 0.3, 0.383333333333333, 0.333333333333333, 0.333333333333333]</Input35>
40 <Input36>[10.0, 3666.66666666667, 0.5, 0.3, 0.383333333333333, 0.333333333333333, 0.333333333333333]</Input36>
41 <Input37>[10.0, 3666.66666666667, 0.3, 0.3, 0.383333333333333, 0.333333333333333, 0.333333333333333]</Input37>
42 <Input38>[10.0, 3666.66666666667, 0.4, 0.3, 0.383333333333333, 0.333333333333333, 0.3]</Input38>
43 <Input39>[10.0, 3666.66666666667, 0.3, 0.3, 0.383333333333333, 0.333333333333333, 0.3]</Input39>
44 </Input/>
45
```

B.2. INITIATOR INPUT FILE

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <cpacs>
3   <toolspecific>
4     <TUDinitiator>
5       <initiator>
6         <aircraft uID="A320-200">
7           <parts>
8             <Wing uID="MainWing">
9               <Type>MainWing</Type>
10              <Sections>
11                <Airfoil>
12                  <AirfoilName>b737a</AirfoilName>
13                </Airfoil>
14                <Airfoil>
15                  <AirfoilName>b737b</AirfoilName>
16                </Airfoil>
17                <Airfoil>
18                  <AirfoilName>b737d</AirfoilName>
19                </Airfoil>
20              </Sections>
21              <Kink>0.4</Kink>
22              <SectionPositions mapType="vector">0.0000000000000000;0.3770000000000000;1.
0000000000000000</SectionPositions>
23              <SparPositions mapType="vector">0.1000000000000000;0.6500000000000000</SparPositions>
24              <FuelTank>
25                <HasTank>1.0000000000000000</HasTank>
26                <SpanPosition mapType="vector">0.0000000000000000;0.8000000000000000</SpanPosition>
27              </FuelTank>
28            </Wing>
29            <Wing uID="HorizontalStabiliser">
30              <Type>HorizontalTail</Type>
31              <Sections>
32                <Airfoil>
33                  <AirfoilName>N64A010</AirfoilName>
34                </Airfoil>
35                <Airfoil>
36                  <AirfoilName>N64A010</AirfoilName>
37                </Airfoil>
38              </Sections>
39              <Tapers>0.3</Tapers>
40              <SparPositions mapType="vector">0.1000000000000000;0.6500000000000000</SparPositions>
41              <FuelTank>
42                <HasTank>0.0000000000000000</HasTank>
43              </FuelTank>
44            </Wing>
45            <Wing uID="VerticalStabiliser">
46              <Type>VerticalTail</Type>
47              <Sections>
48                <Airfoil>
49                  <AirfoilName>N64A010</AirfoilName>
50                </Airfoil>
51                <Airfoil>
52                  <AirfoilName>N64A010</AirfoilName>
53                </Airfoil>
54              </Sections>
55              <Tapers>0.3</Tapers>
56              <FuelTank>
57                <HasTank>0.0000000000000000</HasTank>
58              </FuelTank>
59            </Wing>
60            <Fuselage uID="Fuselage">
61              <Type>Conventional</Type>
62              <HeightMargin>0.3000000000000000</HeightMargin>
63              <NoseShapeFactor>0.1500000000000000</NoseShapeFactor>
64              <TailShapeFactor>0.2000000000000000</TailShapeFactor>
65              <AftRatioWidth>0.1400000000000000</AftRatioWidth>
66              <AftRatioHeight>0.1400000000000000</AftRatioHeight>
67              <FloorOffset mapType="vector">0.0000000000000000;0.0000000000000000</FloorOffset>
68              <PaxDivision>1.0000000000000000</PaxDivision>
69              <CabinHeight>1.9299999999999999</CabinHeight>
70              <Cabins>
71                <Cabin uID="Cabin1">
72                  <ClassDistribution mapType="vector">0.0800000000000000;0.0000000000000000;0.
0000000000000000;0.9200000000000000</ClassDistribution>
73                <Classes>
74                  <EC>
75                    <seatingArr mapType="vector">3.0000000000000000;3.0000000000000000</seatingArr>
76                    <seatingDim mapType="vector">0.4600000000000000;0.0480000000000000;0.
8129999999999999;0.8000000000000000;0.3000000000000000</seatingDim>
```

```

77         </EC>
78         <FC>
79         <seatingArr mapType="vector">2.0000000000000000;2.0000000000000000</seatingArr>
80         <seatingDim mapType="vector">0.5700000000000000;0.0780000000000000;0.
9140000000000000;0.8000000000000000;0.3000000000000000</seatingDim>
81         </FC>
82     </Classes>
83 </Cabin>
84 </Cabins>
85 <FuelTank>
86     <HasTank>0.0000000000000000</HasTank>
87 </FuelTank>
88 <CargoBays uID="BulkBay1">
89     <CargoType uID="LD3-45">
90         <Type>ULD</Type>
91         <ULDType>LD3-45</ULDType>
92     </CargoType>
93     <Locations>.generatedProperties/partsFuselage[1]CargoBays[1]Locations[1].txt</
Locations>
94         <NumContainers>5.0000000000000000</NumContainers>
95     </CargoBays>
96 <CargoBays uID="BulkBay2">
97     <CargoType uID="LD3-45">
98         <Type>ULD</Type>
99         <ULDType>LD3-45</ULDType>
100 </CargoType>
101     <Locations>.generatedProperties/partsFuselage[1]CargoBays[2]Locations[1].txt</
Locations>
102         <NumContainers>8.0000000000000000</NumContainers>
103     </CargoBays>
104 </Fuselage>
105 <Engine uID="Engine1">
106     <Type>TurboFan</Type>
107     <Location>MainWing</Location>
108     <BypassRatio>5.7000000000000002</BypassRatio>
109     <NacelleSection>
110         <Airfoil>
111             <AirfoilName>PARSEC1</AirfoilName>
112         </Airfoil>
113     </NacelleSection>
114 </Engine>
115 <Engine uID="Engine2">
116     <Type>TurboFan</Type>
117     <Location>MainWing</Location>
118     <BypassRatio>5.7000000000000002</BypassRatio>
119     <NacelleSection>
120         <Airfoil>
121             <AirfoilName>PARSEC1</AirfoilName>
122         </Airfoil>
123     </NacelleSection>
124 </Engine>
125 </parts>
126 <missions>
127     <mission uID="Mission1">
128         <Altitude>11280.0000000000000000</Altitude>
129         <CruiseMach>0.7800000000000000</CruiseMach>
130         <DivRange>370.0000000000000000</DivRange>
131         <LoiterTime>30.0000000000000000</LoiterTime>
132         <Pax>156.0000000000000000</Pax>
133         <PayloadMass>20536.0000000000000000</PayloadMass>
134         <Range>3917.0000000000000000</Range>
135     </mission>
136     <DefaultMissionuID>Mission1</DefaultMissionuID>
137 </missions>
138 <configurationParameters>
139     <CompositeStructures mapType="vector">0.0000000000000000;0.0000000000000000;1.
0000000000000000</CompositeStructures>
140     <WingLocation>Low</WingLocation>
141     <Freight>0.0000000000000000</Freight>
142     <WingAspectRatio>11.0</WingAspectRatio>
143     <SupercriticalAirfoil>1.1000000000000001</SupercriticalAirfoil>
144     <TEinboardSweep>0.0000000000000000</TEinboardSweep>
145     <TailType>standard</TailType>
146 </configurationParameters>
147 <performanceParameters>
148     <SFC>0.5</SFC>
149     <LDmax>15.666666666666668</LDmax>
150     <CDw>0.0020000000000000</CDw>
151     <CDf>0.0180000000000000</CDf>

```

```

152 <CLmaxClean>1.6000000000000001</CLmaxClean>
153 <CLmaxTakeOff>2.2000000000000002</CLmaxTakeOff>
154 </performanceParameters>
155 <results>
156 <MassBreakdown>
157 <OEM>45847.5342473829223309</OEM>
158 <FerryMission>
159 <Extended>0.000000000000000</Extended>
160 <RangeFactor>1.000000000000000</RangeFactor>
161 </FerryMission>
162 </MassBreakdown>
163 <WS_TW>
164 <WS>7000.0</WS>
165 <TW>0.3</TW>
166 </WS_TW>
167 <FuelGeometric>
168 <Vs mapType="vector">20.6481407692632501;20.6481407692632501</Vs>
169 <TotalFV>41.2962815385265003</TotalFV>
170 <TotalCG mapType="vector">16.3761238091188730;0.000000000000000;-0.0694242269696460</
TotalCG>
171 <TotalFM>33449.9880462064684252</TotalFM>
172 </FuelGeometric>
173 </results>
174 <designRequirements>
175 <CargoMass>8056</CargoMass>
176 <AirworthinessRegulations>FAR-25</AirworthinessRegulations>
177 <TakeOffDistance>1938.000000000000000</TakeOffDistance>
178 <LandingDistance>1480.000000000000000</LandingDistance>
179 <AirportClassification>FAA-III</AirportClassification>
180 <TimeToClimb mapType="vector">10.000000000000000;4000.000000000000000</TimeToClimb>
181 <LoiterTime>30.000000000000000</LoiterTime>
182 <DivRange>370.000000000000000</DivRange>
183 <CruiseMach>0.780000000000000</CruiseMach>
184 <CruiseSpeed>230.1542971421397965</CruiseSpeed>
185 <Altitude>11280</Altitude>
186 <MaxPax>156</MaxPax>
187 <Pax>156</Pax>
188 <Range>3917.000000000000000</Range>
189 </designRequirements>
190 </aircraft>
191 <programSettings>
192 <uGridScaling>2.000000000000000</uGridScaling>
193 <vGridScaling>2.000000000000000</vGridScaling>
194 <PaxBoundary>20.000000000000000</PaxBoundary>
195 <ShowMessages>1.000000000000000</ShowMessages>
196 <UseArtificialDB>0</UseArtificialDB>
197 <InsideOutMargin>0.001000000000000</InsideOutMargin>
198 <AirfoilPoints>121.000000000000000</AirfoilPoints>
199 <plotWSmin>50.000000000000000</plotWSmin>
200 <plotWSstep>25.000000000000000</plotWSstep>
201 <NumberReferenceAC>10.000000000000000</NumberReferenceAC>
202 <ShowReferenceACNames>0.000000000000000</ShowReferenceACNames>
203 <DCtolerances mapType="vector">0.001000000000000;0.001000000000000;0.001000000000000</
DCtolerances>
204 </programSettings>
205 <independentSettings>
206 <PaxMass>80.000000000000000</PaxMass>
207 <LuggageMass>22.000000000000000</LuggageMass>
208 <CompositeCorrection mapType="vector">0.850000000000000;0.750000000000000;0.
750000000000000</CompositeCorrection>
209 <FuelHq>4350.000000000000000</FuelHq>
210 <FFStartup>0.990000000000000</FFStartup>
211 <FFTaxi>0.990000000000000</FFTaxi>
212 <FuelDensity>810.000000000000000</FuelDensity>
213 <PylonEstimationAirfoil>N0012</PylonEstimationAirfoil>
214 <MaxLoadFactor>2.500000000000000</MaxLoadFactor>
215 <SafetyFactor>1.500000000000000</SafetyFactor>
216 <DiveMachFactor>1.200000000000000</DiveMachFactor>
217 </independentSettings>
218 <referenceData>
219 </referenceData>
220 <aircraftSettings>
221 <DesignMethod>inside-out</DesignMethod>
222 <NoseDroop>0.150000000000000</NoseDroop>
223 <UpSweep>0.300000000000000</UpSweep>
224 <averageLavatoryArea>1.000000000000000</averageLavatoryArea>
225 <galleyAreaPerPassenger mapType="vector">0.110000000000000;0.090000000000000;0.
060000000000000</galleyAreaPerPassenger>
226 <lavatoryPerPassenger mapType="vector">0.041700000000000;0.058800000000000;0.

```

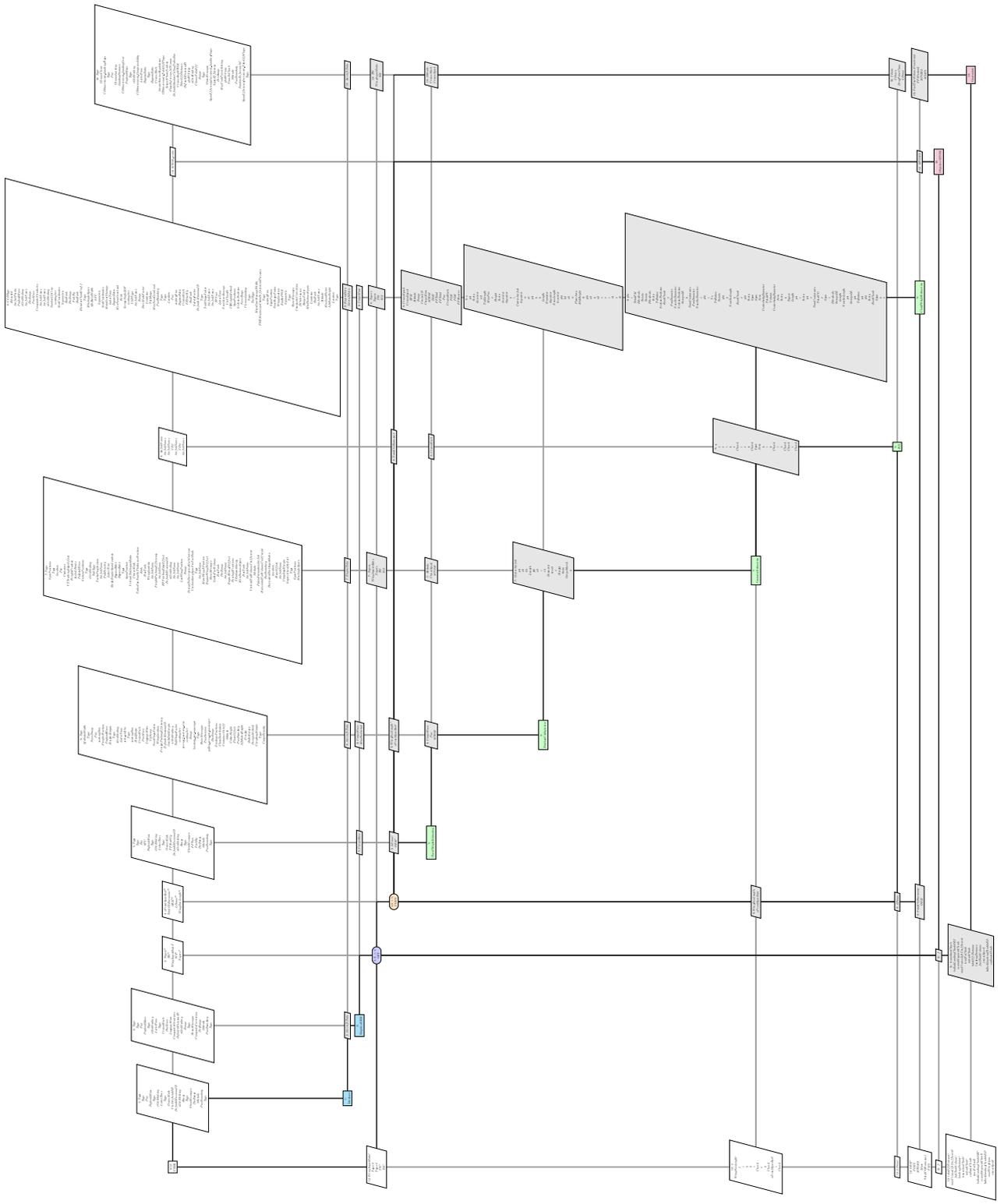
```
226 0333000000000000;0.0167000000000000</Lavatory_per_passenger>
227 <aisle_widths mapType="vector">0.7650000000000000;0.5700000000000000;0.5300000000000000;0.
5100000000000000</aisle_widths>
228 <CargobayMargin>0.1000000000000000</CargobayMargin>
229 <FreightFraction>0.0000000000000000</FreightFraction>
230 <FreightPackingEfficiency>0.5000000000000000</FreightPackingEfficiency>
231 <CargoDensity>423.19999999999986</CargoDensity>
232 <ContainerCutoff>0.6500000000000000</ContainerCutoff>
233 <BulkMargin>0.4000000000000000</BulkMargin>
234 <WingBoxLength>0.0500000000000000</WingBoxLength>
235 <xFrontSparRoot>0.3162000000000000</xFrontSparRoot>
236 <FuseLageFractions mapType="vector">0.1100000000000000;0.7400000000000000;0.
1500000000000000</FuseLageFractions>
237 <HeadRoom>1.649999999999999</HeadRoom>
238 <DesignToDivergenceMachIncrease>0.0150000000000000</DesignToDivergenceMachIncrease>
239 <MainWingXPosition>0.3162000000000000</MainWingXPosition>
240 <LowWingPosition>-0.3000000000000000</LowWingPosition>
241 <MinDistanceSpars>0.0500000000000000</MinDistanceSpars>
242 <MinAngleSpars>8.0000000000000000</MinAngleSpars>
243 <HorizontalTailVolumeCoefficient>1.0000000000000000</HorizontalTailVolumeCoefficient>
244 <HorizontalTailAspectRatio>5.0000000000000000</HorizontalTailAspectRatio>
245 <VerticalTailVolumeCoefficient>0.0900000000000000</VerticalTailVolumeCoefficient>
246 <VerticalTailAspectRatio>1.6000000000000001</VerticalTailAspectRatio>
247 <VTTrailingEdgeOffset>0.0650000000000000</VTTrailingEdgeOffset>
248 <HTTrailingEdgeOffset>0.0620000000000000</HTTrailingEdgeOffset>
249 <TurboFanNacelleThicknessFraction>0.1200000000000000</TurboFanNacelleThicknessFraction>
250 <SingleEngineSpanLocation>0.3400000000000000</SingleEngineSpanLocation>
251 <EngineWingXOffset>0.6300000000000000</EngineWingXOffset>
252 <EngineWingZOffset>0.8500000000000000</EngineWingZOffset>
253 <TurboPropSection>N0012</TurboPropSection>
254 <UseAuxiliarySparForFuelTank>1.0000000000000000</UseAuxiliarySparForFuelTank>
255 <UsableFuelVolume>1.0000000000000000</UsableFuelVolume>
256 <FuelTankVolumeConstraint>8549.9504159228235949</FuelTankVolumeConstraint>
257 <MaxSpanConstraintActive>0.0000000000000000</MaxSpanConstraintActive>
258 <SpanEfficiency>0.8500000000000000</SpanEfficiency>
259 <SpanEfficiencyIncrement_LandingFlaps>0.1000000000000000</
SpanEfficiencyIncrement_LandingFlaps>
260 <SpanEfficiencyIncrement_TakeOffFlaps>0.0500000000000000</
SpanEfficiencyIncrement_TakeOffFlaps>
261 <CD0increment_LandingGear>0.0200000000000000</CD0increment_LandingGear>
262 <CD0increment_LandingFlaps>0.0650000000000000</CD0increment_LandingFlaps>
263 <CD0increment_TakeOffFlaps>0.0150000000000000</CD0increment_TakeOffFlaps>
264 <CD0increment_Compressibility>0.0005000000000000</CD0increment_Compressibility>
265 <CruiseReqMTOM>0.8500000000000000</CruiseReqMTOM>
266 <MaxContPowerSetting>0.9500000000000000</MaxContPowerSetting>
267 <MaxCLmaxLanding>4.0000000000000000</MaxCLmaxLanding>
268 <DefaultAbsoluteCeilingTurboFan>13500.0000000000000000</DefaultAbsoluteCeilingTurboFan>
269 <C2WeightMethod>Torenbeek</C2WeightMethod>
270 <UseClass25>0.0000000000000000</UseClass25>
271 <EnableFWE>0.0000000000000000</EnableFWE>
272 <ElevatorChordCoeff>0.2000000000000000</ElevatorChordCoeff>
273 <NumberOfCrew>7.0000000000000000</NumberOfCrew>
274 <FWEMaximumCabinDifferentialPressure>8.59999999999996</
FWEMaximumCabinDifferentialPressure>
275 <MinNoseGearLoad>0.0600000000000000</MinNoseGearLoad>
276 <MaxNoseGearLoad>0.1500000000000000</MaxNoseGearLoad>
277 <MainGearStowage mapType="vector">0.3500000000000000;0.5000000000000000</MainGearStowage>
278 <NoseGearStowage mapType="vector">0.0500000000000000;0.2000000000000000</NoseGearStowage>
279 <ScrapeAngleLOF>12.0000000000000000</ScrapeAngleLOF>
280 <MainGearStowageWidth>1.0000000000000000</MainGearStowageWidth>
281 </aircraftSettings>
282 </initiator>
283 </TUDInitiator>
284 </toolspecific>
285 </cpacs>
```

B.3. INITIATOR OUTPUT FILE

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <!-- ['MTOM'] -->
3 <Output>
4   <Output0>[75211.3046]</Output0>
5   <Output1>[75166.2119]</Output1>
6   <Output2>[75544.5211]</Output2>
7   <Output3>[83763.6369]</Output3>
8   <Output4>[83747.7931]</Output4>
9   <Output5>[83779.0325]</Output5>
10  <Output6>[83778.4501]</Output6>
11  <Output7>[83970.2531]</Output7>
12  <Output8>[84043.943]</Output8>
13  <Output9>[79336.8328]</Output9>
14  <Output10>[79342.0121]</Output10>
15  <Output11>[79334.685]</Output11>
16  <Output12>[79345.1196]</Output12>
17  <Output13>[86551.1521]</Output13>
18  <Output14>[86816.8862]</Output14>
19  <Output15>[86830.7193]</Output15>
20  <Output16>[87950.6801]</Output16>
21  <Output17>[88068.9741]</Output17>
22  <Output18>[89293.824]</Output18>
23  <Output19>[89290.9976]</Output19>
24  <Output20>[87313.3879]</Output20>
25  <Output21>[87302.9004]</Output21>
26  <Output22>[87058.2096]</Output22>
27  <Output23>[87094.5732]</Output23>
28  <Output24>[87113.0305]</Output24>
29  <Output25>[79819.6456]</Output25>
30  <Output26>[79767.6555]</Output26>
31  <Output27>[83934.1138]</Output27>
32  <Output28>[84011.3423]</Output28>
33  <Output29>[84010.9099]</Output29>
34  <Output30>[84041.5037]</Output30>
35  <Output31>[75727.1725]</Output31>
36  <Output32>[75431.4856]</Output32>
37  <Output33>[75421.7939]</Output33>
38  <Output34>[75424.224]</Output34>
39  <Output35>[75280.8245]</Output35>
40  <Output36>[80181.0953]</Output36>
41  <Output37>[87790.201]</Output37>
42  <Output38>[86590.2338]</Output38>
43  <Output39>[86603.2944]</Output39>
44 </Output>
45
```

C

COMPLETE XDSM OF INITIATOR OPTIMIZATION



D

CASE STUDY 2

D.1. OUTPUT FILE

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <!-- ['MTOM', 'FM'] -->
3 <Output>
4   <Output0>[75211.3046, 18312.8368]</Output0>
5   <Output1>[75166.2119, 18306.1503]</Output1>
6   <Output2>[75544.5211, 18304.0753]</Output2>
7   <Output3>[83763.6369, 19953.9702]</Output3>
8   <Output4>[83747.7931, 19953.9701]</Output4>
9   <Output5>[83779.0325, 19804.1268]</Output5>
10  <Output6>[83778.4501, 19838.7758]</Output6>
11  <Output7>[83970.2531, 22924.5309]</Output7>
12  <Output8>[84043.9430, 22741.9124]</Output8>
13  <Output9>[79336.8328, 22764.3869]</Output9>
14  <Output10>[79342.0121, 20730.6935]</Output10>
15  <Output11>[79334.6850, 15694.1716]</Output11>
16  <Output12>[79345.1196, 15688.8856]</Output12>
17  <Output13>[86551.1521, 15688.8856]</Output13>
18  <Output14>[86816.8862, 15688.8856]</Output14>
19  <Output15>[86830.7193, 15688.3687]</Output15>
20  <Output16>[87950.6801, 19983.6122]</Output16>
21  <Output17>[88068.9741, 19833.1922]</Output17>
22  <Output18>[89293.8240, 19814.7156]</Output18>
23  <Output19>[89290.9976, 19814.7156]</Output19>
24  <Output20>[87313.3879, 18101.4711]</Output20>
25  <Output21>[87302.9004, 18094.1121]</Output21>
26  <Output22>[87058.2096, 13230.4738]</Output22>
27  <Output23>[87094.5732, 13206.0277]</Output23>
28  <Output24>[87113.0305, 13202.7323]</Output24>
29  <Output25>[79819.6456, 13149.3188]</Output25>
30  <Output26>[79767.6555, 13152.2871]</Output26>
31  <Output27>[83934.1138, 14353.5754]</Output27>
32  <Output28>[84011.3423, 19755.8951]</Output28>
33  <Output29>[84010.9099, 19755.8951]</Output29>
34  <Output30>[84041.5037, 19881.6525]</Output30>
35  <Output31>[75727.1725, 19873.0654]</Output31>
36  <Output32>[75431.4856, 19880.144]</Output32>
37  <Output33>[75421.7939, 19880.144]</Output33>
38  <Output34>[75424.2240, 13204.0398]</Output34>
39  <Output35>[75280.8245, 13204.0397]</Output35>
40  <Output36>[80181.0953, 13352.4164]</Output36>
41  <Output37>[87790.2010, 13333.0062]</Output37>
42  <Output38>[86590.2338, 18334.0264]</Output38>
43  <Output39>[86603.2944, 19984.1754]</Output39>
44 </Output>
45
```


BIBLIOGRAPHY

- [1] E. Commission, *Directorate-general for mobility and transport, directorate-general for research and innovation, flightpath 2050 : Europe's vision for aviation : maintaining global leadership and serving society's needs*, Publications Office (2012).
- [2] G. G. Fleming and I. Lepinay, *Environmental trends in aviation to 2050*, International Civil Aviation Organization , 17 (2019).
- [3] F. Flager and J. A. Haymaker, *A comparison of multidisciplinary design, analysis and optimization processes in the building construction and aerospace industries*, 24th international conference on information technology in construction (2007).
- [4] K. G. Bowcutt, *A perspective on the future of aerospace vehicle design*, 12th AIAA International Space Planes and Hypersonic Systems and Technologies (2003).
- [5] P. Piperni, A. DeBlois, and R. Henderson, *Development of a multilevel multidisciplinary-optimization capability for an industrial environment*, AIAA journal **51** (2013).
- [6] P. D. Ciampa and B. Nagel, *The agile paradigm: the next generation of collaborative mdo*, 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2017).
- [7] P. D. Ciampa, P. S. Prakasha, F. Torrigiani, J.-N. Walther, T. Lefebvre, N. Bartoli, H. Timmermans, P. Della Vecchia, and D. Stingo, L.and Rajpal, *Streamlining cross-organizational aircraft development: Results from the agile project*, AIAA Aviation 2019 Forum (2019).
- [8] P. D. Ciampa and B. Nagel, *Agile paradigm: The next generation collaborative mdo for the development of aeronautical systems*, Progress in Aerospace Sciences (2020).
- [9] I. van Gent, B. Aigner, B. Beijer, J. Jepsen, and G. La Rocca, *Knowledge architecture supporting the next generation of mdo in the agile paradigm*, Progress in Aerospace Sciences (2020).
- [10] R. Belie, *Non-technical barriers to multidisciplinary optimisation in the aerospace industry*, 9th AIAA/ISSMO Symposium of Multidisciplinary Analysis and Optimisation (2002).
- [11] I. van Gent and G. La Rocca, *Formulation and integration of mdao systems for collaborative design: A graph-based methodological approach*, Aerospace Science and Technology **90** (2019).
- [12] D. J. Pate, J. Gray, and B. J. German, *A graph theoretic approach to problem formulation for multidisciplinary design analysis and optimization*, Structural and Multidisciplinary Optimization **5** (2014).
- [13] I. van Gent, *Agile MDAO Systems: A Graph-based Methodology to Enhance Collaborative Multidisciplinary Design*, Ph.D. thesis, Delft University of Technology (2019).
- [14] A.-L. Bruggeman, *Automated Execution Process Formulation using Sequencing and Decomposition Algorithms for Collaborative MDAO*, Master's thesis, Delft University of Technology (2019).
- [15] A. B. Lambe and J. R. R. A. Martins, *Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes*, Structural and Multidisciplinary Optimization (2012).
- [16] M. Alder, E. Moerland, J. Jepsen, and B. Nagel, *Recent advances in establishing a common language for aircraft design with cpacs*, Aerospace Europe Conference (2020).
- [17] I. van Gent, G. La Rocca, and M. F. M. Hoogreef, *Cmdows: A proposed new standard to store and exchange mdo systems*, Aerospace Europe 6th CEAS Conference (2017).

- [18] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, *Openmdao: An open-source framework for multidisciplinary design, analysis, and optimization*, Structural and Multidisciplinary Optimization, 1075 (2019).
- [19] D. de Vries, *Towards the Industrialization of MDAO*, Master's thesis, Delft University of Technology (2017).
- [20] D. Seider, P. M. Fischer, M. Litz, and A. Gerndt, *Open source software framework for applications in aeronautics and space*, IEEE Aerospace Conference (2012).
- [21] G. Karypis and V. Kumar, *A fast and high quality multi level scheme for partitioning irregular graphs*, SIAM Journal on scientific Computing (1999).
- [22] A. Saltelli, S. Tarantola, and F. Campolongo, *Sensitivity analysis as an ingredient of modeling*, Statistical Science **15**, 377 (2000).
- [23] B. Iooss and P. Lemaître, *A Review on Global Sensitivity Analysis Methods*, Vol. 59 (Springer, 2015) Chap. Dellino G., Meloni C. (eds) Uncertainty Management in Simulation-Optimization of Complex Systems.
- [24] L. Brevault, M. Balesdent, N. Bérend, and R. Le Riche, *Comparison of different global sensitivity analysis methods for aerospace vehicle optimal design*, 10th World Congress on Structural and Multidisciplinary Optimization (2013).
- [25] P. M. Reed, A. Hadjimichael, K. Malek, T. Karimi, C. Vernon, V. Srikrishnan, R. Gupta, D. Gold, B. Lee, K. Keller, T. Thurber, and J. Rice, *Addressing uncertainty in multisector dynamics research*, (2022).
- [26] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer* (John Wiley, 2008).
- [27] A. Saltelli, K. Aleksankina, W. Becker, P. Fennell, F. Ferretti, N. Holst, S. Li, and Q. Wu, *Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices*, Environmental Modelling and Software (2019).
- [28] A. Saltelli and P. Annoni, *How to avoid a perfunctory sensitivity analysis*, Environmental Modelling and Software (2010).
- [29] A. Saltelli, K. Chan, and E. Scott, *Sensitivity analysis*, Wiley (2000).
- [30] J. C. Helton and F. J. Davis, *Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems*, Reliability Engineering & System Safety (2003).
- [31] I. M. Sobol and F. J. Davis, *On the distribution of points in a cube and the approximate evaluation of integrals*, USSR Computational Mathematics and Mathematical Physics (1967).
- [32] C. Lemieux, *Monte carlo and quasi-monte carlo sampling*, Springer (2009).
- [33] H. Niederreiter, *Quasi-monte carlo methods and pseudo-random numbers*, Bulletin of the American Mathematical Society (1978).
- [34] M. D. Morris, *Factorial sampling plans for preliminary computational experiments*, Technometrics **33**, 161 (1991).
- [35] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity analysis in practice: A guide to assessing scientific models*, Vol. 1 (Wiley, 2004).
- [36] I. M. Sobol, *Sensitivity analysis for non-linear mathematical models*, Mathematical modelling and computational experiment **1**, 407 (1993).
- [37] I. M. Sobol, *Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates*, Mathematics and Computers in Simulation (2001).
- [38] M. J. W. Jansen, *Analysis of variance designs for model output*, Computer Physics Communications (1999).

- [39] T. Homma and A. Saltelli, *Importance measures in global sensitivity analysis of nonlinear models*, Reliability Engineering & System Safety (1996).
- [40] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, *Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index*, Computer Physics Communications (2010).
- [41] M. Duler, X. Roboam, B. Sareni, S. Touhami, Y. Lefevre, and J. Llibre, *Sensitivity analysis of a hybrid-electric aircraft powertrain based on sobol indices*, Aerospace Europe Conference, Bordeaux (2020).
- [42] A. Tremolet, J. Gauvrit-Ledogar, L. Brevault, S. Defoort, F. Morel, and J. Llibre, *Multidisciplinary overall aircraft design and optimization of blended wing body configurations*, 8th European Conference for Aeronautics and Space Sciences (2019).
- [43] R. I. Cukier, C. M. Fortuin, K. E. Shuler, A. G. Petschek, and J. H. Schaibly, *Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients*, The Journal of Chemical Physics **59**, 3873 (1973).
- [44] X. Chonggang and G. Gertner, *Understanding and comparisons of different sampling approaches for the fourier amplitudes sensitivity test (fast)*, Computational Statistics & Data Analysis **55**, 184 (2011).
- [45] A. Saltelli, S. Tarantola, and K. P.-S. Chan, *A quantitative model-independent method for global sensitivity analysis of model output*, Technometrics **41**, 39 (1999).
- [46] M. A. Thierry and S. Tarantola, *Application of global sensitivity analysis of model output to building thermal simulations*, Building Simulation **1**, 290 (2008).
- [47] H.-U. Park, J. Chung, K. Behdinan, and J.-W. Lee, *Multidisciplinary wing design optimization considering global sensitivity and uncertainty of approximation models*, Journal of Mechanical Science and Technology **28** (2014).
- [48] P. Y. Papalambros and J. W. Douglass, *Principles of optimal design: modeling and computation*, 2nd ed. (Cambridge University Press, 2000).
- [49] J. R. R. A. Martins and J. Hwang, *Review and unification of methods for computing derivatives of multidisciplinary computational models*, AIAA Journal **51**, 2582 (2013).
- [50] I. Jocelyn, R. Haftka, and H. Adelman, *Selecting step sizes in sensitivity analysis by finite differences*, National Aeronautics and Space Administration, Scientific and Technical Information Branch **86382** (1985).
- [51] T. Iwanaga, W. Usher, and J. Herman, *Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses*, Socio-Environmental Systems Modelling **4** (2022).
- [52] J. Herman and W. Usher, *SALib: An open-source python library for sensitivity analysis*, The Journal of Open Source Software **2** (2017).
- [53] A. Saltelli and R. Bolado, *An alternative way to compute fourier amplitude sensitivity test (fast)*, Computational Statistics Data Analysis (1998).
- [54] N. Giglioli and A. Saltelli, *Simlab 2.2 reference manual*, Institute for Systems Informatics and Safety (Joint Research Centre, European Commission) (2008).
- [55] J. C. Helton, F. J. Davis, and J. D. Johnson, *A comparison of uncertainty and sensitivity analysis results obtained with random and latin hypercube sampling*, Reliability Engineering & System Safety (2005).
- [56] R. Rosolem, H. V. Gupta, W. J. Shuttleworth, X. Zeng, and L. G. Gonçalves de Gonçalves, *A fully multiple-criteria implementation of the sobol method for parameter sensitivity analysis*, Journal of Geophysical Research Atmosphere (2012).
- [57] T. Ishigami and T. Homma, *An importance quantification technique in uncertainty analysis for computer models*, Proceedings. First International Symposium on Uncertainty Modeling and Analysis (1990).

- [58] D. Zhang, B. Song, P. Wang, and Y. He, *Performance evaluation of mdo architectures within a variable complexity problem*, Mathematical Problems in Engineering (2017).
- [59] R. Elmendorp, *Synthesis of Novel Aircraft Concepts for Future Air Travel*, Master's thesis, Delft University of Technology (2014).
- [60] R. Elmendorp, R. Vos, and G. La Rocca, *A conceptual design and analysis method for conventional and unconventional airplanes*, ICASICAS 2014: Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences , 1–12 (2014).
- [61] E. Torenbeek, *Synthesis of subsonic airplane design*, Delft University Press (1982).
- [62] A. Elham, G. La Rocca, and M. J. L. Van Tooren, *Development and implementation of an advanced, design-sensitive method for wing weight estimation*, Aerospace Science and Technology (2013).
- [63] D. Altman, D. Machin, T. Bryant, and M. Gardner, *Statistics with confidence: confidence intervals and statistical guidelines*, British Medical Journal Books (2000).