



HAL
open science

Bi-level architectures and multi-fidelity algorithms for multidisciplinary optimization in high dimensions.

Yann David

► **To cite this version:**

Yann David. Bi-level architectures and multi-fidelity algorithms for multidisciplinary optimization in high dimensions.. Mathematics [math]. Université de Toulouse, 2024. English. NNT : 2024TL-SEI014 . tel-04871572

HAL Id: tel-04871572

<https://theses.hal.science/tel-04871572v1>

Submitted on 7 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Doctorat de l'Université de Toulouse

préparé à l'INSA Toulouse

Architectures bi-niveau et algorithmes multi-fidélité pour
l'optimisation multidisciplinaire en haute dimension

Thèse présentée et soutenue, le 31 octobre 2024 par

Yann DAVID

École doctorale

EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité

Mathématiques et Applications

Unité de recherche

IMT : Institut de Mathématiques de Toulouse

Thèse dirigée par

Aude RONDEPIERRE

Composition du jury

M. Serge GRATTON, Président, Toulouse INP

M. Michael KOKKOLARAS, Rapporteur, McGill University

M. Fernass DAOUD, Rapporteur, Technische Universität München

M. Christophe BLONDEAU, Examineur, ONERA

Mme Aude RONDEPIERRE, Directrice de thèse, INSA Toulouse

Membres invités

M. Joel BREZILLON, AIRBUS

M. François GALLARD, IRT Saint Exupéry

Bi-level architectures and multi-fidelity algorithms for multidisciplinary optimization in high dimensions.

Yann David

Supervised by:

Aude Rondepierre, Institut National des Sciences Appliquées de Toulouse

François Gallard, IRT Saint Exupéry

Remerciements

La thèse pouvant être une entreprise solitaire, elle est aussi le fruit de discussions et de collaborations. Ces quelques lignes sont l'occasion pour moi de remercier les personnes qui ont participé, de près ou de loin, à l'élaboration de cette thèse.

Je tiens tout d'abord à remercier les membres du jury. Je remercie Serge Gratton de m'avoir fait l'honneur de présider celui-ci. Je remercie également Michael Kokkolaras et Fernass Daoud d'avoir accepté d'être rapporteurs de cette thèse, ainsi que pour leurs retours constructifs. Je remercie enfin les autres membres du jury, Christophe Blondeau et Joel Brezillon, pour leurs questions pertinentes et l'intérêt qu'ils ont porté à mes travaux.

Je remercie chaleureusement Aude Rondepierre et François Gallard pour m'avoir encadré durant ces quelques années. Je tiens à les remercier pour leur expertise et leurs précieux conseils sans lesquels cette thèse n'aurait sans doute jamais vu le jour, ainsi que pour leur grande gentillesse dont ils ont fait preuve à mon égard. J'ai toujours pris grand plaisir à assister à nos différentes réunions et j'en suis toujours ressorti avec des idées plein la tête. Je me considère comme quelqu'un de chanceux d'avoir pu travailler sous leur direction.

Je remercie également Anne Gazaix ainsi que l'ensemble de l'équipe MDA-MDO, qui s'est bien agrandie depuis mon arrivée, pour les discussions du quotidien et les bons moments partagés. Travailler à l'IRT a toujours été un plaisir et j'y ai toujours trouvé l'aide dont j'avais besoin. J'étends ces remerciements à l'ensemble des collaborateurs de l'IRT avec qui j'ai eu la chance de discuter, en particulier les collègues du club de jeux de société qui m'ont offert une véritable bouffée d'oxygène pendant la dernière année.

Sur le plan personnel, je remercie mes parents et, plus largement, l'ensemble de ma famille pour le soutien et l'amour dont ils font preuve à mon égard.

Enfin, je souhaite dédier ces derniers mots de remerciement à mes amis toulousains qui m'accompagnent depuis maintenant 7 ans. Pour tous les bons moments partagés, ainsi que pour leur soutien sans faille dans les moments de réussite comme dans les moments de doute.

Contents

Nomenclature	4
Abbreviations	5
1 Introduction	7
2 Preliminaries	14
2.1 Multidisciplinary Design Optimization	15
2.1.1 MDO's general concepts	15
2.1.2 MDO architectures	17
2.2 Mathematical background	23
2.2.1 The Gauss-Seidel algorithm and local convergence	23
2.2.2 Direct and adjoint methods for gradient computation	29
2.2.3 Optimality conditions	31
2.2.4 Three analysis theorems	33
3 A bi-level architecture	37
3.1 Introduction	39
3.2 State of the art: distributed architectures for large scale MDO problem	41
3.2.1 Distributed architectures	41
3.2.2 Bi-level optimization	51
3.3 From MDF to a bi-level decomposition	57
3.3.1 An equivalent bi-level decomposition	58
3.3.2 Regularity of the system level functions	60
3.4 A solution algorithm for the lower optimization problem	62
3.4.1 The Block Coordinate Descent algorithm	66
3.4.2 Convergence analysis of the BCD-MDF algorithm	70
3.5 Variants of the BCD-MDF algorithm	80
3.5.1 With linear approximation of the constraints	80
3.5.2 A weakly coupled variant	81
3.5.3 Adding target values for difficult couplings	85
3.6 Numerical experiments	86
3.6.1 Discrepancy reduction and local convergence comparisons on SSBJ	86
3.6.2 Scalability study	92
4 A multi-fidelity framework	95
4.1 Introduction	97
4.2 State of the art: multi-fidelity in MDO	100

4.2.1	The multi-fidelity approaches	100
4.2.2	Multi-fidelity methods in numerical optimization	103
4.2.3	Multi-fidelity applied to MDO	106
4.3	Down-selecting fidelity models for MDA	110
4.3.1	Fidelity levels for MDA	111
4.3.2	Two criteria for error estimation	114
4.3.3	Pareto front and post treatment	122
4.4	Multi-fidelity methodologies validation	123
4.4.1	A multi-fidelity refinement framework	124
4.4.2	Coupled adjoint criterion	125
4.4.3	Gradient alignment criterion	130
4.4.4	A multi-fidelity application using bi-level architectures	133
5	Conclusions and perspectives	137
5.1	Contributions	137
5.2	Perspectives	139
A	The Sobieski Super Business Jet (SSBJ) test case	154
B	The Sellar problem and analytical expressions for the two-block variant	156
B.1	The original Sellar problem	156
B.2	Analytical expressions for the two-block Sellar	157
B.2.1	Couplings expressions	157
B.2.2	Couplings's first and second derivatives	158
B.2.3	Objective and constraints's gradient	158
B.2.4	Objective and constraint's Hessian	158
C	Notes on COBYLA	159
C.1	Algorithm	159
C.2	Convergence properties	159

Nomenclature

Notation	Description
p	Number of disciplines
x_0	Shared variables
x_i	Local variables of discipline i
y	Coupling variables
y^t	Target value for coupling variable y
u_i	Internal state of discipline i
r_i	Residual of the governing state equation of discipline i
f	Objective function
g, h	Inequality/Equality constraint
ϕ_i	Disciplinary analysis of discipline i , computes y_i
Ψ	Multi-disciplinary analysis
\mathcal{L}	Lagrangian
λ, μ	Lagrange multipliers
$w = [x, \lambda]$	Primal-Dual pair, $w = [x, \lambda]$
$x_{i:j}$	$x_{i:j} = [x_i, x_{i+1}, \dots, x_{j-1}, x_j]$ $j \geq i$
$x_{\neq i}$	$x_{\neq i} = [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p]$
x^*	Optimal value for variable x
x^k	k^{th} iterate of variable x
x^T	Transpose of vector x
A^{-1}	Inverse of matrix A
ϵ	Error or tolerance
\tilde{m}	Surrogate/approximation of model m
F_i	Number of fidelity models for discipline i
\mathcal{V}	A combination of fidelity models $\mathcal{V} = [\mathcal{V}_1, \dots, \mathcal{V}_p] \in \prod_{i \in [1, p]} [1, F_i]$
$\tilde{\phi}_{i,j}$	Disciplinary analysis of discipline i with fidelity j

Abbreviations

MDO

MDO	Multi-disciplinary Design Optimization
MDA	Multi-disciplinary Analysis
GEMSEO	Generic Engine for Multidisciplinary Scenarios Exploration and Optimization
XDSM	eXtended Design Structure Matrix
CFD	Computational Fluid Dynamics
CSM	Computational Structural Mechanics
SSBJ	Sobieski Super Business Jet

Architectures

AAO	All-At-Once
SAND	Simultaneous Analysis and Design
MDF	Multi-Disciplinary Feasible
IDF	Individual Discipline Feasible
CO	Collaborative Optimization
ATC	Analytical Target Cascading
BLISS	Bi-Level Integrated System Synthesis
BL-IRT	Bi-Level IRT
BL-BCD-MDF	Bi-Level - Block Coordinate Descent - Multi-Disciplinary Feasible
BL-BCD-WK	Bi-Level - Block Coordinate Descent - Weakly coupled

Maths

IFT	Implicit Function Theorem
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
MFCQ	Mangasarian-Fromovitz Constraint Qualification
SCQ	Slater Constraint Qualification
NLP	Non-Linear Programming
DFO	Derivative-Free Optimization

SBO	Surrogate Based Optimization
BCD	Block Coordinate Descent
UBM	Unique Block Minimizer
FOS	First Order Separability
MF	Multi-Fidelity
HF	High-Fidelity
LF	Low-Fidelity

Chapter 1

Introduction

With the constant need for improvement in aircraft design, whether to maximize the efficiency of transport aviation or simply to reduce development costs, the need for new methodological and numerical approaches is ever present. Since aviation is in many ways a complex and high-stakes field, many numerical and economic challenges are met. This is all the more true since the field of aeronautics is inherently multidisciplinary, meaning that a number of fields, with their own experts and constraints, are involved in the elaboration of the product.

In this context, the so-called multidisciplinary design optimization (MDO) emerged between the 1970s and 1990s, originally driven by the structural optimization community, which incorporated models from other disciplines [66, 132]. MDO is a model-based automated design method that facilitates the integration of different models from multiple disciplines to account for all interdependencies, commonly called couplings. This method involves numerous decisions related to the definition of the optimization problem, which models to run, or in what order, which strongly influence the overall behavior of the optimization. This highly organizational aspect of optimization led to the definition of the first MDO architectures [32, 101], strategies for solving multidisciplinary problems. They are often defined as a combination of a reformulated optimization problem, which changes the definition of the optimization problem to be solved, a dataflow description, which shows how data is exchanged between components, and a workflow, which illustrates the computational sequence. The first MDO results in the field of aeronautics were initially focused on the design of aircraft wings [8], then rapidly extended to the whole aircraft [82]. The same methods now extend to many areas of engineering, from automotive design [9] to wind turbine design [77].

While MDO methods were initially designed for coupling simple models, such as pre-project model optimization, recent advances in numerical optimization are driving the integration of these methods into more advanced phases of conception where the models under consideration are more numerous, finer grained, and therefore more costly. This is all the more true today, with the emergence of high performance solvers that have reached maturity [133], in particular CFD (Computational Fluid Dynamics) [134] and CSM (Computational Structural Mechanics) [94] solvers, the problem of their integration into the MDO process is all the more important to take full advantage of their efficiency.

In 1997, *Sobieski* [141] already gave an overview of the progress in MDO and future

perspectives in the field of aircraft design. These perspectives, which can be generalized to other fields of engineering, can be summarized in three axes:

1. Sensitivity analysis to enable efficient implementation of gradient-based methods.
2. Advances in problem decomposition methods to accommodate the organizational constraints of industry.
3. The development of approximate substitution models to reduce restitution costs.

All three of these areas have been the subject of considerable research in recent years, leading to major advances in the field. In particular, the first point, concerning the efficient computation of derivatives, has made great strides in MDO in recent years with the advent of the coupled adjoint [103, 104, 87]. First introduced in MDO within a variant of the Bi-Level Integrated System Synthesis (BLISS) distributed architecture [139], the coupled adjoint has been further experimented with in combination with CFD/CSM solvers and is now increasingly common within high-fidelity MDO frameworks and methods. This progress is all the more remarkable in high-fidelity optimization, where gradient-based algorithms combined with the coupled adjoint offer the best scalability [103]. An obstacle to its systematic use remains the difficulty of implementing the coupled adjoint in an industrial context involving many disciplines [59]. This is mainly due to the fact that many powerful models are either sold as black-box Commercial Off-The-Shelf (COTS) software where gradients are missing (e.g. CAD engine or model generation when morphing is impossible), or are one of many in-house simulation tools that may be too expensive to differentiate. In this regard, automatic differentiation may be the next breakthrough for heavy industrial codes, provided the source code is available [71, 51]. A direct consequence of these difficulties is that gradient-free optimization techniques must be considered instead of their gradient counterparts, even though the latter are known to be much more efficient for large-scale problems. Otherwise, when derivatives are mandatory, finite difference methods are still widely used by default, despite their high cost and lack of precision.

The second line of research, dedicated to decomposition methods, is also very active and will lead in particular to so-called "distributed" architectures [9, 101, 147]. Unlike more traditional monolithic architectures, these architectures define multiple optimization subproblems. In this respect, they try to bring their design closer to the industrial separation of computational tools, in particular by giving more autonomy to each of them. These approaches greatly facilitate the implementation of MDO processes, generally at the cost of a loss of efficiency and a deterioration of convergence properties compared to monolithic architectures [101]. Moreover, the use of gradient-based optimization algorithms is complicated by the need to compute post-optimal coupled derivatives of the various subproblems, i.e. the sensitivities of their optimal value with respect to some parameter, which requires costly or sometimes unavailable second derivatives [139] and further assumptions on the subproblems and their solutions, such as the uniqueness of the solution, a constant set of active constraints, or the verification of the second-order Karush-Kun-Tucker (KKT) conditions [47]. While distributed architectures are used in the industry for their ease of implementation, to our knowledge there is no consensus on a distributed architecture that is competitive with monolithic architectures, yet robust enough to handle a wide range of problems. Such architectures would be highly desirable in the coming years, but for now, despite efforts in this direction, they are still considered to lack convergence properties [146, 148, 147].

Finally, the development and integration of surrogate models into MDO processes is also very popular. Surrogate-based optimization (SBO) is gaining ground in MDO as design methodologies evolve, via the elaboration of increasingly reliable surrogate models that replace models deemed too costly [135, 137, 140, 136, 115]. Many of these surrogate based methods within MDO need to be updated during optimization with high-fidelity model evaluations. Another solution is to use so-called "one-shot" surrogates [48] that are built before optimization begins. These models are therefore less reliable than those built directly into the framework, but this is offset by the fact that, once built, they can be more easily shared and reused within multiple frameworks or applications. The main limitation of purely surrogate-based approaches is that they all face, to some extent, the so-called "curse of dimensionality" [46, 116], which typically prevents their efficient use on the high-dimensional and non-linear problems considered in this manuscript, which may involve thousands of variables and/or constraints [55]. In this regard, even if the training cost can be reduced [21], the lack of precision of the model is still present. In this context, and also more recently, multi-fidelity methods have emerged within the MDO community [45]. These approaches rely on the use of a so-called high-fidelity model, which has the desired accuracy but a high computational cost, and one or more low-fidelity models, which trade accuracy for a shorter execution time. They define a model management framework that describes which model to run and when, as well as what data to exchange, in order to reduce computation time while achieving the same solution as using only the high-fidelity model. While multi-fidelity methods [46, 116] have shown great results for monodisciplinary optimization [116], the literature in MDO on this particular topic is still sparse [158]. Moreover, most of the existing work does not take into account the specificity of MDO, such as the disciplinary separation or the organizational aspect with the so-called MDO architectures.

These three lines of research give us great hope for the future of MDO, as many improvements can be expected from them, especially when considering high-dimensional problems with numerous disciplines and coupling variables. The present work proposes to contribute to these advances by studying two methodologies for MDO in large dimensions, with emphasis on the methodological and mathematical aspects of the problems encountered, while maintaining a link, from the point of view of hypotheses, to the industrial environment.

The first one is dedicated to the development of a new distributed architecture, the bi-level - block coordinate descent (BL-BCD) architecture. The proposed architecture has its origins in the aforementioned BLISS architecture developed by *Sobieski and al.* [139] and the further research done at ONERA [2, 84, 131] on the bi-level MDO architectures, recently continued at IRT Saint Exupéry [55]. In this newly proposed bi-level architecture, a system optimization problem minimizes the objective with respect to the shared variables using a derivative-free algorithm, with each iteration requiring the solution of the so-called lower optimization problem, which is nested within the first optimization problem that optimizes the local variables. The contributions in this are multiple:

- The first contribution in this area is the mathematical formalism of this bi-level approach in the context of MDO, where the regularity of the considered functions can be proved under the assumption that the lower-level optimization problem is solved with sufficient precision. In particular, this bi-level framework does not require system-level derivatives and is based on the correct solution of the lower-level

optimization problem.

- The second input is the proposal of an algorithm to solve the lower-level optimization problem, namely the Block Coordinate Descent (BCD) algorithm. [16, 64, 110, 111, 113, 17, 18], which we prove to be locally convergent under industrially justifiable assumptions. Since this algorithm is based on a block decomposition, well-chosen gradient-based algorithms can be used to solve each block of local variables, while the coupled adjoint for the entire lower-level optimization problem is not necessary.
- Three variants of this newly proposed bi-level architecture are also proposed to provide some flexibility in both the assumptions and the organizational aspect of the MDO framework.
- Finally, applications to classical MDO problems are given to illustrate the pros and cons of the considered approaches and to show the gain in terms of convergence of the overall architecture and scalability with respect to the size of the lower-level optimization problem. The scalability test is performed on a newly created scalable problem that extends the well-known Sellar problem[135] to a two-block version with n -dimensional local variables, coupling variables, and constraints.

The second methodology considers multi-fidelity approaches within an MDO environment. Considering the organizational aspects in MDO and the numerous models involved, many fidelity of the MDO process can emerge from choices on the fidelity for each discipline, value for hyperparameters such as a thinner, or coarser, tolerances, or a change in the architecture considered. In particular, the fidelity of a given model is generally unknown *a priori* because of the coupling variables. The questions of how many fidelity models to consider and, if a choice have to be made, how to select the most interesting ones have raised little interest at the moment within the MDO community. If the former question is generally assumed to be known beforehand , i.e. fixed by the authors [93, 96, 122], or left to the reader to consider more general multi-fidelity approaches with an arbitrary number of levels [108, 156, 158], and hence is generally not considered, the latter, focusing on the classification of fidelity models, already have some elements of response. Two works have adapted the classical mono-disciplinary methodology, in which a (*cost, error*) Pareto front of the fidelity models is built, to consider more general MDO problems [28, 158]. The considered strategy are promising in which they allow to discriminate several fidelity models but they either rely on heavy statistics to estimate the error, or this estimation is directly incorporated within the multi-fidelity framework. This raises a problem in terms of scalability of the methods if the models are numerous and/or high-dimensional. Therefore there is a need for new approaches to classify the numerous possible fidelity models available in a MDO context at an affordable cost. Here the contributions can be described as follows:

- The main contribution in this area is the introduction of two sensitivity criteria, one based on the coupled adjoint and the other on the gradient alignment, for estimating the fidelity of many different models. These two criteria are designed to be used prior to multi-fidelity optimization, to be non-intrusive except for access to the gradients, and to be able to approximate the error of an exponentially large number of fidelity models in a reasonable amount of time. They are subject to approximation errors, but it is expected that this classification will not prevent a regular multi-fidelity algorithm, such as the multi-fidelity refinement approach used

in [108] and considered in this manuscript, to show some gain compared to using only the high-fidelity model.

- A methodology for constructing the $(cost, error)$ Pareto front and down-selecting the fidelity models is also given, as well as a post-processing algorithm that eliminates selected fidelity models that are deemed to be too similar in their cost-precision ratio.
- Several application cases are presented to illustrate the benefits of the approach. Among them is the perturbed Sellar [135] optimization problem, where the fidelity of the models can be modified by changing one parameter, allowing for numerous fidelity generations. Then, an application based on the classic Sobieski Super Business Jet (SSBJ) test case [139] is presented, where low-fidelity models neglect the backward effects coming from the couplings, hence a fidelity is defined as an order of execution for the disciplines. In all of the above experiments, the sensitivity criteria yields a satisfactory fidelity selection that results in a speedup of the multi-fidelity optimization that is equal to, or at least very close to, the optimal speedup at a reasonable cost for error estimation.
- In conjunction with the previous chapter, a multi-fidelity optimization is performed using bi-level architectures as fidelity levels. This last application is intended to illustrate the various links between all the work presented in this manuscript.

These two methodologies fall into the aforementioned axes of research for improvement in MDO, more precisely on the last two. Since this work keeps its emphasis on the high dimensional MDO problem, the availability of the coupled adjoint is still discussed throughout the manuscript. Alternatives are proposed when it is not possible to benefit from its efficiency.

Considering the implementation of the aforementioned methodologies, all the experiments are mainly based on the open source Generic Engine for Multidisciplinary Scenarios, Exploration and Optimization (GEMSEO) [53] Python library. In particular, both lines of research have led to their respective GEMSEO packages:

- The first GEMSEO package concerns the bi-level framework presented within the manuscript. It implements the BL-BCD-MDF architecture and one of its variants, the BL-BCD-WK architecture, that can be automatically created from a set of provided disciplines. All aspects of these architectures are highly customizable, from practical considerations, with the various tolerances, optimization algorithms, and constraints involved that can be set separately, to a more macro view, with architectural changes such as the ability to modify the block decomposition of the lower-level optimization problem. A two-block modification of the original Sellar problem [135] with the ability to scale any design vector, coupling variables and constraints is also provided for testing generic bi-level architectures based on a block decomposition, in particular its ability to handle a high-dimensional lower-level optimization problem.
- The second embeds the different tools and methods for the multi-fidelity MDO. Both adjoint-based and gradient-based criteria are implemented, computing an approximation of the error for each fidelity model provided. The creation of the Pareto front, the down-selection algorithm and its post-processing are also present

in the package. A Sellar optimization problem with configurable fidelity for each discipline is also provided to generate a high number of fidelities and to illustrate the capability of the criteria when the combinatorial explosion of the number of models appends.

These implementations aim to facilitate many aspects of the development, first the testing of the different methodologies with the various optimization algorithms and application cases available, as well as the reproducibility of the results presented in this manuscript by using a unified framework or their industrial transfer. These implementations were the opportunity to port a previous multi-fidelity package, developed by *Romain Olivanti* [108], which already contained several multi-fidelity algorithms and tools for MDO, in order to be compatible with the latest version of GEMSEO. It is from this package that comes the multi-fidelity refinement algorithm used within the multi-fidelity application of this manuscript.

The present manuscript describes the aforementioned development of methodologies and mathematical tools dedicated to the implementation of MDO strategies for the optimization of high-dimensional problems. For this purpose, the thesis is divided into 5 chapters. The Chapter 2 begins with some of the elements that make up MDO, along with a brief overview of monolithic architectures. The second half deals with the various mathematical tools and theorems used throughout this manuscript. Classical results and proofs for the Gauss-Seidel method and its convergence are given in preparation for an important result within Chapter 3. A brief reminder on the coupled adjoint and classical nonlinear optimality conditions is also present. The Chapter 3 explores distributed architectures and in particular their convergence properties. A new bi-level architecture is presented, following the work of [55] with the introduction of the block coordinate descent algorithm to solve the lower optimization problem. In the Chapter 4, the fidelity level selection for multi-fidelity algorithms is discussed. The two aforementioned criteria for fast fidelity estimation are described and tested on small test cases. The last application illustrates the implementation and its capabilities by performing a multi-fidelity optimization using bi-level architectures as fidelity models. Finally, the Chapter 5 lists the various contributions discussed in the manuscript. A final step back enumerates some perspectives for future research and improvements.

Associated communications

- Yann David, Aude Rondepierre, and François Gallard. *Bi-level architectures, elements of convergence proof* (2022, September 20-21). 3rd European Workshop on MDO ” MDO for Industrial Applications in Aeronautics – Towards Greener Aviation ”.
- Yann David, Aude Rondepierre, and François Gallard. *Locally convergent bi-level MDO architectures based on the block coordinate descent algorithm*. Soon to be submitted at the end of October at Journal of Optimization Theory and Applications (JOTA).

Chapter 2

Preliminaries

Summary: This chapter provides an overview of the main concepts, notations, and concerns of MDO. A brief overview of classical monolithic architectures is presented at the end of this reminder.

The second half of the chapter contains a detailed description of the objects, concepts, and tools used throughout the manuscript. The first section is devoted to classical convergence results for the Gauss-Seidel method, immediately followed by implicit analytic methods for gradient computation, optimality conditions for nonlinear programming, and finally three well-known analysis theorems.

Résumé: Ce chapitre présente une vue d'ensemble des principaux concepts, notations et préoccupations en MDO. Un bref aperçu des architectures monolithiques classiques est présenté à la fin de ce rappel.

La seconde moitié du chapitre contient une description détaillée des objets, des concepts et des outils utilisés tout au long du manuscrit. La première section est consacrée aux résultats de convergence classiques pour la méthode de Gauss-Seidel, immédiatement suivie par les méthodes analytiques implicites pour le calcul du gradient, les conditions d'optimalité pour l'optimisation non linéaire, et enfin trois théorèmes d'analyse bien connus.

2.1 Multidisciplinary Design Optimization

The concept of multidisciplinary design optimization (MDO) emerged during the 1980's as a response to the need to design complex systems that require multiple areas of expertise. Initially introduced in aerospace design for aircraft wing design [8] and then for a whole aircraft [82], the MDO methodologies have been recently extended to other engineering fields such as automobile design [9], or wind turbines design [77].

MDO is an engineering domain that integrates numerical methods, engineering tools and mathematical modeling to address design problems involving multiple fields of expertise, commonly referred to as disciplines. These disciplines often arise from the various physical phenomena involved, such as aerodynamics or structures [103, 102]. The underlying principle of MDO is to leverage the efficiency of existing codes, which have been developed to operate independently, to enhance the quality of the solution, the robustness of the process, or to reduce the overall cost.

The different models involved in a MDO process are generally expensive to evaluate and coupled (interdependent) in terms of the overall design. Consequently, the MDO problems require efforts to ensure efficient communication and coherence between disciplines, which increases the cost and complexity of the design process. In this regard, MDO is challenging in many respects, and the MDO domain has developed its own taxonomy and strategies to overcome these specific issues.

2.1.1 MDO's general concepts

As the name suggests, Multidisciplinary Design Optimization (MDO) encompasses design optimization problems that necessitate the integration of multiple disciplines. A discipline is a scientific domain or a specialized field that has an impact on the final design. The disciplines are often regarded as self-sufficient, by the specialists who develop them, and thus by design. Historically, in aircraft design, considered disciplines have been derived from the physics involved : common examples are aerodynamics, structure or propulsion. Recently, disciplines from other domains have also been considered and coupled in MDO processes. For instance with the emergence of environmental concern, aviation starts to take into account other metrics (economical, ecological) [125, 123]. Such a compromise leads to consider multi-objectives functions (e.g. sustainable aviation) and may necessitate the construction of a Pareto front.

In our context, a discipline can be conceptualized as a black box piece of software that takes a series of inputs and returns one or multiple outputs. The design variables for the optimization can be of different natures, with the shared variables (denoted x_0) serving as inputs for several disciplines and the local variables (denoted x_i for the i^{th} discipline) being needed by only one particular discipline.

The coupling variables are of particular interest, as they represent the outputs of a specific discipline that are considered as inputs for at least one other discipline. In the most simple scenario, the coupling variables impose an order of execution for the different disciplines when only forward effects are present. In this case, the disciplines are said to be weakly coupled and the coupling structure can be represented as a directed acyclic graph as illustrated in Figure 2.1. Otherwise, the disciplines are said to be strongly coupled: some outputs of some discipline are inputs for other disciplines, and symmetrically, outputs

of the other disciplines are inputs for the first ones considered. Therefore the associated coupling graph, which summarize the coupling structure, contains cycles as highlighted in the example given in Figure 2.2.

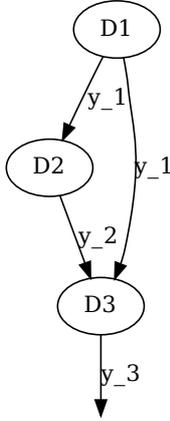


Figure 2.1: A weak (acyclic) coupling structure for 3 disciplines

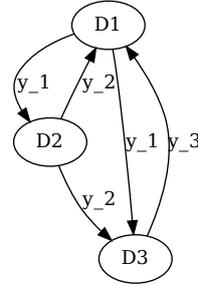


Figure 2.2: A strong coupling structure for 3 disciplines

As complex real-world MDO processes possess numerous disciplines, they typically exhibit both strongly and weakly coupled disciplines.

Moreover, each discipline has its own state equations, which are solved according to the supplied inputs and its own state variables. Executing a discipline indexed i at $(x_0, x_i, y_{\neq i})$, with the convention $y_{\neq i} = [y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_p]$, is then analogous to solve the implicit equation in (u_i, y_i) :

$$r_i(x_0, x_i, y_{\neq i}, y_i, u_i) = 0. \quad (2.1.1)$$

A representation of all variables and dependencies on a generic two disciplines case is given in Figure 2.3 and Figure 2.4, for weak and strong couplings respectively.

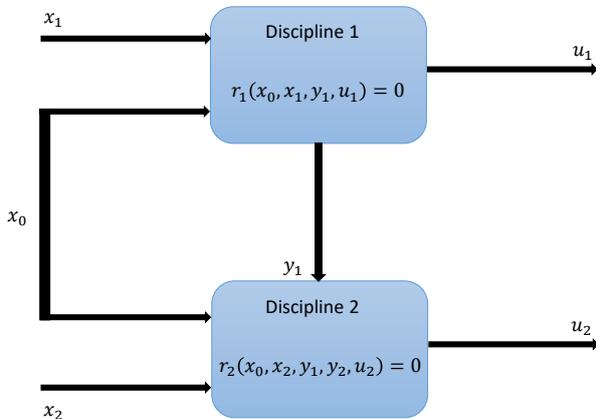


Figure 2.3: Example of a 2 weakly coupled disciplines system

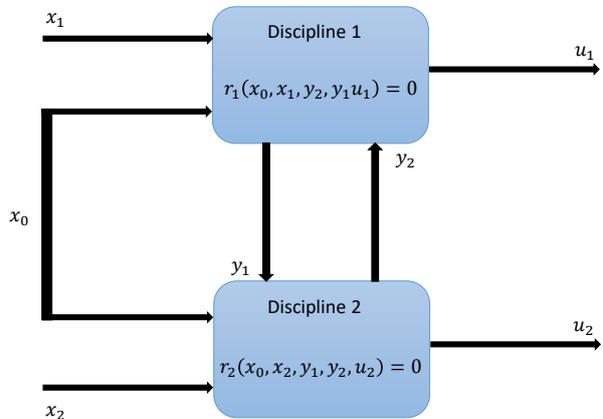


Figure 2.4: Example of a 2 strongly coupled disciplines system

One of the primary challenges in multidisciplinary design is to ensure that the coupling variables are consistent across all disciplines at some point in the optimization process, and especially at the converged solution. This equilibrium is achieved when, for a specific value of the design variables (x_0, x) , the vector y satisfies the coupling constraints:

$$\forall i \in \llbracket 1, p \rrbracket \quad y_i - \phi_i(x_0, x_i, y_{\neq i}) = 0. \quad (2.1.2)$$

In other words, y is at equilibrium if and only if

$$\begin{bmatrix} y_1 - \phi_1(x_0, x_1, y_{\neq 1}) \\ y_2 - \phi_2(x_0, x_2, y_{\neq 2}) \\ \vdots \\ y_p - \phi_p(x_0, x_p, y_{\neq p}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (2.1.3)$$

In this context, ϕ_i represents the so-called coupling function of the i^{th} discipline and is implicitly defined by the i^{th} residual equation (2.1.1). This function explicitly provides the value of y_i , which is related to the discipline's inputs.

In order to identify such a vector y , two strategies are commonly considered, depending on the structure of the problem. One approach is to create copies, also known as target values and generally designated as y^t , of the coupling variables and directly provide the consistency constraints given by Equation (2.1.2) to the optimizer. Conversely, a dedicated algorithm can be provided to solve the consistency constraints (2.1.2). These algorithms, referred as Multidisciplinary Analysis (MDA), are implementations of various algorithms, such as root-finding (Newton, fixed point), least-squares, or less frequently, penalization of the state equations. Typically, the MDAs are assumed to be solved using iterative algorithms that loop over disciplines in parallel, namely the non-linear Jacobi method [112], or sequentially, such as in the non-linear Gauss-Seidel method [112], both processes are illustrated in Figure 2.5.

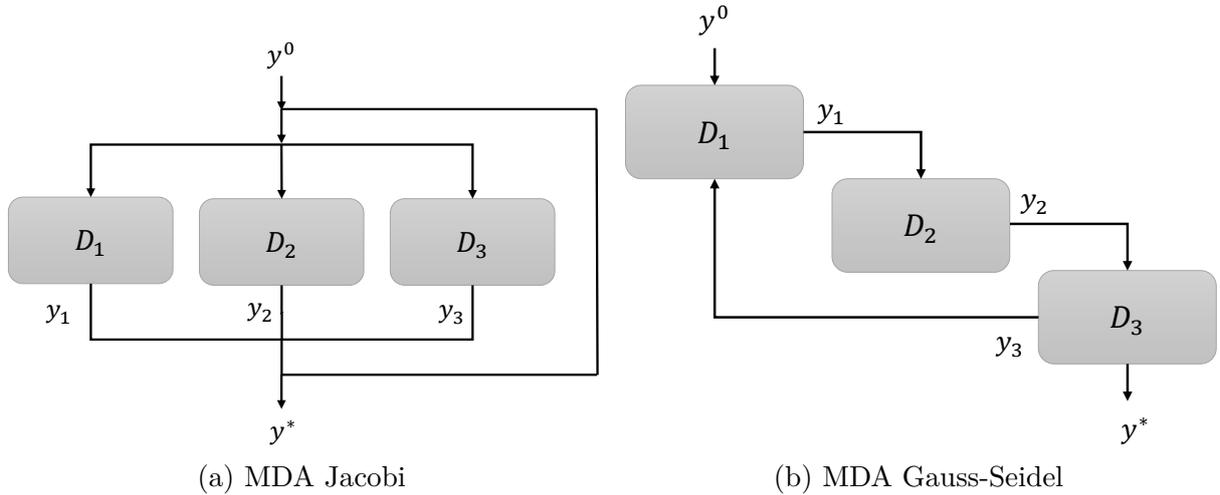


Figure 2.5: Multi-disciplinary Analysis (MDA) algorithms

2.1.2 MDO architectures

Given the multitude of approaches to solve the MDO problems and the potential for significant impact on performance, the MDO community has sought to define a taxonomy

of the strategies implemented under the influence of *Haftka and al* [67] and *Cramer and al* [32] in the 1990s. This taxonomy has been further investigated and classified in other surveys, including those by *Martins and al* [101], *Tosserams and al* [147], or *Balesdent and al.* [9]. Such strategies are referred to as MDO architectures or MDO formulations, equivalently [67, 32, 101, 9].

These are meta-strategies employed in the formulation of one or more optimization problems from a set of disciplines. A MDO architecture applied to a MDO problem is a combination of computational techniques, including fixed-point solvers, suboptimizations, and gradient computation techniques such as the coupled adjoint [139, 103]. These techniques are employed in conjunction to compute the optimization criteria of one or more optimization problems, namely the objective and constraints. The name of an architecture is frequently utilized to describe the associated meta-strategy, which describes a generic solution method for an abstract problem, as-well as to designate an instance of the strategy’s application to a specific problem, typically by giving a reformulation of the original optimization problems, as well as an associated workflow and dataflow driving the optimization.

As the choice of the architecture impacts all aspects of the optimization, this is of primary importance when facing a specific MDO problem. This is in particularly true as, like in regular optimization algorithms, the no-free-lunch theorem does hold [150], this means that the choice of the ”best” architecture, in terms of specified criteria such as the overall cost or the robustness, is highly dependent on the problem under consideration.

In a formal sense, a MDO architecture is a reformulation of an original optimization problem into a unique (monolithic architecture) or multiple optimization problems (distributed architectures), generally smaller, by decomposing, reallocating, or eliminating design variables, design constraints, or state equations. Consequently, these variables, constraints, and equations will no longer be handled by the optimizer. Furthermore, in order to ensure that the architecture is consistent with the original one, in the sense that they have the same set of solutions, a sequence of computations is attached to define neglected design variables and constraints as a numerical process and ensure global coherency of the design. In the context of defining new architectures, we will typically present the reformulated optimization problem and the associated workflow and dataflow in the form of an XDSM [83] diagram, if available, where the XDSM diagram representation is a standard visualization tool used to illustrate both the dataflow (in grey) and the workflow (in bold) of the manipulated MDO architectures.

Given that each choice of architecture entails a modification to the underlying design optimization problem and the overall computational process, it is possible to categorize these according to the nature of the strategies involved. [101] describes four different types of architecture depending on two criteria and summarized within Figure 2.6.

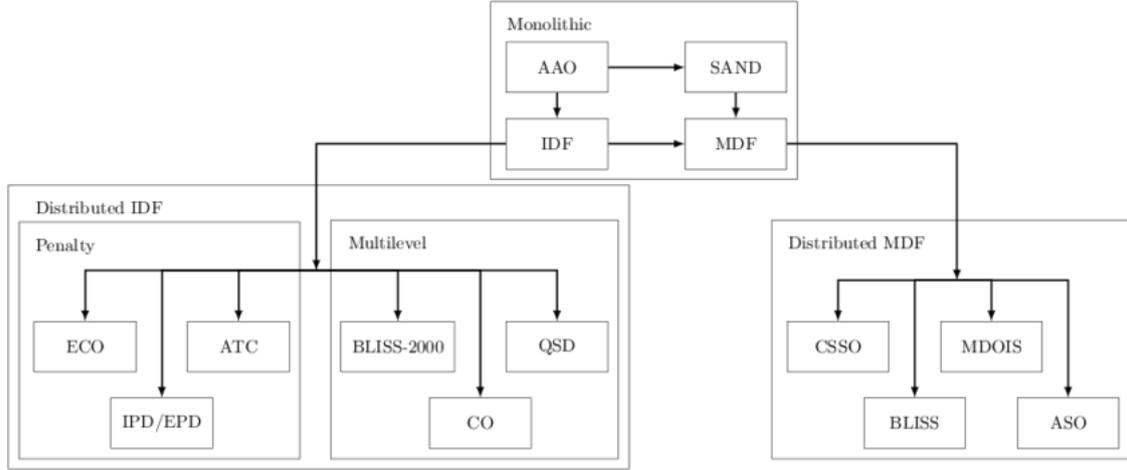


Figure 2.6: A classification of MDO architectures [101]

The first criterion is the governance of couplings. The architecture may rely on the use of a MDA to solve the disciplines' interactions (MDF-based architectures), or it may add target values and consistency constraints (IDF-based architectures). Moreover, architectures can be classified as either monolithic architectures or distributed architectures. As the former consists of fundamental MDO architectures that collectively encompass the most prevalent methodologies for addressing MDO challenges, a brief summary of the most important monolithic architectures is given hereafter. It should be noted that distributed architectures, which are derived from the monolithic ones, will be studied in Section 3.2.1.

Monolithic architectures

Upon initial encounter with a MDO problem, it is probable that the AAO (All-At-Once) [32, 101, 9] architecture will be encountered first. This architecture is considered the most generic, since it handles all variables, states variables, constraints, and residuals. This architecture is defined by the following optimization problem:

$$\begin{aligned}
 \min_{(x_0, x, y, y^t, u)} \quad & f(x_0, x, y) = f_0(x_0, x, y) + \sum_{k=1}^n f_i(x_0, x_i, y_i) \\
 \text{subject to} \quad & g_0(x_0, x, y) \leq 0 \\
 & g_i(x_0, x_i, y_i) \leq 0 \\
 & r_i(x_0, x_i, y_{j \neq i}^t, y_i, u_i) = 0 \\
 & y - y^t = 0
 \end{aligned} \tag{AAO}$$

The system objective and constraints, designated by f_0 and g_0 , respectively, depend on all design variables. In contrast, the disciplinary objective and constraints, represented by f_i and g_i ($i \in \llbracket 1, p \rrbracket$), depend on shared variables and on a particular subset of local variables and coupling variables. y^t are the so-called target values for the coupling variables, they are used as inputs for the disciplines to evaluate their respective residual equations. At convergence of the optimization, all residuals should be equals to zero and the design must be consistent, meaning that the target values vector y^t must match the coupling variables vector y . This is represented by the addition of a consistency constraint $y - y^t = 0$ in

addition of other non-linear constraints. In general, bounds constraints, which for the sake of simplicity will not be shown, are also present for every design variables to restrict the scope of research, which is typically too large for general MDO problems.

(*AAO*) is primarily utilized for theoretical purposes, as it ultimately represents the problem that is to be solved. In practice, (*AAO*) is the largest of all MDO architectures, in the sense that all possible functions and design variables are given to the optimizer, and is not intended to be solved directly. The removal of target values y^t and consistency constraints $y - y^t = 0$, results in the SAND (Simultaneous Analysis and Design) architecture:

$$\begin{aligned} & \min_{(x_0, x, y, u)} && f(x_0, x, y) \\ \text{subject to} &&& g_0(x_0, x, y) \leq 0 \\ &&& g_i(x_0, x_i, y_i) \leq 0 \\ &&& r_i(x_0, x_i, y_{\neq i}, y_i, u_i) = 0 \end{aligned} \tag{SAND}$$

Clearly, by simply injecting $y^t = y$, the (*SAND*) and (*AAO*) architectures are equivalent, in the sense that they both possess the same set of optimal solutions. The coherency of the couplings is enforced at convergence by zeroing all disciplinary residuals. Because of the simplicity of this transformation, the MDO literature often treats both SAND and AAO designations indifferently when referring to (*SAND*) [32, 101]. (*SAND*) is a basic approach that can be used in classical mono-disciplinary optimization where no exact disciplinary analysis are performed.

Among the known limitations, (*SAND*) is potentially a large-scale optimization problem, as the optimizer must address both state variables u_i and residuals r_i for each discipline. Consequently, for high-dimensional problems, especially those with numerous disciplines and coupling variables, it is possible that convergence may be hindered and/or that multidisciplinary feasibility may be difficult to achieve when the optimization process converges. Another limitation is of a practical nature: often, especially in industry and especially when using COTS software, disciplines are considered in a black-box fashion, meaning that residuals and state variables are not directly accessible without intrusive code modifications.

This last observation motivates the use of architectures that do not rely on the availability of state variables and residuals. Two such architectures are the well-known Individual Discipline Feasible (IDF) and Multi-Disciplinary Feasible (MDF) architectures [32, 101]. Both architectures rely on the implicit function theorem to rewrite state variables as (at least) continuously differentiable functions. In particular, each discipline residual equation allows the definition of the so-called disciplinary analysis $y_i = \phi_i(x_0, x_i, y_{\neq i})$.

Individual Discipline Feasible (IDF) The Individual Discipline Feasible (IDF) architecture [32, 101] is one of the two classical monolithic architectures in MDO. Each coupling variable y_i is considered to be implicitly defined by the shared design vector x_0 , the associated local variables x_i , and other couplings $y_{\neq i}$ through the coupling function ϕ_i . To ensure the coherence of the design at the optimum, copies, also called target values, of the coupling variables are created, denoted as y^t . Equality with the output of the disciplinary analysis is then enforced by consistency constraints. The resulting IDF optimization problem is as follows:

$$\begin{aligned}
& \min_{(x_0, x, y^t)} && f(x_0, x, y^t) \\
& \text{subject to} && g_0(x_0, x, y^t) \leq 0 \\
& && g_i(x_0, x_i, \phi_i(x_0, x_i, y_{\neq i}^t)) \leq 0 \quad \forall i = 1 \dots p \\
& && y_i^t - \phi_i(x_0, x_i, y_{\neq i}^t) = 0 \quad \forall i = 1 \dots p
\end{aligned} \tag{IDF}$$

A simple illustration of the computational process of the IDF architecture, which involves two strongly coupled disciplines, is shown in Figure 2.7.

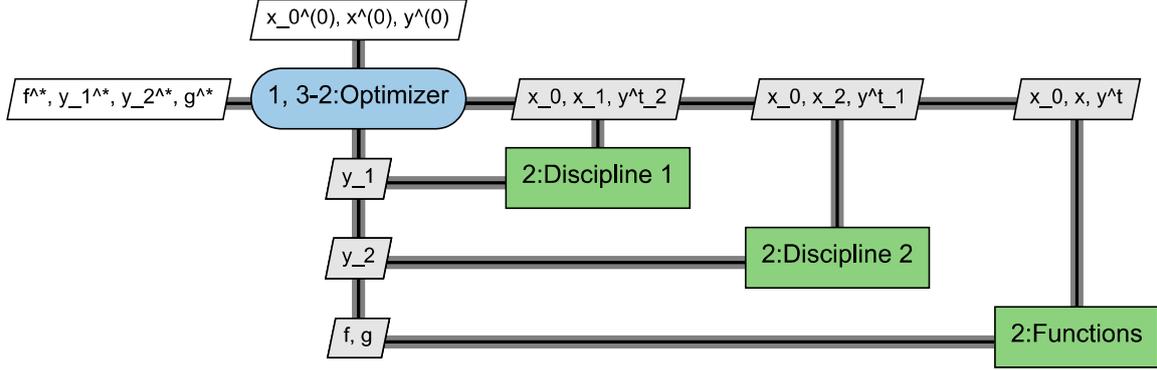


Figure 2.7: XDSM of the IDF architecture

In addition to not relying on the availability of residuals, the introduction of copies of the couplings allows for parallel computation of the disciplinary analysis, resulting in relatively fast system function evaluations. However, this approach does not ensure multidisciplinary feasibility at each iteration of the optimization process. In addition, it generally requires more system iterations to achieve a fully consistent design. When all derivatives are available, gradient-based algorithms can be very efficient when considering IDF, especially in the case of low-dimensional couplings.

However, as the complexity of the system problem increases with the number of coupled variables, high-dimensional couplings, and non-linear constraints, may result in memory and time-consuming Jacobian computations. Furthermore, the optimizer may encounter difficulties in retrieving multidisciplinary feasibility at the end of the optimization process, leading to an inconsistent design in the event of early termination.

Multi-Disciplinary Feasible (MDF) The other classical monolithic architecture is the so-called Multi-Disciplinary Feasible (MDF) architecture [32, 101]. Deriving from the IDF architecture, the MDF architecture incorporates an additional layer of implicit function theorem to eliminate consistency constraints. Remind that for a specific value of the design variable (x_0, x) , the coupling vector y is said to satisfy the coupling constraints if and only if it is solution of Equation (2.1.3).

Assuming that the implicit function theorem holds (cf. Theorem 2.2.5), there exists a unique continuously differentiable function, denoted Ψ , such that $y = \Psi(x_0, x)$ satisfies the coupling constraints (2.1.2). In this context, the MDF optimization problem can be expressed as follows:

$$\begin{aligned}
& \min_{(x_0, x)} && f(x_0, x, \Psi(x_0, x)) \\
& \text{subject to} && g_0(x_0, x, \Psi(x_0, x)) \leq 0 \\
& && g_i(x_0, x_i, \Psi_i(x_0, x)) \leq 0 \quad \forall i = 1 \dots p
\end{aligned}
\tag{MDF}$$

A straightforward illustration of the MDF architecture computational process, utilizing a Gauss-Seidel MDA and encompassing two strongly coupled disciplines, is presented in Figure 2.8.

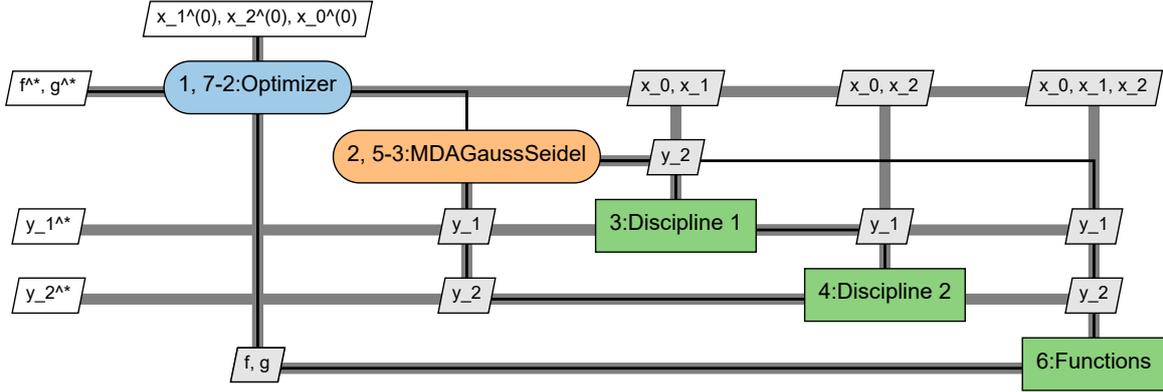


Figure 2.8: XDSM of the MDF architecture using a Gauss-Seidel MDA

In contrast to the approach taken by the IDF architecture, the coupling vector y is required to satisfy consistency constraints (2.1.2) at each stage of the optimization process. The primary limitations of the MDF architecture pertain to the practical and theoretical utilization of a MDA. It is assumed that each design vector yields a unique feasible coupling vector, $y = \Psi(x_0, x)$, and that the selected algorithm is capable of successfully attaining the solution. From experience, it can be observed that the degeneracy of the MDA is rarely encountered in practice. This is because modern solvers are designed to be as robust as possible and, in the worst cases, to return an approximate solution instead of a complete failure. Otherwise, the problem may be considered ill-posed, which increases the risk of failing to compute a multidisciplinary feasible optimum, even for non-MDA-based architectures. Furthermore, in the MDF approach, a complete MDA must be carried out at each iteration. Consequently, the disciplines that are potentially time-consuming will be executed multiple times per iteration of the system problem, resulting in, on average, significantly more costly iterations compared to IDF. An overall cost further increased by the necessity to compute coupled derivatives for the application of gradient-based optimization algorithms to MDF. Several areas for improvement already exist, such as the use of acceleration techniques for fixed point solvers [127] or solving both the state equations and the MDA simultaneously [108].

In contrast, the MDF architecture exhibits several advantageous properties. Primarily, the system problem is the smallest possible for a monolithic architecture, as the coupled variables and the consistency constraints are completely handled by the MDA, thus eliminating the need for the optimizer to consider them explicitly. Consequently, there are fewer system optimizer calls than in IDF. Finally, although it cannot be guaranteed that infeasible points will not arise, MDF has the advantage of always returning a fully consistent system design, since the coupling variables y are enforced to be solution of

Equation (2.1.3) at each optimization iteration, which is highly valuable in the event of early termination of the optimization process.

2.2 Mathematical background

This section is dedicated to the various tools and mathematical results used throughout this manuscript. The first part is dedicated to the Gauss-Seidel algorithm and its convergence properties that are treated first in the classical linear case and then extended to the nonlinear case. Then two classical implicit analytic methods for computing derivatives are recalled, in particular the so-called adjoint vector is defined. Classical optimality conditions for general nonlinear programming are given, covering the Lagrangian reformulation of the problem, the first- and second-order Karush-Kuhn-Tucker (KKT) conditions, and the considered constraint qualifications. Finally, 3 classical analysis theorems that allow to define continuously differentiable functions from a set of hypotheses are presented.

2.2.1 The Gauss-Seidel algorithm and local convergence

The Gauss-Seidel method [63] is an iterative algorithm originally designed to solve a system of linear equations. This method, which is part of a broader family of iterative algorithms called the Successive Over-Relaxation (SOR) methods [63], has been extensively studied in the linear case and extended to more general problems, such as nonlinear systems of equations [113, 111, 110, 112]. These methods and their convergence properties are of paramount importance in this manuscript, both for the solution of the multidisciplinary analysis (cf. Section 2.1.1), and for an important local convergence result for a closely related algorithm within Section 3.4.2.

For these reasons, this section is dedicated to classical convergence results that will be helpful for the understanding of the manipulated concepts and classical hypotheses that would be useful along this manuscript.

The linear case

Let $A \in \mathcal{GL}_p(\mathbb{R})$ be non-singular, let $b \in \mathbb{R}^p$. Our goal is to find $x \in \mathbb{R}^p$ that solves the classical linear algebra problem:

$$Ax = b. \tag{2.2.1}$$

To solve (2.2.1), it is common to rely on iterative algorithms of the form

$$x^{k+1} = Hx^k + b, \tag{2.2.2}$$

where H depends on the method used and is often referred to as the iteration matrix. A standard convergence condition for these algorithms is related to the spectrum and the spectral radius of the iteration matrix H . Often, the properties on H can be directly deduced from the matrix A when available. An important property lies in A being positive-definite.

Definition 2.2.1 (Spectrum). Let $A \in \mathcal{M}_n(\mathbb{R})$. We call spectrum of A , and note $Sp(A)$, the set of eigenvalues related to A :

$$Sp(A) \equiv \{\lambda \in \mathbb{R} \mid \exists x \neq 0, Ax = \lambda x\}. \tag{2.2.3}$$

Definition 2.2.2 (Spectral radius). Let $A \in \mathcal{M}_n(\mathbb{R})$. We call spectral radius, and note $\rho(A)$, the radius of the smallest ball containing all elements of $Sp(A)$. Equivalently:

$$\rho(A) \equiv \max_i \{|\lambda_i| \mid \lambda_i \in Sp(A)\}. \quad (2.2.4)$$

Definition 2.2.3 (Positive-definite). Let $A \in \mathcal{M}_n(\mathbb{R})$ be symmetric. We say that A is positive-definite, and note $A > 0$, if and only if:

$$\forall x \in \mathbb{R}^n \setminus \{0\}, \quad x^T A x > 0. \quad (2.2.5)$$

Equivalently:

$$A > 0 \iff \lambda > 0, \quad \forall \lambda \in Sp(A). \quad (2.2.6)$$

Theorem 2.2.1 (Convergent matrix [151]). Assume that $\rho(H) < 1$, then H is said to be a convergent matrix, in the sense that $\lim_{k \rightarrow \infty} H_{i,j}^k = 0$ for all $(i, j) \in \llbracket 1, p \rrbracket^2$.

As a consequence, the iterative procedure given in (2.2.2) is convergent towards the solution of (2.2.1) for every initial vector, i.e. $\lim_{k \rightarrow \infty} x^k = A^{-1}b$.

To find a suitable iteration matrix H , it is common to rely on a matrix splitting of the matrix A , i.e. a rewriting of A as a sum of matrices. The most used matrix splitting of A is defined as

$$A = M + D + U, \quad (2.2.7)$$

where M is the strictly lower triangle of A , D is the diagonal of A , and U is the strictly upper triangle of A .

One of the most popular methods to solve (2.2.1) is the Jacobi method [63], where each diagonal term is solved in parallel. Assume that D is non-singular, by using previous splitting (2.2.7), (2.2.1) can be rewritten as

$$Ax = b \quad (2.2.8)$$

$$\iff (M + D + U)x = b \quad (2.2.9)$$

$$\iff Dx = b - (M + U)x \quad (2.2.10)$$

$$\iff x = D^{-1}(b - (M + U)x), \quad (2.2.11)$$

from which we can deduced the well-known Jacobi updates:

$$x^{k+1} = D^{-1}(b - [M + U]x^k) = D^{-1}b - D^{-1}[M + U]x^k. \quad (\mathbf{Jacobi})$$

It is well known that a sufficient condition for the said iteration method to be convergent is that $\rho(D^{-1}[M + U]) < 1$ according to Theorem 2.2.1. This is ensured, for example, if the matrix A is strictly or irreducibly diagonally dominant [63]. These classical convergence results may be too restrictive in general and, when convergence is not certain, other methods may be preferred.

One way to strengthen these convergence results is to recycle the information more frequently. For example, the Gauss-Seidel method solves each equation sequentially,

from the first equation to the last, with each solution being injected into the subsequent equations to speed up convergence. Let $L = M + D$ and assume that L is non-singular. (2.2.1) can be rewritten as

$$Ax = b \tag{2.2.12}$$

$$\iff (L + U)x = b \tag{2.2.13}$$

$$\iff Lx = b - Ux \tag{2.2.14}$$

$$\iff x = L^{-1}(b - Ux). \tag{2.2.15}$$

Thus, two successive iterations are linked through the relation

$$Lx^{k+1} + Ux^k = b, \tag{2.2.16}$$

which leads to the well-known Gauss-Seidel iteration scheme

$$x^{k+1} = -L^{-1}Ux^k + L^{-1}b. \tag{Gauss-Seidel}$$

Theorem 2.2.2 (Convergence of the Gauss-Seidel method). *Let the following linear system $Ax = b$ as described in (2.2.1). Consider the splitting given in (2.2.7), $A = L + U$ and assume that L is non-singular. Then the Gauss-Seidel iteration sequence (Gauss-Seidel) given by:*

$$x^{k+1} = -L^{-1}Ux^k + L^{-1}b,$$

converges to the solution of $Ax = b$ if and only if $\rho(L^{-1}U) < 1$.

Proof: Let x^p be the p^{th} iterate obtained by using (Gauss-Seidel) with $p \in \mathbb{N}$. The next iterate x^{p+1} is then defined as

$$x^{p+1} = -L^{-1}Ux^p + L^{-1}b. \tag{2.2.17}$$

Developing x^p gives the following expression:

$$x^{p+1} = -L^{-1}U[-L^{-1}Ux^{p-1} + L^{-1}b] + L^{-1}b. \tag{2.2.18}$$

Repeating this process, an immediate induction gives the following formula for any $p \in \mathbb{N}$:

$$x^{p+1} = (-L^{-1}U)^p x^0 + \left(\sum_{k=0}^p (-L^{-1}U)^k \right) L^{-1}b, \tag{2.2.19}$$

where x^0 is the initial guess.

This iteration scheme converges if and only if $\lim_{p \rightarrow \infty} \|(-L^{-1}U)^p\|$ is finite for any induced norm $\|\cdot\|$ and that the series $\sum (-L^{-1}U)^p$ is convergent which is true if and only if $\rho(L^{-1}U) < 1$.

Assume now that $\rho(L^{-1}U) < 1$. Let us show that the Gauss-Seidel iteration scheme converges to the solution of $Ax = b$.

In this case, it follows that $\lim_{p \rightarrow \infty} (-L^{-1}U)^p = 0$ and that $\sum (-L^{-1}U)^p = (I_n + L^{-1}U)^{-1}$. Therefore the iteration scheme converges to

$$x^* = (I_n + L^{-1}U)^{-1}L^{-1}b. \tag{2.2.20}$$

Multiplying each side by $A = L + U$, it leads to

$$\begin{aligned} Ax^* &= (L + U)(I_n + L^{-1}U)^{-1}L^{-1}b \\ &= L(I_n + L^{-1}U)(I_n + L^{-1}U)^{-1}L^{-1}b. \\ &= b \end{aligned} \tag{2.2.21}$$

Finally the iteration scheme converges if and only if $\rho(L^{-1}U) < 1$, and it converges to x^* the solution of the system $Ax = b$. \square

Since the Gauss-Seidel method uses all available information at each step, it is indeed faster, in the sense that it takes fewer iterations to converge, and more robust than the Jacobi method. This is due to the inequality $\rho(L^{-1}U) \leq \rho(D^{-1}[M + U])$ being strict in the vast majority of events, equality happening for some limit cases such as A being diagonal [63]. The Gauss-Seidel method belongs to the more general category of the Successive Over-Relaxation (SOR) methods. Let $\omega > 1$, the generic SOR iteration rewrites (2.2.1) as

$$(D + \omega M)x = \omega b - [\omega U + (\omega - 1)D]x, \tag{2.2.22}$$

which is solved using the following updates:

$$x^{k+1} = (D + \omega M)^{-1}[\omega b - [\omega U + (\omega - 1)D]x^k]. \tag{2.2.23}$$

It has been proved that the SOR method is convergent for every $0 < \omega < 2$ whenever A is symmetric positive-definite by showing that we necessarily have $\rho(-(D + \omega M)^{-1}[\omega U + (\omega - 1)D]) < 1$ [63].

In particular, this is also true for the Gauss-Seidel algorithm, obtained with the special case $\omega = 1$. Therefore the choice of ω has no effect on the convergence properties compared to the classical Gauss-Seidel method [63]. However, the introduction of the parameter ω is often considered to speed-up the convergence. Unfortunately, it is not straightforward to find the optimal relaxation parameter that guarantees the best convergence speed, since it requires knowledge of the spectral radius of the Jacobi iteration matrix $D^{-1}[M + U] = I - D^{-1}A$ [63].

The Gauss-Seidel method is easy to understand and implement, so it is widely used in practice for a variety of applications. For MDO problems, since most of the functions encountered in industry are nonlinear, the nonlinear Gauss-Seidel method is generally considered instead. Classical results for the nonlinear Gauss-Seidel approach are presented below in the context of minimizing a nonlinear function.

The non-linear case as a local solution of an unconstrained problem

Let $f : \prod_{i=1}^p \mathbb{R}^{n_i} \mapsto \mathbb{R}$ be a nonlinear continuously differentiable function with p input vectors $x_i \in \mathbb{R}^{n_i}$ to be minimized. It is well-known that local minimizers of f are characterized by zeroing its gradient. Therefore all locally optimal points $x \in \prod_{i=1}^p \mathbb{R}^{n_i}$ are characterized by the following condition:

$$\nabla f(x_1, \dots, x_p) = 0. \tag{2.2.24}$$

This is a nonlinear system of equations, that we want to solve by using the nonlinear Gauss-Seidel approach [113, 111, 110]. The method is also known in the literature as the

Block Coordinate Descent (**BCD**) algorithm [16], the Block Gauss-Seidel (**BGS**) method [64] or the Alternating Optimization (**AO**) method [17, 18]. For the minimization of an unconstrained function, or equivalently for the solution of a nonlinear system of equations, our preference goes to the denomination nonlinear Gauss-Seidel, as it translates that it is a direct application of the linear algorithm in the nonlinear case. For the constrained case, as encountered in Chapter 3, the term Block Coordinate Descent (BCD) is used instead to emphasize the consideration of constraints.

The nonlinear Gauss-Seidel method is derived directly from the original linear approach. Let x^k , with $k \in \mathbb{N}$, be the actual iterate, x^{k+1} is obtained by successively defining x_i^{k+1} , from $i = 1$ to $i = p$, as the solution of

$$\frac{\partial f}{\partial x_i}(x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \dots, x_p^k) = 0. \quad (2.2.25)$$

The following theorem, given by Ortega [110], expresses sufficient conditions for the nonlinear Gauss-Seidel method to be locally convergent:

Theorem 2.2.3 (Local convergence of non-linear Gauss-Seidel). *Let x^* be a local minimizer of f . Let N be a neighborhood of x^* where f is at least C^2 , strictly convex and $\nabla^2 f(x^*)$ is positive-definite. Assume that every block have a unique minimizer.*

Then the non-linear Gauss-Seidel method is locally convergent towards x^ q -linearly, i.e. there exist $q \in [0, 1[$ and $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, $\|x^{k+1} - x^*\| \leq q \|x^k - x^*\|$.*

Proof: Let x^k be the current iterate. First observe that defining x_i^{k+1} as the solution of (2.2.25) for all $i \in \llbracket 1, p \rrbracket$ is equivalent as finding a vector u such that for an already known x^k

$$g(u, x^k) = 0, \quad (2.2.26)$$

where $g : \prod_{i=1}^p \mathbb{R}^{n_i} \times \prod_{i=1}^p \mathbb{R}^{n_i} \rightarrow \prod_{i=1}^p \mathbb{R}^{n_i}$, the gradient of f where components of v are replaced by the components of u at each partial derivative computation, similar to the Gauss-Seidel pattern, is defined as

$$g(u, v) \equiv \begin{bmatrix} \frac{\partial f}{\partial x_1}(u_1, v_2, \dots, v_p) \\ \frac{\partial f}{\partial x_2}(u_1, u_2, v_3, \dots, v_p) \\ \vdots \\ \frac{\partial f}{\partial x_{p-1}}(u_1, \dots, u_{p-1}, v_p) \\ \frac{\partial f}{\partial x_p}(u_1, \dots, u_p) \end{bmatrix}. \quad (2.2.27)$$

As f is at least C^2 , g is at least C^1 with respect to both u and v . Furthermore, x^* being a local minimizer of f , it follows that $g(x^*, x^*) = \nabla f(x^*) = 0$.

After a direct computation at the optimum x^* , we observe that $\frac{\partial g}{\partial u}(x^*, x^*)$ is a $p \times p$ lower triangular matrix filled with the second derivatives of f at x^* . Similarly, $\frac{\partial g}{\partial v}(x^*, x^*)$ is a $p \times p$ strictly upper triangular matrix composed with the upper-triangular part of the Hessian of f at x^* .

Let $H(x^*)$ be the Hessian of f at x^* , it directly follows that

$$H(x^*) = \frac{\partial g}{\partial u}(x^*, x^*) + \frac{\partial g}{\partial v}(x^*, x^*). \quad (2.2.28)$$

Therefore, the Hessian of f at x^* can be decomposed as:

$$H(x^*) = L(x^*) - U(x^*), \quad (2.2.29)$$

with the direct identifications $L(x^*) = \frac{\partial g}{\partial u}(x^*, x^*)$ and $U(x^*) = -\frac{\partial g}{\partial v}(x^*, x^*)$.

Since every block has a unique minimizer, every diagonal block of $L(x^*)$ is positive definite. Hence $L(x^*) = \frac{\partial g}{\partial u}(x^*, x^*)$ is invertible, therefore the **Implicit Function Theorem** implies the existence of a unique continuous differentiable function T such that for every point x^k near x^* , we have

$$g(x^{k+1}, x^k) = 0 \iff x^{k+1} = T(x^k). \quad (2.2.30)$$

Note that, in particular, at the solution x^*

$$g(T(x^*), x^*) = g(x^*, x^*) = 0. \quad (2.2.31)$$

The Taylor expansion at x^k near the solution x^* gives

$$T(x^k) - x^* = \nabla T(x^*)(x^k - x^*) + r(x^k - x^*). \quad (2.2.32)$$

Hence:

$$x^{k+1} - x^* = \nabla T(x^*)(x^k - x^*) + r(x^k - x^*), \quad (2.2.33)$$

with $\frac{\|r(x^k - x^*)\|}{\|x^k - x^*\|} \rightarrow 0$, when $\|x\| \rightarrow 0$

It follows, according to Theorem 2.2.1, that the sequence $\{x^k\}_{k \in \mathbb{N}}$ converge to x^* if and only if $\rho(\nabla T(x^*)) < 1$.

To compute the spectral radius of $\nabla T(x^*)$, we first compute the total derivative of g :

$$\frac{dg}{dv} = \frac{\partial g}{\partial v} + \frac{\partial g}{\partial u} \frac{dT}{dv} = \frac{\partial g}{\partial v} + \frac{\partial g}{\partial u} \nabla T. \quad (2.2.34)$$

Hence at x^* , as $\frac{dg}{dv}(x^*) = 0$, it follows:

$$\nabla T(x^*) = \left[\frac{\partial g}{\partial T}(x^*) \right]^{-1} \frac{\partial g}{\partial v}(x^*). \quad (2.2.35)$$

According to previous notations

$$\nabla T(x^*) = [L(x^*)]^{-1} U(x^*). \quad (2.2.36)$$

It should be noted that, near the solution x^* , for a particular iterate x^k and its successor x^{k+1} , previous equation indicates that g can be rewritten as:

$$g(x^{k+1}, x^k) = L(x^*)x^{k+1} - U(x^*)x^k = 0, \quad (2.2.37)$$

in which we recognize the form of the linear (**Gauss-Seidel**) as expected.

Hence, to finish the proof, we need to show that $\rho([L(x^*)]^{-1}U(x^*)) < 1$.

From now on, the end of the demonstration relies on a corollary of Householder John's theorem (see [110, Corollary 2.11]), the statement of which is given below:

Lemma 2.2.1 (Householder–John corollary). Let H be a non-singular matrix. Split H into: $H = D - L - U$ where D , $-L$ and $-U$ respectively denote the diagonal, the strictly lower and the strictly upper triangular parts of H . Assume that H is Hermitian and that its block diagonal matrix D is positive definite. Then $\rho((D - L)^{-1}U) < 1$ if and only if H is positive definite.

By assumption $H(x^*) = L(x^*) - U(x^*)$ is hermitian and non-singular. Moreover, we already shown that $L(x^*)$ is also non-singular as a lower-triangular matrix in which all diagonal blocks admits a unique minimizer, i.e. the block diagonal matrix of $H(x^*)$ is definite positive.

Therefore, as $H(x^*)$ is definite positive by assumption, by [Householder–John corollary](#), $\rho([L(x^*)]^{-1}U(x^*)) < 1$, so [Theorem 2.2.1](#) applies which completes the proof on the convergence of the method towards x^* . \square

This well-known and classical result, as well as the associated proof, are of primary importance for understanding the local convergence result given in [Chapter 3](#), where the nonlinear equations are replaced by nonlinear, constrained and strongly coupled optimization problems. In particular, [Lemma 2.2.1](#) will no longer apply due to the presence of constraints that break the positive definiteness of the block diagonal matrix D .

2.2.2 Direct and adjoint methods for gradient computation

Let $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \mapsto \mathbb{R}^{n_f}$ be a continuously differentiable function computing some quantity of interest $f(x, y)$, such as objectives or constraints, for some inputs x and y . Let us recall the Implicit Function Theorem:

Theorem 2.2.4 (Implicit Function Theorem). *Let $R : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$ be a continuously differentiable function, and let \mathbb{R}^{n+m} have coordinates (x, y) .*

Fix a point (a, b) such that $R(a, b) = 0_{\mathbb{R}^m}$. If the Jacobian matrix $\left[\frac{\partial R_i}{\partial y_j}(a, b) \right]_{(i,j) \in \llbracket 1, m \rrbracket^2}$ is invertible, then there exists an open set $\mathcal{U} \subset \mathbb{R}^n$ and a unique continuously differentiable function $\Psi : \mathcal{U} \rightarrow \mathbb{R}^m$ such that $\Psi(a) = b$ and $R(x, \Psi(x)) = 0$ for all $x \in \mathcal{U}$.

Suppose there exists a point (\bar{x}, \bar{y}) where a continuously differentiable residual equation $R(\bar{x}, \bar{y}) = 0$ is satisfied and the Jacobian of R with respect to the coordinate y is invertible at (\bar{x}, \bar{y}) . Thus, by [Implicit Function Theorem](#), there exists a neighborhood of \bar{x} and a unique continuously differentiable function Ψ such that for every x in this neighborhood

$$R(x, y) = 0 \iff y = \Psi(x). \quad (2.2.38)$$

This section explores two classical implicit analytic methods related to the computation of the gradient of f with respect to x in the considered neighborhood, by exploiting the implicit relation described above.

Two different approaches

We restrict ourselves to the neighbourhood of \bar{x} in which the relation [\(2.2.38\)](#) is verified. As [Equation \(2.2.38\)](#) is always verified for every selected x , then the state equation's

derivatives along x is zero:

$$\frac{dR}{dx}(x, \Psi(x)) = \frac{\partial R}{\partial x}(x, \Psi(x)) + \frac{\partial R}{\partial y}(x, \Psi(x)) \frac{d\Psi}{dx}(x) = 0, \quad (2.2.39)$$

from which we can use the invertibility $\frac{\partial R}{\partial y}$ to express the total derivative of the implicitly defined function Ψ with respect to x :

$$\frac{d\Psi}{dx}(x) = - \left[\frac{\partial R}{\partial y}(x, \Psi(x)) \right]^{-1} \frac{\partial R}{\partial x}(x, \Psi(x)). \quad (2.2.40)$$

By injecting this expression in f 's total derivative, it follows that

$$\frac{df}{dx}(x, \Psi(x)) = \frac{\partial f}{\partial x}(x, \Psi(x)) + \frac{\partial f}{\partial y}(x, \Psi(x)) \frac{d\Psi}{dx}(x) \quad (2.2.41)$$

$$= \frac{\partial f}{\partial x}(x, \Psi(x)) - \frac{\partial f}{\partial y}(x, \Psi(x)) \left[\frac{\partial R}{\partial y}(x, \Psi(x)) \right]^{-1} \frac{\partial R}{\partial x}(x, \Psi(x)), \quad (2.2.42)$$

where all terms are partial derivatives. The computation of the total derivative given in Equation (2.2.42) can be done in two distinct manners described below.

- The first approach, often referred as the direct method, consists in computing the total derivative of the implicit function with respect to x :

$$\alpha(x) \equiv \left[\frac{\partial R}{\partial y}(x, \Psi(x)) \right]^{-1} \frac{\partial R}{\partial x}(x, \Psi(x)) = - \frac{d\Psi}{dx}(x), \quad (2.2.43)$$

by solving first the related linear system:

$$\frac{\partial R}{\partial y}(x, \Psi(x)) \alpha(x) = \frac{\partial R}{\partial x}(x, \Psi(x)). \quad (2.2.44)$$

The obtained result can then be directly injected onto Equation (2.2.42):

$$\frac{df}{dx}(x, \Psi(x)) = \frac{\partial f}{\partial x}(x, \Psi(x)) - \frac{\partial f}{\partial y}(x, \Psi(x)) \alpha(x). \quad (2.2.45)$$

- The adjoint approach (also known as the reverse method), contrary to the former strategy, first compute the adjoint vector β :

$$\beta^T(x) \equiv \frac{\partial f}{\partial y}(x, \Psi(x)) \left[\frac{\partial R}{\partial y}(x, \Psi(x)) \right]^{-1}, \quad (2.2.46)$$

which is obtained as the solution of the related linear system

$$\frac{\partial R^T}{\partial y}(x, \Psi(x)) \beta(x) = \frac{\partial f^T}{\partial y}(x, \Psi(x)). \quad (2.2.47)$$

The adjoint vector $\beta^T(x)$ can be interpreted as the effect of a perturbation of the residual R on f , which can be written, with a notation abuse, as $\frac{df}{dR}(x\Psi(x))$. Injecting the adjoint vector β into Equation (2.2.42) allows to write

$$\frac{df}{dx}(x, \Psi(x)) = \frac{\partial f}{\partial x}(x, \Psi(x)) - \beta^T(x) \frac{\partial R}{\partial x}(x, \Psi(x)). \quad (2.2.48)$$

Choice of the method

Both approaches seem similar at first hand, both strategies imply the computation of the partial derivative of f , the resolution of a linear system and a matrix multiplication and both methods have the advantage of approximating the total derivative with a precision matching that of the considered functions.

However the cost of solving the linear system changes drastically depending on the chosen method. As the direct method involves the partial derivatives of the residual with respect to x : $\frac{\partial R}{\partial x}$, the cost of solving Equation (2.2.44) scales with the number of design variables n_x . On the other hand, Equation (2.2.47) depends on the partial derivatives of the objective with respect to the implicitly defined set of variables: $\frac{\partial f}{\partial y}$, and therefore scales with the number of functions (or the size of f) n_f .

This indicates that the choice of using one method or the other depends above all on dimensions of the design variables n_x and functions n_f . In the event where the number of functions is higher than the number of variables (i.e. $n_f > n_x$), it is best to use the direct approach summarized in Equation (2.2.45). Otherwise (i.e. $n_x > n_f$), the adjoint approach (cf. Equation (2.2.48)) should be considered instead. This is mainly why the adjoint method is generally preferred, as it is common for many applications to have much more design variables than functions.

However, they also require implementation efforts, often intrusive ones, since they require the derivatives of the residuals with respect to the state variables y . This explains in part why the use of the adjoint method in this industry is not necessarily systematic for complex problems with many disciplines [59], despite the fact that for high-fidelity optimization, gradient-based approaches with the coupled adjoint offer the best scalability [103].

2.2.3 Optimality conditions

In this manuscript, most considered optimization problems are nonlinear and subject to constraints. This section is dedicated to classical results on nonlinear programming on sufficient and necessary conditions for a design point to be locally optimal. Given a generic optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i \in \{1, \dots, m\}, \\ & h_j(x) = 0, \quad j \in \{1, \dots, r\} \end{aligned} \tag{2.2.49}$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ denotes the inequality constraints and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, r$ are the equality constraints.

It is possible to define its Lagrangian,

$$\mathcal{L}(x, u, v) = f(x) + \sum_{i=1}^m u_i g_i(x) + \sum_{j=1}^r v_j h_j(x), \tag{2.2.50}$$

with $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^r$ be the so-called Lagrange multipliers. Solving the original problem (2.2.49) is related to the search for saddle points of \mathcal{L} where each components of

u are positive, or in other words, to the solutions of the following primal problem

$$\min_x \left[\max_{u,v:u_i>0;\forall i} \mathcal{L}(x,u,v) \right], \quad (2.2.51)$$

and dual problem:

$$\max_{u,v:u_i>0;\forall i} \left[\min_x \mathcal{L}(x,u,v) \right]. \quad (2.2.52)$$

It should be noted that for every fixed x , the function $(u,v) \mapsto \mathcal{L}(x,u,v)$ is an affine function of u and v .

Assuming that all the considered functions are at least C^1 functions, optimality conditions for general non-linear optimization problems are given by the well-known first order Karush–Kuhn–Tucker (KKT) conditions:

Definition 2.2.4 (1st order KKT condition). (x,u,v) is said to verify the first order Karush–Kuhn–Tucker (KKT) conditions if and only if

$$\begin{aligned} \text{(Stationarity)} \quad & 0 = \nabla f(x) + \sum_{i=1}^m u_i \nabla g_i(x) + \sum_{j=1}^r v_j \nabla h_j(x) \\ \text{(Complementary slackness)} \quad & u_i g_i(x) = 0 \quad \forall i \quad . \quad \text{(KKT)} \\ \text{(Primal feasibility)} \quad & g_i(x) \leq 0, \quad h_j(x) = 0 \quad \forall i, j \\ \text{(Dual feasibility)} \quad & u_i \geq 0 \quad \forall i \end{aligned}$$

In order to use KKT conditions as necessary 1st order conditions for optimality, a feasible point \bar{x} must satisfy some assumptions, otherwise an optimization problem may have a solution that does not satisfy the [1st order KKT condition](#). Since there are many ways to define a feasible set using constraints, it is advisable to define them in such a way that the KKT conditions hold. These assumptions are known as constraint qualification, and they ensure that the feasible set is well approximated by the actual formulation of the constraints. Let us introduce the two most popular ones:

Definition 2.2.5 (MFCQ). A design point \bar{x} is said to satisfies the Mangasarian Fromovitz Constraint Qualification (MFCQ) if and only if

- The gradients of the equality constraints h_j are linearly independent at the point \bar{x} .
- There is a vector $d \in \mathbb{R}^n$ such that $\nabla h_j(\bar{x})^T d = 0$ for all equality constraints and $\nabla g_i(\bar{x})^T d \leq 0$ for all active inequality constraints.

Definition 2.2.6 (LICQ). A design point \bar{x} is said to satisfies the Linear Independence Constraint Qualification (LICQ) if and only if the gradients of the active constraints,

$$\begin{aligned} & \nabla h_j(\bar{x}) && \text{for all } j \\ & \nabla g_i(\bar{x}) && \text{for all } i \text{ such that } g_i(\bar{x}) = 0' \end{aligned} \quad (2.2.53)$$

are linearly independent.

For every feasible point \bar{x} there is the well-known relationship:

$$LICQ(\bar{x}) \implies MFCQ(\bar{x}). \quad (2.2.54)$$

The **LICQ** is generally considered to be the strongest constraint. It guarantees the existence and also the uniqueness of the Lagrange multipliers at the solution. The **MFCQ** is also often used instead of the **LICQ**, although this results in the loss of the desired uniqueness of the Lagrange multipliers. Throughout the manuscript we will mostly rely on **LICQ** for most of the results given, **MFCQ** will eventually be mentioned for classical results within Chapter 3. In some cases, such as in the convex case (or more generally for the wider class of type 1 invex functions), **1st order KKT condition** are also sufficient [100].

Let us now assume that **LICQ** is verified on some design point x , by further assuming that the involved functions are at least C^2 functions, second order conditions can be considered to ensure that the previous solution is a minimizing point. These second order conditions, also referred as second order KKT conditions, can take different forms depending on the problem considered. One of the most classic 2^{nd} order sufficient conditions will be considered in this manuscript.

Definition 2.2.7 (2^{nd} order KKT conditions). Assume that (x, u, v) verifies **LICQ** and the first order KKT conditions. (x, u, v) is said to verify the 2^{nd} order KKT conditions if and only if every direction $d \neq 0$ such that

- $\nabla h_j(x)^T d = 0$ for all j
- $\nabla g_i(x)^T d = 0$ for all i such that $g_i(x) = 0$ and $u_i > 0$
- $\nabla g_i(x)^T d \leq 0$ for all i such that $g_i(x) = 0$ and $u_i = 0$

also verifies:

$$d^T \nabla^2 \mathcal{L}(x, u, v) d > 0, \quad (2.2.55)$$

where $\nabla^2 \mathcal{L}(x, u, v)$ denotes the bordered Hessian of (2.2.49) at (x, u, v) :

$$\nabla^2 \mathcal{L}(x, u, v) = \begin{bmatrix} \nabla_{x,x}^2 \mathcal{L}(x, u, v) & [\nabla g(x) \quad \nabla h(x)]^T \\ [\nabla g(x) \quad \nabla h(x)] & 0_{(r+m)^2} \end{bmatrix}. \quad (2.2.56)$$

According to the definition of the admissible direction d , the 2^{nd} order KKT condition described above is strictly equivalent to

$$d^T \nabla_{x,x}^2 \mathcal{L}(x, u, v) d, \quad (2.2.57)$$

being positive definite. In other words, the Hessian of \mathcal{L} along x must be positive definite when considering feasible directions. Moreover it is known that if the Hessian of the Lagrangian with respect to x , $\nabla_{x,x}^2 \mathcal{L}(x, u, v)$, is positive definite, then the matrix $d^T \nabla_{x,x}^2 \mathcal{L}(x, u, v) d$ is positive definite even though the converse is not necessarily true.

2.2.4 Three analysis theorems

Here are 3 analysis theorems related to the definition of continuously differentiable functions from a set of hypotheses, namely the Implicit Function Theorem (**IFT**), Fiacco's Theorem, and the Envelope Theorem. Each of these is relevant and used at some point in the manuscript.

The implicit function theorem

A classical result lies in the already introduced Implicit Function Theorem (**IFT**):

Theorem 2.2.5 (Implicit Function Theorem). *Let $R : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$ be a continuously differentiable function, and let \mathbb{R}^{n+m} have coordinates (x, y) .*

Fix a point (a, b) such that $R(a, b) = 0_{\mathbb{R}^m}$. If the Jacobian matrix $\left[\frac{\partial R_i}{\partial y_j}(a, b) \right]_{(i,j) \in [1,m]^2}$ is invertible, then there exists an open set $\mathcal{U} \subset \mathbb{R}^n$ and a unique continuously differentiable function $\Psi : \mathcal{U} \rightarrow \mathbb{R}^m$ such that $\Psi(a) = b$ and $R(x, \Psi(x)) = 0$ for all $x \in \mathcal{U}$.

The **Implicit Function Theorem** is a classical theorem in analysis that allows to define a subset of variables as a continuously differentiable function from a set of equations that must be satisfied. This is of primary importance in MDO, as most architectures are based on using one or more layers of this theorem to eliminate variables. For example, it allows to eliminate state variables in classical MDO architectures, or, in MDF-based architectures, to redefine the coupling variables as an implicit function of the design, which is later solved by a dedicated algorithm called multidisciplinary analysis (or MDA). This is discussed in greater detailed in Section 2.1.1, dedicated to general MDO concepts.

Fiacco's theorem

Considering the aforementioned **Implicit Function Theorem**, it is possible to apply it to the optimality conditions of a parameterized nonlinear optimization problem. Under certain regularity conditions of the solution on a neighborhood of the parameter, the optimal primal and dual variables, with respect to a change in the parameter, are in fact continuously differentiable functions of the parameter in said neighborhood. This result is known as Fiacco's theorem [47]:

Theorem 2.2.6 (Fiacco's theorem). *Let the following parameterized optimization problem:*

$$\begin{aligned} \min_x & f(x, \alpha) \\ \text{s.t.} & g_i(x, \alpha) \leq 0 \quad i = 1, \dots, n_g, \\ & h_j(x, \alpha) = 0 \quad j = 1, \dots, n_h \end{aligned} \quad (P(\alpha))$$

with $x \in \mathbb{R}^n$ be the design variable and $\alpha \in \mathbb{R}^p$ be a parameter vector.

Without loss of generality, consider the case $\alpha = 0$ and the resulting optimization problem $P(0)$. Assume that there exist x^ such that $x \mapsto f(x, 0)$, $x \mapsto g_i(x, 0)$, $i = 1, \dots, n_g$ and $x \mapsto h_j(x, 0)$, $j = 1, \dots, n_h$ are all twice continuously differentiable in a neighborhood of x^* .*

Further assume that:

1. f , g_i , $i = 1, \dots, n_g$ and h_j , $j = 1, \dots, n_h$ are all twice continuously differentiable in (x, α) in a neighborhood of $(x^*, 0)$.
2. x^* be a local minima of the problem $P(0)$, hence there exist λ_i^* , $i = 1, \dots, n_g$ and μ_j^* , $j = 1, \dots, n_h$ be the so-called Lagrange multipliers associated with g_i , $i = 1, \dots, n_g$ and h_j , $j = 1, \dots, n_h$ respectively such that (x^*, λ^*, μ^*) verifies both **1st order KKT condition** and **2nd order KKT conditions**.
3. The gradients of the active constraints at $(x^*, 0)$ are linearly independent (cf. **LICQ**).

4. *Strict complementary slackness holds at $(x^*, 0)$*

Hence it follows that:

1. x^* is a local minimizer of $P(0)$ and the associated Lagrange multipliers, λ^* and μ^* are unique.
2. For α in a neighborhood of 0 there exist a unique triplet of once continuously differentiable function $[x(\alpha), \lambda(\alpha), \mu(\alpha)]$ satisfying *2nd order KKT conditions* of problem $P(\alpha)$ such that $[x(0), \lambda(0), \mu(0)] = (x^*, \lambda^*, \mu^*)$. In other words, $x(\alpha)$ is the unique local minimum of $P(\alpha)$ with associated unique Lagrange multipliers $\lambda(\alpha)$ and $\mu(\alpha)$.
3. *Strict complementarity (with respect to $\lambda(\alpha)$ and the inequality constraints) and LICQ hold at $x(\alpha)$ for α near 0.*

As noted, Fiacco's theorem is a direct application of [Implicit Function Theorem](#), which gives a first-order sensitivity analysis of a second-order local solution with respect to a parameter. This is a very strong theorem because it guarantees that the arg min operator is indeed a continuously differentiable function with respect to the parameter. However, this result comes at the cost of strong hypotheses. Apart from verifying both [LICQ](#) and [2nd order KKT conditions](#), which is not always the case, [Fiacco's theorem](#) requires that the set of active constraints remains constant as the parameter varies in the neighborhood considered.

The envelope theorem

The third theorem is the so-called envelope theorem [\[144\]](#):

Theorem 2.2.7 (Envelope theorem). *Let f and g_i , $i \in \llbracket 1, m \rrbracket$ be continuously differentiable functions. We consider the optimization problem for a given α :*

$$\begin{aligned} \min_x f(x, \alpha) \\ \text{s.t. } g_i(x, \alpha) = 0 \quad i = 1, \dots, m \end{aligned} \quad (2.2.58)$$

with $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}^p$, and the associated Lagrangian,

$$\mathcal{L}(x, \lambda, \alpha) \equiv f(x, \alpha) + \lambda^T g(x, \alpha), \quad (2.2.59)$$

where $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers.

We denote $x^*(\alpha)$ and $\lambda^*(\alpha)$ the solution that minimize the objective function f .

Thus we define

$$\mathcal{L}^*(\alpha) \equiv f(x^*(\alpha), \alpha) + \lambda^*(\alpha) \cdot g(x^*(\alpha), \alpha), \quad (2.2.60)$$

and the value function:

$$V(\alpha) \equiv f(x^*(\alpha), \alpha). \quad (2.2.61)$$

Then if V and \mathcal{L} are continuously differentiable, it follows that

$$\frac{\partial V}{\partial \alpha_k}(\alpha) = \frac{\partial \mathcal{L}^*}{\partial \alpha_k}(\alpha) = \frac{\partial f}{\partial \alpha_k}(x^*(\alpha), \alpha) + \lambda^*(\alpha) \cdot \frac{\partial g}{\partial \alpha_k}(x^*(\alpha), \alpha) \text{ for } k \in \llbracket 1, p \rrbracket. \quad (2.2.62)$$

The [Envelope theorem](#), mostly known and used in economics, is an important result about the differentiability properties of the value function of a parameterized optimization problem. Given a certain regularity stated in the theorem, this result allows to derive the derivative of the optimal value of an optimization problem with respect to a parameter, stating that this partial derivative is equal to that of the optimal Lagrangian value with respect to the same parameter. In other words, it allows to ignore the dependence of the optimal design variables x^* and the optimal Lagrangian multipliers λ^* on said parameter.

Chapter 3

A bi-level architecture

Summary: This chapter examines so-called "distributed" MDO architectures where multiple optimization problems are present. In particular, bi-level architectures are considered, where one optimization problem (the lower-level problem) is nested within another (the system optimization problem), and their convergence properties are discussed. A pre-existing bi-level architecture, the BL-IRT architecture, is considered with respect to the target applications. A first mathematical model of the bi-level decomposition used is presented, and the equivalence of the solution between this decomposition and the monolithic MDF problem is established. Sufficient assumptions on the solutions of the lower optimization problem, given by Fiacco's theorem, are made to ensure that the system-level functions are continuously differentiable and do not degrade the convergence properties of standard algorithms. Since these properties can only be verified if the lower-level optimization problem is solved with sufficient accuracy, we propose to solve the lower-level optimization problem by block decomposition using a well-known algorithm, the Block Coordinate Descent (BCD) algorithm. It is shown that the BCD guarantees that a fixed point is necessarily feasible and that, under certain assumptions such as the unique block minimizer and the first-order separability, the local convergence of the algorithm can be proved. Several variants of the newly introduced bi-level architecture are also presented. These variants propose slight changes to the overall architecture to relax certain assumptions that are considered too restrictive, typically by using a different block decomposition that reallocates the block constraints or the coupling variables. Finally, numerical experiments on a classical benchmark MDO problem (SSBJ) demonstrate a significant reduction in the error on system functions and the resulting improved convergence properties of these architectures. A final experiment illustrates the value of so-called bi-level architectures in terms of scalability when coupled derivatives are not available and the number of local variables, coupling variables, and constraints increases.

Résumé: Ce chapitre examine les architectures MDO dites "distribuées" où plusieurs problèmes d'optimisation sont présents. En particulier, les architectures classifiées comme bi-niveaux sont étudiées, dans lesquelles un problème d'optimisation (le sous-problème) est imbriqué dans un autre (le problème d'optimisation système), et leurs propriétés de convergence sont discutées. Une architecture bi-niveaux préexistante, l'architecture BL-IRT, est examinée en ce qui concerne les applications cibles. Un premier modèle mathématique de la décomposition à deux niveaux utilisée est présenté, et l'équivalence de la solution entre cette décomposition et le problème MDF monolithique est établie. Des hypothèses suffisantes sur les solutions du sous-problème d'optimisation, données par le théorème de Fiacco, sont faites pour garantir que les fonctions du niveau système sont continuellement différentiables et ne dégradent pas les propriétés de convergence des algorithmes standard. Comme ces propriétés ne peuvent être vérifiées que si le sous-problème d'optimisation est résolu avec une précision suffisante, nous proposons de résoudre le sous-problème d'optimisation par une décomposition en blocs à l'aide d'un algorithme bien connu, l'algorithme BCD (Block Coordinate Descent). Il est démontré que le BCD garantit qu'un point fixe est nécessairement faisable et que, sous certaines hypothèses telles que l'unicité du minimiseur des blocs et leur séparabilité au premier ordre, la convergence locale de l'algorithme peut être prouvée. Plusieurs variantes de l'architecture bi-niveaux nouvellement introduite sont également présentées. Ces variantes proposent de légères modifications de l'architecture globale afin d'assouplir certaines hypothèses jugées trop restrictives, généralement en utilisant une décomposition en blocs différente qui réaffecte les contraintes des blocs ou les variables de couplage. Enfin, des expériences numériques sur un problème MDO classique de référence (SSBJ) démontrent une réduction significative de l'erreur sur les fonctions du système et des propriétés de convergence améliorées qui en résultent pour ces architectures. Une dernière expérience illustre l'intérêt des architectures bi-niveaux en termes de scalabilité lorsque les dérivées couplées ne sont pas disponibles et que le nombre de variables locales, de variables de couplage et de contraintes augmente.

3.1 Introduction

In the context of MDO problems, the choice of the architecture is of particular importance, as it drives the overall optimization process, the data exchanges, and the allocation of computational resources [150]. In particular, when large-scale problems are encountered, these considerations become even more pronounced. Among the MDO architectures presented in Section 2.1.2, the distributed architectures have not yet been introduced. Contrary to their monolithic counterparts, the distributed architectures reformulate the original problem into several optimization problems, generally smaller, by exploiting the inherent separation within MDO problems.

The decomposition aligns with the design environment, restoring greater autonomy to each discipline and ensuring greater flexibility of the design as a whole, as evidenced by the fact that the discipline’s code is developed with the goal of achieving autonomy for most of the optimization process while incorporating updates from external sources. This is achieved by allowing for better load balancing and the use of different, well-chosen algorithms and optimization tools to solve the multiple optimization problems, possibly in parallel. These factors contribute to the continued investigation of distributed architectures in MDO, despite the perception that they are less efficient and exhibit less convergence properties than their monolithic counterparts [101]. This is largely attributed to the division of optimization problems into multiple components, necessitating the implementation of costly coordination mechanisms to ensure design coherence. In addition, the use of gradient-based approaches often requires costly post-optimal analysis due to the presence of multiple optimization problems, and the presence of strong assumptions to ensure the feasibility of the design or ill-posed problems is not uncommon. In practice, however, these architectures are still used in many industrial applications. Since monolithic architectures impose many constraints on their tools, such as access to the fully coupled adjoint, it is often preferred to use a distributed architecture that is less constraining. In other words, it may be reasonable to pay higher computational costs and lose robustness rather than aim for a theoretical computational cost of a fully coupled process by having access to a fully coupled adjoint that may never arrive.

We are interested in a particular subset of distributed architectures, the multi-level architectures, and more specifically the bi-level architectures, which rely on a hierarchical structure of the different optimization problems and their convergence properties. In practice, the computation of the objective and constraints of the system optimization problem requires the solution of another optimization problem, called the lower-level optimization problem, which is parametrized by the system design variables. These nested architectures, reminiscent of Bender’s decomposition [57], include architectures such as the Collaborative Optimization (CO) architecture [24] or the Bi-Level Integrated System Synthesis (BLISS or BLISS98) architecture [139] and its, quite different successor, BLISS2000 [140].

As part of this category, the Bi-Level IRT (BL-IRT) [56] architecture has been developed with the specific intention of addressing the challenges posed by large-scale MDO problems, where the shared design variables are considered to be relatively low-dimensional in comparison to the number of couplings and local variables. The BL-IRT architecture is strongly inspired by the BLISS architecture, wherein the distinction between shared and local variables is achieved through a system and a lower optimization problem. This latter aspect is further decomposed into multiple block optimizations, each driven by a

single discipline and its own set of local variables. This approach effectively eliminates the necessity to have access to all coupled derivatives at the system level while allowing its use at the lower levels. However, there are two key differences between the two architectures. Firstly, the BL-IRT architecture involves purely non-linear sub-optimization problems, rather minimizing linear approximations of the objective. Secondly, the system optimization problem is expected to be solved using a gradient-free approach, which eliminates the necessity for costly post-optimal derivatives.

If this newly proposed architecture has been successfully applied to a large-scale, high-fidelity aircraft pylon optimization using industrial simulation software [55], it also encounters convergence problems due to the lack of coordination mechanisms between the block optimizations within the lower-level optimization problem, which is handled by only two MDAs before and after the block optimizations, respectively. This results in the emergence of noisy system functions (cf. Section 3.6.1) due to the high dependency of the lower-level resolution on the path taken by the local variables, potentially preventing convergence of the system optimizer (cf. Section 3.6.1).

This is the main reason for developing a novel bi-level architecture, which is closely related to the aforementioned BL-IRT architecture, with the objective of addressing these convergence issues. The overall goal is to replace the lower optimization problem resolution method to a more robust one, potentially ensuring convergence properties of the entire bi-level optimization problem if certain sets of hypothesis are met. This chapter presents a new bi-level architecture, which builds upon the aforementioned BL-IRT architecture. The objective is to replace the lower optimization problem resolution method with a more robust one, which may ensure convergence properties of the entire bi-level optimization problem if certain sets of hypotheses are met. The Block Coordinate Descent (BCD) [16, 64, 110, 111, 113, 17, 18] algorithm has been selected to achieve this goal, leading to the newly proposed MDF-based bi-level architecture that we will name Bi-Level - Block Coordinate Descent - Multi-Disciplinary Feasible (or BL-BCD-MDF in short) architecture (cf. Section 3.4.1).

The selection of the BCD algorithm to solve the lower optimization problem is motivated by the fact that the Gauss-Seidel approach, which shares the same underlying principles but differs in its application to non-linear system of equations, is well-known in the MDO community. Therefore, the BCD algorithm benefits from the already abundant literature on the Gauss-Seidel method, which, to some extent, can be directly applied to the first order optimality conditions of block optimizations. Similarly, acceleration techniques for fixed point solvers [127] that have already been developed and/or implemented require only minor adaptations. Finally, the BCD algorithm does not rely on coupled derivatives to solve the lower optimization problem in the same way that gradient-based approaches such as MDF do. This is consistent with the objective of employing a block decomposition approach to solve the lower optimization problem. The joint problem is characterized by a considerable number of local and coupling variables, rendering even gradient-based algorithms ineffective due to the prohibitive computational cost associated with the coupled Jacobian.

One of the principal objectives behind the deployment of the BCD algorithm is to improve the overall convergence properties of the proposed bi-level decomposition. These include enhanced regularity for system functions (cf. Section 3.3.2) and better convergence properties for both the system and lower optimization problems (cf. Section 3.4.2).

This chapter is partitioned as follows: the first section (cf. Section 3.2.1) presents a brief overview of the different classical distributed architectures, with a focus on the BL-IRT to tackle the targeted use cases. This is directly followed by the presentation of the usual formalization for bi-level programming. This investigation led to the conclusion that an implicit reformulation of the lower-optimization problem would be the most appropriate approach. Once both the bi-level strategy and formalism have been established, Section 3.3 presents the decomposition steps to transition from a generic MDF optimization problem to the desired bi-level framework, as presented in the original paper [56]. This section demonstrates the equivalence of the solutions of the two optimization problems. It also provides sufficient conditions for ensuring that all system functions are continuously differentiable. Section 3.4 examines two approaches to the block resolution of the lower optimization problem: the distributed method employed in BL-IRT and the newly proposed BCD algorithm. The well-definiteness of the BCD algorithm, the feasibility of the solution, and a local convergence result are demonstrated under a series of assumptions that can be justified by industry standards. The following section, Section 3.5, proposes three variants for the BL-BCDMDF architecture, allowing the assumptions to be modified according to specific requirements and prior expertise. Finally, in Section 3.6, two classical MDO problems are considered to evaluate the noise reduction and enhanced robustness of the presented bi-level framework in comparison to the previous BL-IRT. A scalability study is also performed on a newly created scalable problem that extends the well-known Sellar problem [135], which now considers two blocks of local variables, to demonstrate the capacity of bi-level decomposition when the number of shared variables is significantly smaller than the number of local variables, coupling variables, and constraints.

3.2 State of the art: distributed architectures for large scale MDO problem

3.2.1 Distributed architectures

Due to their composite nature, the distributed architectures appear to be well-suited to solve high-dimensional problems: difficulties are divided among several computational resources and each subproblem is solved with an appropriate algorithm, potentially in a distributed manner, one could expect these architectures to be, a priori, faster than their monolithic counterparts when solving high-dimensional MDO problems. Furthermore, the inherent structure of general MDO problems readily suggests methods for decomposing the problem into several subproblems, according to disciplinary, variable type, or gradient availability.

For all these reasons, and because of the vast possibilities it offers to reformulate the original optimization problem, distributed architectures have been the subject of extensive research over the past decades. A direct classification of such architectures can be derived from the monolithic one [101]. The distributed-MDF architectures regroup the distributed architectures which solve couplings through MDAs, while the distributed-IDF architectures regroup the ones which rely on defining target values and consistency constraints. Another classification, introduced by *Tosserams and al.* [147], distinguishes the distributed architectures on three distinct factors.

- The first factor depends on the interactions between the subproblems. One approach

is to solve the optimization subproblems in a nested fashion. This involves establishing a hierarchy, as seen in the multilevel methods, where each new iterate of an optimization problem triggers a complete optimization of the ones just below it in the hierarchy. Alternatively, the subproblems can be solved in an alternating fashion, with each subproblems solved entirely one after the other until convergence.

- Secondly, distributed architectures can be classified according to whether the optimizers exert control over the entire design vector (open design) or whether some parameters are beyond the optimizer’s control (closed design) and are typically handled by another component of the process other than the optimization algorithm directly.
- Finally, the distributed architectures may also be categorized as to whether the optimizers have control over the consistency constraints (open consistency, e.g. IDF) or not (closed consistency, e.g. MDF) similarly as in the previous classification.

As with the monolithic architectures, the choice of a particular distributed architecture is highly problem dependent [150]. A multitude of considerations must be taken into account when choosing the optimal architecture, including the number of strongly coupled disciplines, the separability assumptions, and the mathematical properties of the considered functions and equations. One of the main objectives of this work is to provide a theoretical framework for the study of hypotheses, on the different models/functions involved, as well as on the structure of the problem addressed and the results that could be obtained, especially in terms of convergence.

The formulation of multiple optimization problems in an MDO architecture typically exploits the separation between the shared variables (the variables that are inputs for multiple disciplines) and the local variables (the variables that are inputs for a single discipline) by assigning them to different optimization problems. In the following sections, the subproblem responsible for handling the linking variables (the shared design variables and potentially the couplings) will be referred to as the system subproblem, while the subproblems responsible for the local variables and the disciplinary analysis will be referred to as the disciplinary subproblems. The choice of whether to rely on an MDA or on consistency constraints to solve the couplings has a significant impact on the design, due to the differing approaches taken in each case. In the former, the variables are considered as a linking implicit function, with the consistency constraint therefore being closed. In the latter, the variables are considered as linking variables, necessitating a coordination mechanic.

In order to illustrate the vast possibilities offered by the distributed architectures, a brief overview of the most well-known examples is presented hereafter. For more exhaustive overviews on classical MDO distributed architectures and their particularities, see [101, 147, 9].

Distributed-IDF architectures

In the context of IDF-based distributed architectures, the linking variables are considered as coupling variables. These distributed architectures are more extensively considered than the MDF-based ones, primarily due to the advantageous properties of making copies of the linking variables on separability.

Collaborative Optimization (CO) The Collaborative Optimization (CO) architecture is a straightforward architecture for general MDO problems. Introduced by Braun [24], copies of the shared variables x_0 and couplings y are made, the so-called target values x_0^t and y^t respectively. The former is shared to every disciplinary subproblem, while the latter is shared to the disciplinary subproblem in charge of computing the corresponding disciplinary analysis. Each disciplinary subproblem is responsible for controlling its own local design variables, x_i ($i \in 1 \dots p$), for computing its own disciplinary analysis and coupling variables, $\phi_i(x_0, x_i, y_{\neq i}^t)$. It is also responsible for ensuring the satisfaction of its own disciplinary constraints and for minimizing the system inconsistency in both x_i and y_i . The system subproblem is tasked with minimizing the objective function in both the linking variables (x_0 and y) and coordinating the activities of all the disciplinary subproblems.

The CO system subproblem is defined as follows:

$$\begin{aligned} \min_{(x_0, x^t, y^t)} \quad & f(x_0, x^t, y^t) \\ \text{subject to} \quad & g_0(x_0, x^t, y^t) \leq 0 \\ & J_i^*(x_0, x_i^t, y^t) = 0 \quad i = 1 \dots p \end{aligned} \quad (P_{CO}^0)$$

while the disciplinary subproblems ($i \in 1 \dots p$) are given by:

$$\begin{aligned} J_i^*(x_0, x_i^t, y^t) = \min_{(x_{0,i}^t, x_i)} \quad & J_i(x_{0,i}^t, x_i, \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t); x_0, x_i^t, y^t) \\ \text{subject to} \quad & g_i(x_{0,i}^t, x_i, \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t)) \leq 0 \end{aligned} \quad (P_{CO}^i(x_0, x_i^t, y^t))$$

where

$$J_i(x_{0,i}^t, x_i, y_i; x_0, x_i^t, y_i^t) = \|x_{0,i}^t - x_0\|_2^2 + \|x_i - x_i^t\|_2^2 + \|y_i - y_i^t\|_2^2. \quad (3.2.1)$$

It has been shown that (P_{CO}^0) is equivalent to (IDF) [24] in the sense that it has the same set of optimal solutions. For its simplicity and flexibility, CO has been the subject of extensive studies, with numerous variations developed to address the main issues. The main one is that each design variable is assigned to the system subproblem, there is a significant number of equality constraints to be handled by the system, which can lead to instability at convergence. In addition, without further assumptions, disciplinary subproblems may be infeasible and the process would fail. Finally, the efficiency of the CO method decreases as the number of coupling variables increases. Among the variants, [137] proposed the use of surrogates to approximate the post-optimal derivatives of the disciplinary subproblems, [23] relaxed the system subproblem's consistency constraints by using a penalty parameter. In the latter approach, the management of the penalty parameter has been improved in [86] and [160]. The Enhanced Collaborative Optimization (ECO) [130] architecture switches roles between the system and the disciplinary subproblems. The former is now an unconstrained optimization problem, tasked with minimizing inconsistency. The latter minimizes a quadratic approximation of the objective and inconsistency along its own set of variables. Linear approximations of other disciplinary subproblems are also added to increase awareness.

BLISS-2000 Another prevalent nested IDF-based distributed architecture lies in the BLISS-2000 architecture presented in [140]. This approach entails minimizing the objective

along the system-wise variables x_0 while the coupling functions ϕ_i are replaced by surrogates models. Additionally, target values for the coupling variables are incorporated to enforce the consistency constraints.

The surrogates constructed for the ϕ_i are designed to minimize the objective along x_i by minimizing a weighted sum of their outputs with a well-chosen weight vector w_i . Each weight is selected to reflect the impact of the disciplinary output on the system objective. The real coupling analysis ϕ_i is used in the disciplinary subproblems to construct the corresponding surrogate $\tilde{\phi}_i$ by a user-defined construction method.

Consequently, the disciplinary subproblems are supplanted by sets of response surface-based models that approximate their optimized outputs along the x_i from the inputs provided by the system optimizer. Throughout the optimization process, the response surfaces can be enhanced by the addition and removal of points.

The BLISS-2000 system subproblem is defined as follows:

$$\begin{aligned} \min_{(x_0, y^t, w)} \quad & f(x_0, x, \tilde{\phi}_{1:p}(x_0, x, y^t, w)) \\ \text{subject to} \quad & g_0(x_0, x, \tilde{\phi}_{1:p}(x_0, x, y^t, w)) \leq 0 & (P_{BLISS-2000}^0(x)) \\ & y^t - \tilde{\phi}_i(x_0, x_i, y_{\neq i}^t) = 0 \quad i = 1 \dots p \end{aligned}$$

while the disciplinary subproblems ($i \in 1 \dots p$), that compute the local variables x_i , are given by:

$$\begin{aligned} \min_{(x_i)} \quad & w_i^T \phi_i(x_0, x_i, y_{\neq i}^t) \\ \text{subject to} \quad & g_i(x_0, x_i, \phi_i(x_0, x_i, y_{\neq i}^t)) \leq 0 & (P_{BLISS-2000}^i(x_0, y_{\neq i}^t, w_i)) \end{aligned}$$

The BLISS-2000 architecture is highly dependent on the surrogates constructed during the optimization process, which may require manual validation by experts to be effective. Consequently, the optimization process is significantly influenced by the quality of the response surfaces that approximate the solutions of the disciplinary subproblems. Therefore, BLISS-2000 reaches its limits when problems with large numbers of functions and/or design variables are encountered.

Analytical Target Cascading (ATC) In the absence of a specific hierarchy in the subproblems, the penalty relaxation methods may be employed instead. One such method is the Analytical Target Cascading (ATC) approach [146, 148], which is a multi-level MDO method that hierarchically propagates system and sub-system level targets through the different subsystem levels. The specified design targets are propagated from the system level to the lower levels and are also sent back to the higher levels after being optimized at the lower levels. At each level of the design process, a specific optimization problem is formulated to minimize the discrepancies between the level outputs and the propagated objectives. This ensures the consistency of the coupling variables between the upper and lower optimization levels.

The ATC system subproblem is defined as follows:

$$\begin{aligned} \min_{(x_0, y^t)} \quad & f_0(x_0, x_1, \dots, x_p, y^t) + \sum_{i=1}^p \epsilon_i (x_{0,i}^t - x_0, y_i^t - \phi_i(x_0, x_i, y_{\neq i}^t)) + \\ & \epsilon_0 (g_0(x_0, x_1, \dots, x_p, y^t)) & (P_{ATC}^0(x_0^t, x)) \end{aligned}$$

while the disciplinary subproblems ($i \in 1 \dots p$), which compute their own set of local variables x_i and shared variables $x_{0,i}^t$, are given by:

$$\begin{aligned}
& \min_{(x_{0,i}^t, x_i)} && f_0(x_{0,i}^t, x, y_{<i}^t, \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t), y_{>i}^t) && + \\
& && f_i(x_{0,i}^t, x_i, \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t)) && + \\
& && \epsilon_i(y_i^t - \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t), x_{0,i}^t - x_0) && + && (P_{ATC}^i(y_{\neq i}^t)) \\
& && \epsilon_0(g_0(x_{0,i}^t, x, y_{<i}^t, \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t), y_{>i}^t)) \\
\text{subject to} && g_i(x_{0,i}^t, x_i, \phi_i(x_{0,i}^t, x_i, y_{\neq i}^t)) \leq 0
\end{aligned}$$

where ϵ_i ($i \in 1 \dots p$) is a penalty relaxation for the discipline i consistency constraints and ϵ_0 a penalty relaxation of the global design constraint.

The Exact and Inexact Penalty Decomposition (EPD and IPD) architectures [36] can be considered as nested version of the ATC architecture. In these architectures, a L_1 and quadratic penalty function are used, respectively, to ensure that all copies of a given variable converge to the same value.

The convergence properties of these two types of architectures have been extensively studied in the context of quasi-separable problems, which are defined as problems in which the system-wide objective function (f_0) or the constraints (g_0) are absent. It has been demonstrated that the EPD/IPD architectures are capable of converging to a first-order KKT point under certain conditions, including the LICQ. The convergence results for quasi-separable problems presented in [146] demonstrate the efficacy of an augmented Lagrangian strategy involving the creation of copies of all linking variables (x_0, y) for each subproblem. For each of these copies, an affine constraint is added to enforce coherence, which can be computed in parallel.

The objective is set in the form of an augmented Lagrangian, where a relaxation of the coupling constraint is added. The augmented Lagrangian is solved by ADMM [15]. For each penalty parameter, a one-pass Block Coordinate Descent [16] (also known as Alternating Optimization [17, 18] or Block Gauss Seidel [113, 111]) is executed over a subproblem that minimizes the relaxed consistency constraints (P_0) and the minimization of p disciplinary subproblems in parallel (P_i), each of which yields a new value of x_i and y_i . The disciplinary subproblems are responsible for minimizing the objective along the disciplinary variables (x_i and y_i) while satisfying the corresponding disciplinary constraints. These subproblems are penalized by the relaxed consistency constraints over y_i and are made independent by the copies and coordination mechanics in P_0 . The proposed method has been generalized to the case with coupling objective and constraints [148]. This generalization comes at the price of more costly inner loop, which is necessary to ensure convergence towards a KKT point.

A common limitation of the IDF-based architectures is the generation of copies of the linking variables for each subproblem. As the number of coupling variables increases, the number of consistency constraints and added design variables (referred to as target-values) can become a significant challenge for most optimization algorithms, even for gradient-based ones [34, 150]. The objective of this study is to address the challenges posed by large-scale MDO problems, which include a significant number of coupling variables. Consequently, the focus has been placed on the investigation of the MDF-based strategies for coupling management.

Distributed-MDF architectures

In contrast, distributed-MDF architectures employed fixed-point algorithms to guarantee the feasibility of the design under consideration.

Concurrent Subspace Optimization (CSSO) One of the oldest distributed-MDF architectures is the Concurrent Subspace Optimization (CSSO) architecture proposed by Sobieski [138]. The fundamental principle of this approach is to utilize approximations of the disciplinary analysis to optimize the original optimization problem component by component in parallel.

The attached algorithm alternates between solving the system subproblem in the entire design variable vector x using an approximation of the coupling variables and the parallel optimization of disciplinary subproblems, which are solved with respect to the shared design variables x_0 and its local variable block x_i ($i \in 1 \dots p$). The consistency constraints are guaranteed to be feasible by employing a MDA between the system and the disciplinary subproblems optimization, as well as by utilizing the real disciplinary analysis function for the corresponding coupling variable in each disciplinary subproblem. Ultimately, the results of the system analysis are employed to enhance the disciplinary analysis approximation.

The CSSO system subproblem is defined as follows:

$$\begin{aligned} \min_{(x_0, x)} \quad & f(x_0, x, \tilde{\Psi}(x_0, x, \tilde{y})) \\ \text{subject to} \quad & g_0(x_0, x, \tilde{\Psi}(x_0, x, \tilde{y})) \leq 0 \end{aligned} \quad (P_{CSSO}^0(\tilde{y}))$$

while the disciplinary subproblems ($i \in 1 \dots p$), which consider both the shared variables x_0 and its own set of local variables x_i , are given by:

$$\begin{aligned} \min_{(x_0, x_i)} \quad & f(x_0, x_i, \tilde{y}_{1:i-1}, \phi_i(x_0, x_i, \tilde{y}_{\neq i}), \tilde{y}_{i+1:p}) \\ \text{subject to} \quad & g_0(x_0, x_i, \tilde{\Psi}(x_0, x, \tilde{y})) \leq 0 \\ & g_i(x_0, x_i, \phi_i(x_0, x_i, \tilde{y}_{\neq i})) \leq 0 \\ & g_j(x_0, x_j, \tilde{\Psi}_j(x_0, x, \tilde{y})) \leq 0 \quad j \neq i \end{aligned} \quad (P_{CSSO}^i(\tilde{y}_{\neq i}))$$

where $\tilde{\Psi}$ is an approximation of the real MDA, that uses all the real coupling functions ϕ_i $i \in (1, \dots, p)$, and \tilde{y}_i ($i \in 1 \dots p$) are the approximations of the coupling variables y_i , whose dependencies, similar to ϕ_i , have been hidden to simplify the notation. The MDA was originally approximated by linear approximations, using post-optimal analysis at each iteration [138]. More general surrogates with a higher degree of information can be used instead [135], eventually the system analysis evaluations are used to improve their accuracy during the optimization process.

The CSSO has a number of inherent limitations. One such limitation is that the system level problem deals with all the design variables and constraints simultaneously, while each disciplinary subproblem also considers all constraints. This indicates that the approximation of the coupling variables may not be worth the introduction of a two optimization alternating process in terms of scalability, which can be observed in practice [145]. Furthermore, the CSSO architecture is susceptible to a lack of convergence properties, particularly when the utilized surrogates are inadequately constructed [33].

Bi-Level Integrated System Synthesis (BLISS) A more popular approach within the distributed-MDF family, also proposed by Sobieski, is the well-known Bi-Level Integrated System Synthesis (BLISS or BLISS98) architecture [139, 147, 101, 9]. This BLISS architecture, also proposed by *Sobieski and al.* [139], is markedly distinct from the BLISS-2000 architecture proposed two years later.

The BLISS98 architecture assigns the shared design variables x_0 to the system subproblem and the local variables x_i ($i \in 1 \dots p$) to the disciplinary subproblems. Each subproblem handles its set of local variables x_i and constraints c_i with an additional bound constraint, which is analogous to a trust region mechanism. In contrast to CSSO, each disciplinary subproblem considers only the linking constraints and the constraints that depend directly on its set of local variables.

The workflow is based on an alternating method, as follows: the sub-problems ($i \in 1 \dots p$) first optimize, in parallel, the contribution of each of the local design variables x_i ($i \in 1 \dots p$) on the objective for a fixed value of x_0 . This is followed by a post-optimal analysis, which is used to build a linear approximation of the objective in x_0 which is then optimized. The process is repeated until convergence is achieved. The overall workflow is given in Figure 3.1:

A generic XDSM diagram is also provided in Figure 3.2.

Initially, two methods were proposed to obtain the post-optimal derivatives, i.e. , the derivatives of the optimal value function of the local variables x_i^* with respect to the shared variables x_0 . The first method employs the Generalization of Global Sensitivity Equation (BLISS-A), while the second method is based on the Lagrange multipliers as prices (BLISS-B). It should be noted that this is, to our knowledge, the first appearance of the coupled adjoint in a generic MDO architecture.

The disciplinary subproblems separation, regarded as black boxes by the system subproblem, permits the utilization of specific optimization tools for each one of them without modification of the overall design. Each subproblem minimizes a linear approximation of the objective over a non-linearly constrained feasible set, resulting in a generally low resolution cost for the subproblems. However, the non-linear constraints may be challenging to satisfy.

One of the inherent drawbacks of the BLISS architecture is that it heavily relies on linear approximations, which must be kept up to date. In the event of strong non-linearity, this can result in a slowdown in the convergence process. Moreover, in addition to running MDAs, numerous costly post-optimal derivatives are necessary, which is why BLISS is expected to be more effective in cases where there is a small number of global design variables x_0 [9]. The overall cost can be reduced by using surrogates to replace the disciplinary and post-optimal analysis [79, 80].

Building upon these observations on BLISS, [56] proposed a Bi-Level distributed MDO architecture, termed the bi-level IRT architecture. This architecture employs a similar separation as BLISS with nested optimization problems but no longer relies on linear approximations for the sub-optimizations. Instead, each subproblem minimizes the original objective function along its own block of variables. The optimization process is as follows: at each system choice x_0 , the objective is minimized along all block directions x_i ($i \in 1 \dots p$) in parallel. Prior to and/or subsequent to the optimization of the disciplinary subproblems,

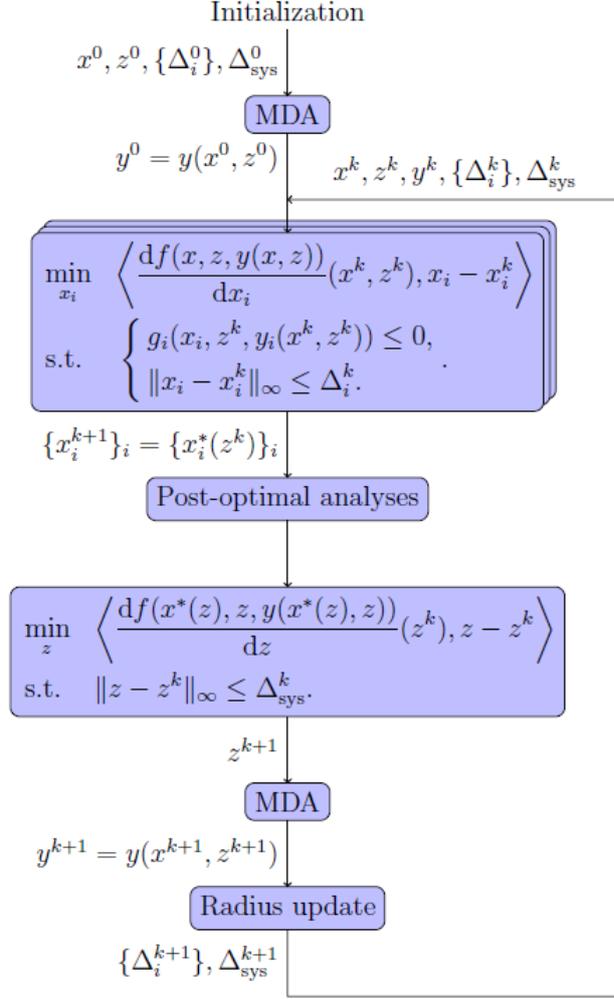


Figure 3.1: Workflow for the BLISS architecture

MDAs are employed to guarantee consistency.

As only the bi-level decomposition and the computational process were given in the original paper [56], we propose to modelize the BL-IRT system subproblem as follows:

$$\begin{aligned}
 & \min_{(x_0)} && f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \\
 & \text{subject to} && g_0(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \leq 0
 \end{aligned} \tag{P_{IRT}^0}$$

where each disciplinary subproblem ($i \in 1 \dots p$) gives $x_i^*(x_0)$ as the solution of:

$$\begin{aligned}
 & \min_{(x_i)} && f(x_0, x_i, y_{<i}, \phi_i(x_0, x_i, y_{\neq i}), y_{>i}) \\
 & \text{subject to} && g_i(x_0, x_i, \phi_i(x_0, x_i, y_{\neq i})) \leq 0
 \end{aligned} \tag{P_{IRT}^i(x_0)}$$

As mentioned before, all coupling variables y_i ($i \in 1 \dots p$), other than the one under consideration, are kept fixed in each block optimization ($P_{IRT}^i(x_0)$), their current values being the solution of an MDA computed on the previous iterate of the design variable

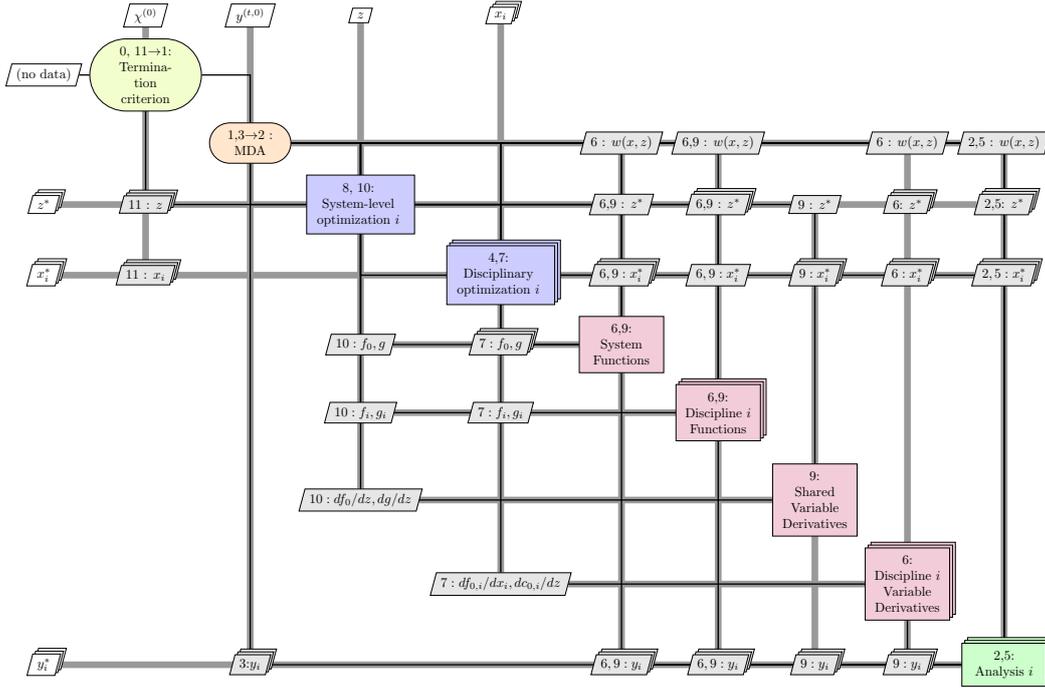


Figure 3.2: A general XDSM diagram for the BLISS architecture [101]

(x_0, x) . This dependency is not shown in the optimization problems as it is completely hidden from the system's perspective. Similar to BLISS, the BL-IRT architecture benefits from low-dimensional shared variables x_0 . This allows the use of gradient-based algorithms for solving the disciplinary subproblems and derivative-free algorithms for solving the system subproblem, where the number of design variables is reduced. This eliminates the need for post-optimality analysis.

In contrast to BLISS, the primary advantages of the BL-IRT architecture are its lower complexity and its flexibility: at the disciplinary level, truly non-linear optimization problems are solved without the imposition of a search direction from the system level.

One of the variants of the BL-IRT architecture, which employs a design of experiment strategy at the system subproblem level, has been successfully applied to a large-scale, high-fidelity aircraft pylon optimization using industrial simulation software [55].

One of the main pitfalls of the BL-IRT architecture is its lack of convergence properties. The coordination between disciplines is only achieved through MDAs before and after each pass of disciplinary subproblems optimization and warm-start strategies between two different system iterates. Consequently, for a specific value of x_0 the obtained design and coupling vectors may not be a solution of the lower optimization problem. In the most severe cases, the fulfilled disciplinary constraints can be broken after the coupling resolution. This can result in instances where the subproblems disagree on the feasibility of the design (cf. Section 3.6.1).

As a final recourse, the disciplinary constraints can be aggregated and addressed at the system level to preclude the acceptance of non-feasible designs [55]. However, it does not address the noise-induced impact on the system functions and the potential for convergence to be slowed down or completely prevented. Indeed, reliability and performance are often linked. The continuity of the functions of interest is often necessary for the gradient-

free optimizer convergence results, being in addition continuously differentiable is often beneficial when it is not necessary in certain cases [1, 62].

Choice of the distributed architecture

As previously stated, the targeted applications are the non-linear large-scaled MDO problems. In other words, the researched architecture must be capable of handling a large amount of constraints and couplings while taking advantage of the natural separability between the different components involved.

The choice has been made to explore the BL-IRT architecture in greater depth for several reasons:

- It is derived from the MDF strategy: as the coupling variables are considered to be numerous, it is anticipated that the MDF-based strategies will be more robust than the IDF-based methodologies. This is even more true for some of the targeted applications, such as aerostructure problems, where very effective MDAs exist due to couplings between CFD and CSM solvers that are well industrialized [81, 128, 27]. Furthermore, the multidisciplinary feasibility is enforced at each system iteration, which is a highly desirable property, especially in case of early termination of the algorithm.
- It shared some assets with the BLISS architecture:
 - The size of the system subproblem is minimized, simplifying its resolution. This is of primary importance when considering nested optimizations problems.
 - Well suited for MDO problems where shared variables are few in number.
 - It is possible to rely on already existing very efficient disciplinary solvers such as structure sizing [66, 132] or aerodynamic optimizations [73, 118].
- But also differs in several manners:
 - There is no linear approximations to be kept updated. Each disciplinary subproblem is non longer linearized and can be processed independently with well-chosen optimization algorithms and optimization tools, offering more flexibility for the overall design. A direct consequence is that a search direction is no longer imposed, allowing to handle strongly non-linear problems more effectively at the disciplinary level.
 - Post-optimal analysis are not necessary, derivative-free optimization algorithms may be employed when it is advisable, particularly at the system subproblem.
- For practical reasons: the said architecture is already implemented within the GEMSEO framework, allowing for easy extension and reproducible research results.

However, in this bi-level framework, the shared design variables at the system subproblem level imply that the lower optimization problem, which contains all disciplinary subproblems, must ensure both the design variables convergence and the consistency. Moreover, since no coordination mechanism or relaxation is done at the system level, the lower optimization problem must ensure the coordination itself and be solved as precisely as possible. Indeed, continuity of the functions of interest is often necessary for gradient-free optimizer convergence results. Demonstrating that the functions of interest are also

continuously differentiable is often advantageous when it is not necessary in certain cases [1, 62]. In accordance with the two preceding classifications, the proposed distributed architecture would be categorized as a distributed-MDF [101] and Nested, Closed Design, Closed Consistency [147], respectively.

The BL-IRT architecture does not provide these guarantees by design, and it is unclear when the potentially poor resolutions of the lower optimization problem can prevent convergence of common derivative-free optimization algorithms. The objective of this chapter considering the BL-IRT architecture are plural:

1. To propose a mathematical modelization for the BL-IRT architecture, allowing for a better comprehension of its mathematical behavior concerning the sufficient conditions for convergence, its equivalence with MDF and the regularity of the functions of interest.
2. To highlight the benefits and limitations of the approach, in particular in term of convergence properties.
3. To propose novel algorithmic solutions that preserves both the bi-level and block decomposition, while enhancing the robustness of the methodology.

As the proposed architecture is part of the bi-level programming, a brief overview of its definition, modelization choices and convergence properties is provided herewith.

3.2.2 Bi-level optimization

The bi-level optimization problems have been extensively studied over the past decades. The concept was first introduced in market economy and game theory in 1934 through the well-known Stackelberg's games [142] (or leader-follower games), which were subsequently extended beyond the scope of game theory to other scientific disciplines due to their practical and theoretical interest.

Due to research in numerical optimization, the bi-level programming emerged within this field during the 70's with one of the first work by [22]. This is a mathematical field that emerged from hierarchical (or nested) optimization problems involving two or more players. Formally, a bi-level optimization problem consists of a parametric optimization problem where the parameters are handled by another optimization problem. These two optimization problems are known by many names depending on the field of study, it has been chosen to refer to the former as the lower optimization problem (the follower) and the latter as the system optimization problem (the leader).

In its most general form, this can be expressed as follows:

$$\begin{aligned}
 & \min_{x \in \mathcal{X}, y \in \mathcal{Y}} f(x, y) \\
 & \text{s.t.} \quad g(x, y) \leq 0 \\
 & \text{with} \quad y \in \left\{ \arg \min_{y \in \mathcal{Y}} \{F(x, y) \text{ s.t. } G(x, y) \leq 0\} \right\}
 \end{aligned} \tag{Bi-level}$$

where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{Y} \subset \mathbb{R}^{n_y}$ are the design spaces for the system and the lower-optimization problem, respectively. $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ and $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ the respective objective functions and $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^{n_{cu}}$, $G : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^{n_{cl}}$ the respective constraints.

All the system level variables that appear in the lower optimization problem constraints are designated as linking variables.

In the form given in Equation (Bi-level), the lower problem may have more than one optimal response to a certain selection of the system problem. This implies that the lower problem is a set-valued map, and that the system's objective function is one as well. To circumvent this issue, the bi-level programming literature typically constrains the lower problem's possible outputs to two options: either the best or the worst lower problem's solution with respect to the system's objective function is assumed. In other words, it is assumed that the lower problem acts either in a cooperative manner (as described in (Bi-level)) or in an aggressive manner ((Bi-level) if the system objective f is no longer minimized but maximized with respect to y). The resulting bi-level problem is designated as either an optimistic bi-level programming problem or a pessimistic bi-level programming problem, respectively. Another pessimistic version of the bi-level programming emerges when the system level constraints, g , are explicitly dependent on the lower optimization problem set of variables, y . These joint constraints (or coupling constraints) are considered to be part of the pessimistic paradigm, as they permit the lower problem's response to contravene the system's feasibility.

In order to limit the possibilities and to evacuate the most difficult cases, it is common to consider only the case where the lower optimization problem always yields a unique output and there are no joint constraints in the system optimization problem. This is all the more true for the targeted MDO problems, where the uniqueness of the minimizer is often considered, and where all components are designed to act cooperatively toward minimizing a common objective. Note that the uniqueness of the response of the lower problem makes the optimistic and pessimistic paradigms equivalent, so that the variable y can be retrieved from the system's perspective. This leads to the optimistic bi-level optimization problem, which can be rewritten as follows:

$$\begin{aligned}
 & \min_{x \in \mathcal{X}} f(x, y^*(x)) \\
 & \text{s.t.} \quad g(x) \leq 0 \\
 & \text{with } y^*(x) = \arg \min_{y \in \mathcal{Y}} \{F(x, y) \text{ s.t. } G(x, y) \leq 0\}
 \end{aligned}
 \tag{Bi-level Opt}$$

Even in this commonly acknowledged form, the bi-level optimization problems are considered to be difficult to solve. Consequently, a significant portion of the classical literature considers bi-level problems that are mathematically well-behaved. This typically involves the assumptions that all the functions are linear, quadratic or convex, and that strong assumptions such as continuous differentiability and lower semi-continuity are made.

Indeed, the bi-level optimization problem has been demonstrated to be NP-hard even in the case where all the functions involved are linear [26]. This is evidenced by [153], who showed that even verification of local optimality for a feasible solution is in general NP-hard. Furthermore, most used constraint qualifications are violated in every feasible point [90]. Finally, the bi-level optimization problems are typically nonconvex and non-differentiable optimization problems, even when all the functions satisfy these assumptions. This renders the computation of an optimal solution a challenging task.

When considering the characterization of optimal solutions and the convergence of nested optimization problems, the bi-level optimization problems are often reformulated as a single

optimization problem [40]. This approach permits the application of a substantial body of literature in numerical optimization, where there is a unique optimization problem.

Enforcing lower level optimality as a constraint

If the lower optimization problem is assumed to be convex for all feasible system choices $x \in \mathcal{X}$, it is possible to replace the lower optimization problem optimality constraint with its generalized equation [40]:

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \quad & f(x, y) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & 0 \in \nabla_y F(x, y) + N_{K(x)}(y) \end{aligned} \tag{\mathcal{P}_{var}}$$

where $N_{K(x)}(y)$ denotes the normal cone in the sense of convex analysis. This variational reformulation has been studied by [41] and shown to be equivalent to (**Bi-level Opt**).

Given that the approximation of $N_{K(x)}(y)$ is inherently complex, the conventional approach entails substituting the generalized constraint equation with its first-order KKT optimality conditions:

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathcal{Y}, u \in \mathbb{R}^{n_{cl}}} \quad & f(x, y) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & \nabla_y F(x, y) + u^T \nabla_y G(x, y) = 0 \\ & G(x, y) \leq 0, \quad u \geq 0, \quad u^T G(x, y) = 0 \end{aligned} \tag{\mathcal{P}_{kkt}}$$

which introduces u as the Lagrange multipliers of the lower problem's constraints G .

The obtained single optimization problem is an instance of Mathematical programming with equilibrium constraints (or MPEC), which are non-smooth, non-convex optimization problems. These problems violate common constraint qualifications such as the Mangasarian-Fromowitz constraint qualification (MFCQ) at any feasible point of the problem [90] and that is an important (and minimalist) assumption for NLP. The failure of the MFCQ is straightforward and can be found in [90, 40]:

Proposition 3.2.1. Consider an MPEC in the general form:

$$\begin{aligned} \min_z \quad & f(z) \\ \text{s.t.} \quad & g(z) \leq 0, \quad h(z) = 0, \\ & 0 \leq G(z) \perp H(z) \leq 0 \end{aligned} \tag{MPEC}$$

with $0 \leq G(z) \perp H(z) \leq 0$ being a rewriting of the equilibrium constraints $G(z) \geq 0, \quad H(z) \geq 0, \quad G(z)^\top H(z) = 0$.

Then MFCQ does not hold at any feasible points.

Proof. Let \bar{z} be a feasible point of (MPEC). Suppose $I := \{1 \leq i \leq n : G_i(\bar{z}) = 0\} \neq \emptyset$. The complementary slackness condition implies that $H_i(z) = 0 \quad \forall i \in I^c := \{1 \leq i \leq n : G_i(\bar{z}) \neq 0\}$. Let $\phi(z) := G(z)^\top H(z)$.

Suppose that there exists $d \in \mathbb{R}^n$ such that:

$$\nabla G_i(\bar{z})^\top d = \sum_{j=1}^n d_j \nabla_j G_i(\bar{z}) < 0, \quad \forall i \in I$$

and

$$\nabla H_i(\bar{z})^\top d = \sum_{j=1}^n d_j \nabla_j H_i(\bar{z}) < 0, \quad \forall i \in I^c$$

Hence:

$$\nabla \phi(\bar{z})^\top d = \sum_{i \in I} H_i(\bar{z}) \sum_{j=1}^n d_j \nabla_j G_i(\bar{z}) + \sum_{i \in I^c} G_i(\bar{z}) \sum_{j=1}^n d_j \nabla_j H_i(\bar{z}) < 0$$

which implies that the **MFCQ** does not hold. \square

Failure of the **MFCQ** implies, among other things, that the Lagrange multipliers set is unbounded, that the constraint gradients are linearly dependent. (since $LICQ(\bar{x})$ implies $MFCQ(\bar{x})$), and that the central path does not exist.

Moreover, [39] shows that the relation between **Bi-level Opt** and \mathcal{P}_{kkt} is not straightforward and requires verification on the set of admissible Lagrange multipliers. For the enunciation of the theorem, let us first consider the following constraints qualification for bi-level programming with a convex lower optimization problem:

Definition 3.2.1 (SCQ). A design point \bar{x} of (**Bi-level Opt**) is said to satisfies the Slater Constraint Qualification (SCQ) if and only if

$$\{y \in \mathcal{Y} | G(\bar{x}, y) < 0\} \neq \emptyset, \quad (3.2.2)$$

i.e. (**Bi-level Opt**)'s lower optimization problem, parameterized by \bar{x} , admits at least on strictly feasible point y .

Let $\Lambda(\bar{x}, \bar{y})$ be the set of Lagrange multipliers for the lower optimization problem:

$$\Lambda(\bar{x}, \bar{y}) = \left\{ u \left| \begin{array}{l} \nabla_y F(\bar{x}, \bar{y}) + u^\top \nabla_y G(\bar{x}, \bar{y}) = 0 \\ u \geq 0, \quad G(\bar{x}, \bar{y}) \leq 0, \quad u^\top G(\bar{x}, \bar{y}) = 0 \end{array} \right. \right\} \quad (3.2.3)$$

Theorem 3.2.1 ([39]). Assume that the (**Bi-level Opt**)'s lower optimization problem is convex.

Let (\bar{x}, \bar{y}) be a global (resp. local) optimal solution of (**Bi-level Opt**) and assume that **SCQ** is satisfied at \bar{x} . Then, for each $\bar{u} \in \Lambda(\bar{x}, \bar{y})$, the point $(\bar{x}, \bar{y}, \bar{u})$ is a global (resp. local) optimal solution of problem (\mathcal{P}_{kkt}) . Conversely, let **SCQ** (3.2.1) hold at all $x \in X$ (resp. at \bar{x}).

Further assume that $(\bar{x}, \bar{y}, \bar{u})$ is a global optimal solution (resp. local optimal solution for all $\bar{u} \in \Lambda(\bar{x}, \bar{y})$) of problem (\mathcal{P}_{kkt}) . Then, (\bar{x}, \bar{y}) is a global (resp. local) optimal solution of (**Bi-level Opt**)

It follows that under the right assumptions the global solution of (**Bi-level Opt**) and (\mathcal{P}_{kkt}) are equal. However, if a local solution of (**Bi-level Opt**) can be readily identified as a local solution of (\mathcal{P}_{kkt}) , the converse is not necessarily true. This is because the triplets (\bar{x}, \bar{y}, u) must be local solutions of (\mathcal{P}_{kkt}) for all $u \in \Lambda(\bar{x}, \bar{y})$. This last condition

is not only difficult to verify in practice, as it results in a combinatorial problem best handled by enumeration algorithms such as branch-and-bound methods [30], it is also necessary [39]. In more precise terms, the primal KKT reformulation, which incorporates the introduction of the multipliers u , has the effect of breaking the direct equivalence between the local solution of (**Bi-level Opt**) and (\mathcal{P}_{var}).

Another popular approach to reformulating nested optimization problems into a single one is to introduce the optimal value function of the lower optimization problem.

Let $V : \mathcal{X} \rightarrow \bar{\mathbb{R}}$ be the optimal value function of the lower optimization problem:

$$V(x) := \inf_y \{F(x, y) \mid G(x, y) \leq 0\} \quad (3.2.4)$$

with $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ the extended real line.

The bi-level problem (**Bi-level Opt**), can be reformulated into the value function problem [114]:

$$\begin{aligned} \min_{x,y} \quad & f(x, y) \\ \text{s.t.} \quad & F(x, y) - V(x) \leq 0 \\ & G(x, y) \leq 0 \\ & g(x) \leq 0 \end{aligned} \quad (\mathcal{P}_{vf})$$

For all possible $x \in \mathcal{X}$, assume that $V(x)$ is well defined. This implies that the lower optimization problem always admits at least one feasible point, regardless of the system's choice. Under this unique assumption, it can be demonstrated that (\mathcal{P}_{vf}) is strictly equivalent to (**Bi-level Opt**) [114].

Nevertheless, even with strong assumptions, V is not differentiable in general. The classical constraint qualifications such as the **MFCQ**, still fail at every feasible point [42]. Furthermore, evaluating V implies solving the lower optimization problem entirely.

A less frequently considered reformulation is the implicit reformulation. This is achieved by applying the assumptions needed by the implicit function theorem (cf Theorem 2.2.5) to hold to the optimality conditions. This allows the lower optimization problem's design variables to be rewritten as an implicit function of the system variables optimization problem [37].

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x, y^*(x)) \\ \text{s.t.} \quad & g(x) \leq 0 \\ \text{with} \quad & y^*(x) = \arg \min_{y \in \mathcal{Y}} \{F(x, y) \mid G(x, y) \leq 0\} \end{aligned} \quad (\text{Bi-Level implicit})$$

This rewriting, which is based on the field of parametric optimization, facilitates the convergence analysis of the (**Bi-level Opt**) by replacing the solution of the lower optimization problem entirely by a (at least) continuous function. Both (**Bi-level Opt**) and (**Bi-Level implicit**) are in fact identical, as shown by their definitions, but the latter is generally assumed to make more regularity assumptions on its mapping function y^* , typically provided by the **Implicit Function Theorem**. However, this reformulation is often overlooked as the assumptions needed on the problem's structure for the implicit

theorem to hold are very strong, especially for non-linear constrained optimization. In addition to the objective and constraints of the lower optimization problem being at least C^2 functions, for each value of x , the unique solution $y^*(x)$ must be continuously differentiable in a neighborhood of x , meaning that, in the said neighborhood, $y^*(x)$ must satisfy the LICQ, the strict complementarity, and the second-order KKT conditions [47]. In particular, this implies that in the considered neighborhood, the set of active constraints remains unchanged, which is also a very strong assumption, as optimality often occurs at constraints intersections. In practice, since we aim at using a derivative-free optimization algorithm at the system-level, it is reasonable to expect that with few and punctual changes to the active constraint set, the process can still reach a satisfactory solution.

Among the advantages of this formalism are its ease of implementation and the fact that the resolution of the lower optimization problem is fully embedded within a mathematical function from the system's perspective. Consequently, the burden of verifying the optimality of the lower level does not fall on the system optimizer. In contrast, other methodologies typically add constraints to check the optimality of the lower level, leading to the failure of common constraints qualification as seen before.

For a comprehensive examination of the one-level reformulation, a characterization of their respective solutions throughout variational analysis, and a delineation of the link between their respective solutions, see the manuscript from [162].

For a comprehensive bibliography encompassing all facets of bi-level optimization problems, see [38]. Another bibliography can be found in [154]. See [40] for a global perspective on the bi-level programming and [88] for a survey on multilevel decision-making. [30] enumerates classical applications and approaches to solve bi-level problems.

Choice of the bi-level formalization

As just shown, the reformulation into a single optimization problem and the characterization of the solutions are not straightforward. The initial three reformulations failed the classical constraints qualifications and necessitate the use of variational analysis and generalized differentiation for further characterization of the solutions. Furthermore, due to the inherent difficulty of manipulating the mathematical objects involved in all these optimality conditions, they have few practical uses [30]. Conversely, in the case of implicit reformulation, the underlying assumptions are quite strong and may not be true in many applications. Furthermore, these assumptions are typically presumed without further examination, as they are not susceptible to empirical verification. However, using a derivative-free optimization algorithm at the system-level can help overcome these difficulties, especially if these discontinuities are not very frequent. In addition to these considerations, it illustrates that single-level reformulation must be approached with caution. Even with strong assumptions, the characterization of the solutions is complex, and standard NLP results do not hold, eventually the equivalence of the solutions with the original bi-level optimization problem can be lost.

In every case, the use of a gradient-based optimizer at the system level is complex. In the majority of instances, the problem is non-smooth, necessitating the use of costly directional derivatives to approximate the sensitivities of the lower-problem solution mapping. When considering first-order conditions for optimality in the case where

derivatives are available or can be approximated, it is necessary to verify the validity of the solution. As highlighted before, this is an NP-hard problem [153] which present combinatorial issues [39, 40]. Considering that for the targeted real-life applications, the derivatives along the upper-level variables may be either missing or extremely costly to obtain or approximate, this further push toward the avoidance of KKT conditions and most variational analysis tools for the upper-level.

For this reason, the initial three one-level reformulations of the optimistic bi-level will not be further investigated in this manuscript, as the implicit reformulation seems to be the best methodology when considering a derivative-free algorithm at the system level, which is in accordance with the choice of the BL-IRT architecture. The implicit reformulation alleviates such issues, provided that strong assumptions are made about the lower optimization problem. These assumptions serve as a baseline for a considered perfect behavior. In practice, experiments on several applications have shown that strong assumptions such as strong convexity are not mandatory [146] for these algorithms to converge to a satisfactory solution.

3.3 From MDF to a bi-level decomposition

Consider the MDF reformulation of a generic MDO problem:

$$\begin{aligned} \min_{(x_0, x) \in \mathcal{X}_0 \times \mathcal{X}} \quad & f(x_0, x, \Psi(x_0, x)) \\ \text{subject to} \quad & G(x_0, x, \Psi(x_0, x)) \leq 0 \end{aligned} \quad (\mathcal{P}_{MDF})$$

where $\mathcal{X}_0 \subset \mathbb{R}^{n_0}$ and $\mathcal{X} \subset \mathbb{R}^n$ denote the design spaces of the shared and local variables respectively, $f : \mathcal{X}_0 \times \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}$ the objective, and $G : \mathcal{X}_0 \times \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_c}$ the constraints. The function $\Psi : \mathbb{R}^{n_0+n} \rightarrow \mathbb{R}^m$ denotes the MDA function defined by the implicit function theorem (cf. Theorem 2.2.5) which computes the couplings at equilibrium for some design vectors x_0 and x .

As the (*MDF*) architecture is the most widely used architectures in the MDO community, it will be our reference for elaborating a multidisciplinary feasible strategy based bi-level architecture. In particular, all assumptions typically made to ensure the validity and convergence of the (*MDF*) architecture for gradient-based optimizers are assumed to be true. In particular the MDA Ψ is assumed to be well-defined, so that the existence and uniqueness of fixed point for the MDA is true for every design vector x_0, x . Apart from considering the MDA iteration map as a contraction map for the Banach fixed-point theorem [10] to hold, all disciplines are considered as C^2 bounded functions, so that the implicit function theorem (cf. Theorem 2.2.5) holds at each design point (x, x_0) to eliminate the coupling constraints from the optimization problem. Note that these assumptions ensure that the objective and constraints are also at least C^2 bounded functions.

This section is dedicated to the presentation of the bi-level decomposition. Starting from the MDF architecture's optimization problem (\mathcal{P}_{MDF}), a transformation is carried out to obtain the same decomposition as in the IRT-BL architecture. This decomposition further assumes that the local variables x are implicitly defined by the value x_0 through the resolution of the lower optimization problem. To ensure that the said mapping is at

least continuously differentiable, sufficient regularity conditions are given on the lower optimization problem.

3.3.1 An equivalent bi-level decomposition

In order to exploit the separation of the design variables between the shared variables x_0 and the local ones $x = (x_1, \dots, x_p)$, a straightforward bi-level reformulation of the previous optimization problem is proposed:

$$\begin{aligned} & \min_{x_0 \in \mathcal{X}_0} f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \\ \text{s.t.} \quad & g^{up}(x_0) \leq 0 \\ & x^*(x_0) = \arg \min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\} \end{aligned} \quad (\mathcal{P}_{Bi-level})$$

where:

- the function g^{up} defined on \mathbb{R}^{n_0} , gathers the constraints from $G(x_0, x, y)$ that depend only on the shared variables x_0 . They are handled by the system level optimization problem.
- the function g defined on $\mathbb{R}^{n_0} \times \mathbb{R}^n \times \mathbb{R}^m$, gathers the constraints from G that depend on the shared variables x_0 , the coupling variables y and at least one block $x_i (i \in \llbracket 1, p \rrbracket)$ of the local design variable x . They are handled by the lower optimization problem:

$$\min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\}. \quad (\mathcal{P}_{low}(x_0))$$

By definition, by rearranging the components of G , we have:

$$G(x_0, x, y) = [g^{up}(x_0), g(x_0, x, y)], \quad (3.3.1)$$

Schematically, at the higher-level the bi-level process is as follow:

Algorithm 1: Bi-level algorithm

Data: x_0^0, x^0, y^0

Result: x_0^*, x^*, y^*

$k = 0$;

while x_0^k not solution of $\mathcal{P}_{Bilevel}$ **do**

choose next system iterate: x_0^{k+1} ;

Solve $\mathcal{P}_{low}(x_0^{k+1})$: $x^{k+1} = \arg \min_x \left\{ \begin{array}{l} f(x_0^{k+1}, x, \Psi(x_0^{k+1}, x)) \\ \text{s.t. } g(x_0^{k+1}, x, \Psi(x_0^{k+1}, x)) \leq 0 \end{array} \right\}$;

Retrieve couplings value: $y^{k+1} = \Psi(x_0^{k+1}, x^{k+1})$;

Compute the system objective: $f^{k+1} = f(x_0^{k+1}, x^{k+1}, y^{k+1})$;

Compute the system constraints: $g^{up\ k+1} = g^{up}(x_0^{k+1})$;

$k = k + 1$;

end

$x_0^*, x^*, y^* = x_0^k, x^k, y^k$;

This reformulation addresses the optimization problem by decomposing the optimization of the design vector (x_0, x) into two optimization processes. The first process is a system level optimization problem that optimizes with respect to the shared variable x_0 under constraints that do not depend on x . The second process is a lower optimization problem that minimizes the same objective function according to the local design variable x for a particular value of x_0 under constraints that depend on at least one block of local constraints.

This bi-level reformulation is driven by the practical observation that shared variables x_0 are expected to be low-dimensional and thus manageable by a gradient-free algorithm, while local variables ($x_i \ i \in \llbracket 1, p \rrbracket$) and couplings ($\Psi(x_0, x)$) are expected to be much more numerous, justifying the combination of a gradient-based optimizer and fixed-point algorithm to handle them effectively. Furthermore, coupled derivatives with respect to the shared variables x_0 ($\frac{\partial \Psi}{\partial x_0}$) may not be available, so that gradient-based optimization is only possible if the shared variables are eliminated.

One of the main assumptions underlying $(\mathcal{P}_{Bi-level})$ is that the mapping $x^*(x_0)$ is, in fact, a function. Consequently, it is postulated that for each value of x_0 , there exists a unique minimizer $x^*(x_0)$ for the lower optimization problem. It is important to note that for this assumption to be valid in practice, it is essential to solve $(P_{low}(x_0))$ as precisely as possible. Without further assumptions, we prove that $(\mathcal{P}_{Bi-level})$ and (\mathcal{P}_{MDF}) are equivalent.

Proposition 3.3.1. Assume that the lower optimization problem $(P_{low}(x_0))$ admits a unique solution denoted by $x^*(x_0)$ for all the possible values of $x_0 \in \mathcal{X}_0$. Then the two architectures $(\mathcal{P}_{Bi-level})$ and (\mathcal{P}_{MDF}) are said equivalent in the sense that:

- If x_0 is a solution of the $(\mathcal{P}_{Bi-level})$ problem, then $(x_0, x^*(x_0))$ is a solution of the (\mathcal{P}_{MDF}) problem.
- If (x_0, x) is a solution of the (\mathcal{P}_{MDF}) problem then x_0 is a solution of the $(\mathcal{P}_{Bi-level})$ problem and $x = x^*(x_0)$.

Proof Let x_0 be a solution of the $(\mathcal{P}_{Bi-level})$ optimization problem and let

$$x^*(x_0) = \arg \min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\} \quad (3.3.2)$$

be the unique solution of the (\mathcal{P}_{MDF}) optimization problem. Let us prove that the point $(x_0, x^*(x_0))$ is solution of the (\mathcal{P}_{MDF}) problem.

By definition of x_0 and $x^*(x_0)$, we easily check that $(x_0, x^*(x_0))$ is an admissible point for the (\mathcal{P}_{MDF}) optimization problem. Let $(x'_0, x) \in \mathcal{X}_0 \times \mathcal{X}$ be any admissible point for the (\mathcal{P}_{MDF}) problem i.e. such that: $G(x'_0, x, \Psi(x'_0, x)) \leq 0$, which means that:

$$g^{up}(x'_0) \leq 0, \quad g(x'_0, x, \Psi(x'_0, x)) \leq 0. \quad (3.3.3)$$

In particular observe that the point x is admissible for the lower level optimization problem computed at x'_0 , hence:

$$f(x'_0, x, \Psi(x'_0, x)) \geq f(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))), \quad (3.3.4)$$

Remembering that x_0 is an optimal solution of the problem $(\mathcal{P}_{Bi-level})$, we finally get:

$$f(x'_0, x, \Psi(x'_0, x)) \geq f(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))) \geq f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))), \quad (3.3.5)$$

proving that $(x_0, x^*(x_0))$ is an optimal solution of the (\mathcal{P}_{MDF}) problem.

Conversely, let (x_0, x) be an optimal solution of the (\mathcal{P}_{MDF}) problem. Let us verify that x_0 is an optimal solution of the $(\mathcal{P}_{Bi-level})$ problem. Let x'_0 be an admissible point for the problem $(\mathcal{P}_{Bi-level})$ i.e. such that: $g^{up}(x'_0) \leq 0$. By definition of $x^*(x'_0)$, we have:

$$g(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))) \leq 0 \quad (3.3.6)$$

so that the point $(x'_0, x^*(x'_0))$ is an admissible point for the (\mathcal{P}_{MDF}) problem. Remembering that (x_0, x) is an optimal solution of the (\mathcal{P}_{MDF}) problem and then using the definition of $x^*(x_0)$, we finally get:

$$f(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))) \geq f(x_0, x, \Psi(x_0, x)) \geq f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \quad (3.3.7)$$

proving that x_0 is an optimal solution of the $(\mathcal{P}_{Bi-level})$ optimization problem as expected. \square

In the event that g^{up} also depends on the coupling variable y , the equivalence property is undermined. More precisely, if (x_0, x) is a solution of the (\mathcal{P}_{MDF}) problem, it cannot be guaranteed that $x = x^*(x_0)$ and that $g^{up}(x_0, \Psi(x_0, x^*(x_0))) \leq 0$. Consequently, x_0 is not guaranteed to be a feasible point for $(\mathcal{P}_{Bi-level})$. However it should be noted that the bi-level architecture remains relevant as any solution of the bi-level problem remains a solution of the (\mathcal{P}_{MDF}) problem.

Choice have been made to exclude this potential dependency of the system level constraints on the coupling vector to ensure Proposition 3.3.1 and as real-life applications generally fulfill this requirement (See for instance Section 3.6). Otherwise, equivalency is lost but as a solution of $(\mathcal{P}_{Bi-level})$ is still guaranteed to be a solution for (\mathcal{P}_{MDF}) , this bi-level optimization may still be relevant to find a (\mathcal{P}_{MDF}) solution.

3.3.2 Regularity of the system level functions

As previously stated, the $(\mathcal{P}_{Bi-level})$ reformulation of the (\mathcal{P}_{MDF}) optimization problem can be used to find the solution of the original problem. However, this reformulation may have altered the mathematical properties of the problem, deteriorating the convergence properties of common optimization algorithms when applied directly to $(\mathcal{P}_{Bi-level})$. Among the most crucial assumptions in non-linear optimization is the continuity of the functions of interest. This is highly desirable, and in many cases, it is even necessary for the convergence of most algorithms. Similarly, the assurance that these same functions are also continuously differentiable is a key factor, even for gradient-free optimizers. For these reasons, the following results aim to provide sufficient conditions to ensure that all functions of interest are, from the perspective of the system optimization problem, continuously differentiable with regard to the shared variables x_0 .

The main difference between the (\mathcal{P}_{MDF}) and the $(\mathcal{P}_{Bi-level})$ architecture lies in the mapping $x^* : x_0 \mapsto x^*(x_0)$ of the optimal local design variable according to each shared design variable x_0 . For the system optimizer, the objective function $(x_0, x) \mapsto f(x_0, x, \Psi(x_0, x))$ is recast as $x_0 \mapsto F(x_0) = f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0)))$, which is no longer guaranteed to be C^1 with respect to the shared variables x_0 . The introduction of x^* may therefore result in a deterioration of the convergence properties for the (\mathcal{P}_{MDF}) architecture, particularly when regularity conditions are required on the functions of interest. The following proposition

establishes the continuity and the differentiability with respect to the shared variables x_0 by applying Fiacco's theorem [47, Theorem 2.1], [143, Theorem 6.7] to the lower optimization problem $P_{low}(x_0)$ defined by $(P_{low}(x_0))$.

Proposition 3.3.2. Let $f : \mathbb{R}^{n_0} \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ and $G : \mathbb{R}^{n_0} \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^{n_c}$ be at least C^2 functions. Let $\bar{x}_0 \in \mathbb{R}^{n_0}$ be a particular value of the shared variables x_0 such that the constraint qualification holds at \bar{x}_0 for $P_{low}(\bar{x}_0)$, in the sense that the gradients of the active constraints at \bar{x}_0 are linearly independent.

Let $x^*(\bar{x}_0)$ be a Karush-Kuhn-Tucker (KKT) point of the problem $P_{low}(\bar{x}_0)$ and $\lambda^*(\bar{x}_0)$ the associated Lagrange multiplier. Assume that strict complementary slackness holds (component-wise):

$$\lambda_i^*(\bar{x}_0) > 0 \text{ when } g_i(\bar{x}_0, x^*(\bar{x}_0), \Psi(\bar{x}_0, x^*(\bar{x}_0))) = 0,$$

and that the second order sufficient KKT conditions holds at $x^*(\bar{x}_0)$ i.e. that for every direction $d \neq 0$ satisfying

$$\nabla g_i(\bar{x}_0, x^*(\bar{x}_0), \Psi(\bar{x}_0, x^*(\bar{x}_0)))^\top d = 0 \quad \forall i \text{ s.t. } g_i(\bar{x}_0, x^*(\bar{x}_0), \Psi(\bar{x}_0, x^*(\bar{x}_0))) = 0,$$

we have: $d^\top \nabla^2 \mathcal{L}(x^*(\bar{x}_0), \lambda^*(\bar{x}_0)) d > 0$ where \mathcal{L} denotes the Lagrangian associated to the problem $P_{low}(\bar{x}_0)$. Then there exists a neighborhood $\bar{\mathcal{V}}_0$ of \bar{x}_0 such that:

1. $x^*(\bar{x}_0)$ is a local isolated minimum point of $P_{low}(\bar{x}_0)$ and the associated multiplier, denoted by $\lambda^*(\bar{x}_0)$, is uniquely defined.
2. For all shared variables x_0 in $\bar{\mathcal{V}}_0$, there exist unique continuously differentiable functions $x_0 \mapsto (x^*(x_0), \lambda^*(x_0))$ satisfying the second order sufficient conditions for a local minimum of $P_{low}(x_0)$, and $x^*(x_0)$ is a unique local minimizer of $P_{low}(x_0)$ with associated unique Lagrange multiplier $\lambda^*(x_0)$.
3. Strict complementarity and linear independence of the binding constraint gradients hold at $x^*(x_0)$ for x_0 in $\bar{\mathcal{V}}_0$.

Proposition 3.3.2 can be viewed as a direct application of the implicit function theorem (cf. Theorem 2.2.5) on the first order KKT conditions (cf. Definition 2.2.4) of $(P_{low}(x_0))$, a parametrized (in x_0) and constrained optimization problem. It should be noted that although Fiacco's theorem is a strong result, it implies the verification of assumptions that are difficult to verify and obtain. In particular, the existence of continuously differentiable functions (x^*, λ^*) implies that, in the considered neighborhood, the set of active constraints remains unchanged. Consequently, in the vicinity of the solution point (x_0^*, x^*) , the inequality-constrained optimization problems $(P_{low}(x_0))$ need to be reducible to an equality-constrained one, where only active inequality constraints are considered. In practice, however, it is expected that most derivative-free algorithms will be able to handle a few and punctual discontinuities of the functions whenever the set of active constraints is changed by a system design x_0 update. Considering this set of hypotheses, in the aforementioned vicinity, the strict complementarity constraint of the first order KKT conditions no longer needs to be verified, which will be useful when we come to study the resolution of the lower-problem (cf. Section 3.4.2).

From this point forward, it will be assumed that this result is valid for any value of the shared variables x_0 . In other words, the bi-level reformulation $(\mathcal{P}_{Bi-level})$ can be reduced to a generic non-linear optimization problem in which all the functions of interest are

continuously differentiable, a particularly well-known field. The presented bi-level problem can be directly utilized to solve the original MDF problem, provided two optimization algorithms are available to solve the system optimization problem and the lower optimization problem, respectively.

However, reconsidering the initial problem, namely high-dimensional applications where numerous local and coupling variables are present and where the fully coupled derivatives are inaccessible, the MDF architecture cannot be directly utilized on the lower optimization problem ($P_{low}(x_0)$). Indeed, the problem typology indicates that a gradient-based approach is mandatory, but is impeded by the unavailability of the coupled gradients, which is a common situation due to the substantial implementation effort required [95].

As previously stated, the resolution of the lower optimization problem ($P_{low}(x_0)$) must be achieved with sufficient precision and in a reasonable amount of time for previous results to hold. Therefore there is a need for a dedicated strategy to obtain the solution of the lower-problem ($P_{low}(x_0)$). The subsequent section is thus dedicated to demonstrating that the lower optimization problem can be solved in practice without the fully coupled gradients. In order to achieve this objective, a block decomposition, analogous to the one utilized in the original IRT architecture [56], is proposed and an algorithmic solution, designated as the Block Coordinate Descent (**BCD**) algorithm, is presented.

3.4 A solution algorithm for the lower optimization problem

Let us now consider the lower-problem independently from the bi-level architecture:

$$\min_{x \in \mathcal{X}} f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0 \quad (\mathcal{P}_{low}(x_0))$$

for given shared variables x_0 . Assume that the design vector x can be partitioned in p non overlapping blocks denoted by $x_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, p$, such that omitting the dependence on the shared variables x_0 to lighten the notations, the problem ($\mathcal{P}_{low}(x_0)$) can be reformulated as:

$$\min_{x \in \prod_{i=1}^p \mathcal{X}_i} f(x, \Psi(x)) \text{ s.t. } g_i(x_i, \Psi(x)) \leq 0 \quad i = 1, \dots, p \quad (\mathcal{P}_{low})$$

where, reordering if necessary the components of x :

$$x = (x_1, \dots, x_p) \in \mathcal{X}_1 \times \dots \times \mathcal{X}_p \quad (3.4.1)$$

and \mathcal{X}_i denotes the box-constrained subspace of \mathbb{R}^{n_i} where lies the i^{th} block x_i . The functions $g_i : \mathbb{R}^{n_i} \times \mathbb{R}^m \mapsto \mathbb{R}^{n_{c_i}}$ define the constraints which directly depend on the block variable x_i and indirectly on the others block variables $x_{\neq i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p)$ through the coupling function Ψ .

This partitioning of the local variable and constraints allows for the definition of p optimization sub-problems that consider only a subset of the local variable and constraints. For $i \in \llbracket 1, p \rrbracket$, the i^{th} optimization sub-problem is defined by:

$$\begin{aligned} \min_{t \in \mathcal{X}_i} f(x_{1:i-1}, t, x_{i+1:p}, \Psi(x_{1:i-1}, t, x_{i+1:p})) \\ \text{s.t. } g_i(t, \Psi(x_{1:i-1}, t, x_{i+1:p})) \leq 0. \end{aligned} \quad (\mathbf{PB}_i(x_{\neq i}))$$

This block decomposition of (\mathcal{P}_{low}) serves several purposes. Primarily, this separation in blocks aligns with the industry standard, matching the decomposition in terms of disciplines. It allows for greater flexibility in the use of specific optimizers or algorithms to handle each subset of local variables, while dealing with much smaller optimization problems $(\mathbf{PB}_i(x_{\neq i}))$. Second, the full coupled derivatives $\frac{\partial \Psi}{\partial x}(x)$ may not be available, preventing the use of a gradient approach that considers all local variables at once. Although the full vector of coupled derivatives is not directly available, each block can have access to the partial derivatives of the coupled gradient $\left(\frac{\partial \Psi}{\partial x_i}(x)\right)_{i=1..p}$ along its own subset of variables. This is because, in practice, each block may have a disciplinary solver that already includes its own approximations to other couplings as well as the gradients associated with its block of local variables, thus allowing gradient-based approaches. An example of such block decomposition with well-chosen disciplinary solvers could be to solve a CFD aeroelasticity problem in a particular block with a low-fidelity structure model coupled to a structure block using a CSM solver that incorporates its own approximation of the aero couplings and their gradients. This type of approach is typically done in the doublet-lattice method [3] and further motivates the block decomposition.

Considering back the BL-IRT architecture [56], the proposed distributed-MDF bi-level architecture assumes that each block has a unique solution and that the objective function is minimized block by block, in parallel. The constraints g_i are handled by the block i that optimize f in x_i , as the aforementioned subset of variables is directly dependent on this subset. Considering an initial guess x is provided to solve (\mathcal{P}_{low}) , the solving algorithm proposed in [56] can be summarized by the following block optimizations:

$$x_i^*(x_{\neq i}) = \arg \min_{t \in \mathcal{X}_i} \{f(x_{1:i-1}, t, x_{i+1:p}, \Psi(x)) \text{ s.t. } g_i(t, \Psi(x)) \leq 0\} \quad i \in \llbracket 1, p \rrbracket. \quad (3.4.2)$$

All of these block optimizations are being computed in parallel. Two MDAs are employed to ensure that couplings are at equilibrium before ($y = \Psi(x)$) and after ($y = \Psi(x^*(x))$) the block updates. The overall process is illustrated for a generic two strongly coupled disciplines in Figure 3.3, with the associated disciplinary block optimizations gathered in Figure 3.4.

Since all blocks are solved in parallel, they do not exchange information about the solution found with the others until the next system iteration. Consequently, the only synchronization mechanism between the block optimizations are the two MDAs that compute the couplings and the system iterations that warm start each block optimization with the previous optimal value of each subset of local variables. This results in a significant dependence of x^* on the initial guess x , which is at the heart of BL-IRT's convergence issues. This dependency could result in disparate solutions for the same shared variables x_0 vector, depending on the initial guess x . Since the local variables x are hidden from the system optimizer, it introduces noise into the system objective and constraints with respect to x_0 . In addition to the high dependence of system functions on the path taken by the design variables, Fiacco's Theorem (Proposition 3.3.2) is unlikely to hold due to the noise introduced in the resolution of (\mathcal{P}_{low}) .

However, these observations are counterbalanced by the fact that, upon convergence of the system level optimizer, the two terms become equal, eliminating any induced error on the optimum. Additionally, in practice, the system-level optimization algorithm

does not propose the same value for the design variables twice, resulting in noise on the system functions instead of pure randomness. Some algorithms are more robust to noise than others. Therefore, in [56], the COBYLA algorithm was utilized [120]. This algorithm makes first-order approximations of the function and constraints, rendering it more tolerant to noisy functions in practice.

While solving (\mathcal{P}_{low}) using Equation (3.4.2) as a block update may be highly effective in terms of time consumption when convergence occurs at the system level, it is challenging to predict in advance the range of problems it can solve and/or if the discrepancy will circumvent a typical optimization algorithm to converge towards an acceptable solution. Referring this bi-level architecture as bi-level-IRT (BL-IRT), the following presents new architectures which aim to extend the range of problems this bi-level approach can solve by providing more stability on the resolution of the lower optimization problem, thus lowering the discrepancy and allowing for convergence proofs in certain cases. The choice fell on a Block Gauss-Seidel approach to solve the lower optimization problem (\mathcal{P}_{low}), namely a BCD-MDF algorithm, a Block Coordinate Descent algorithm using a MDF resolution at each step.

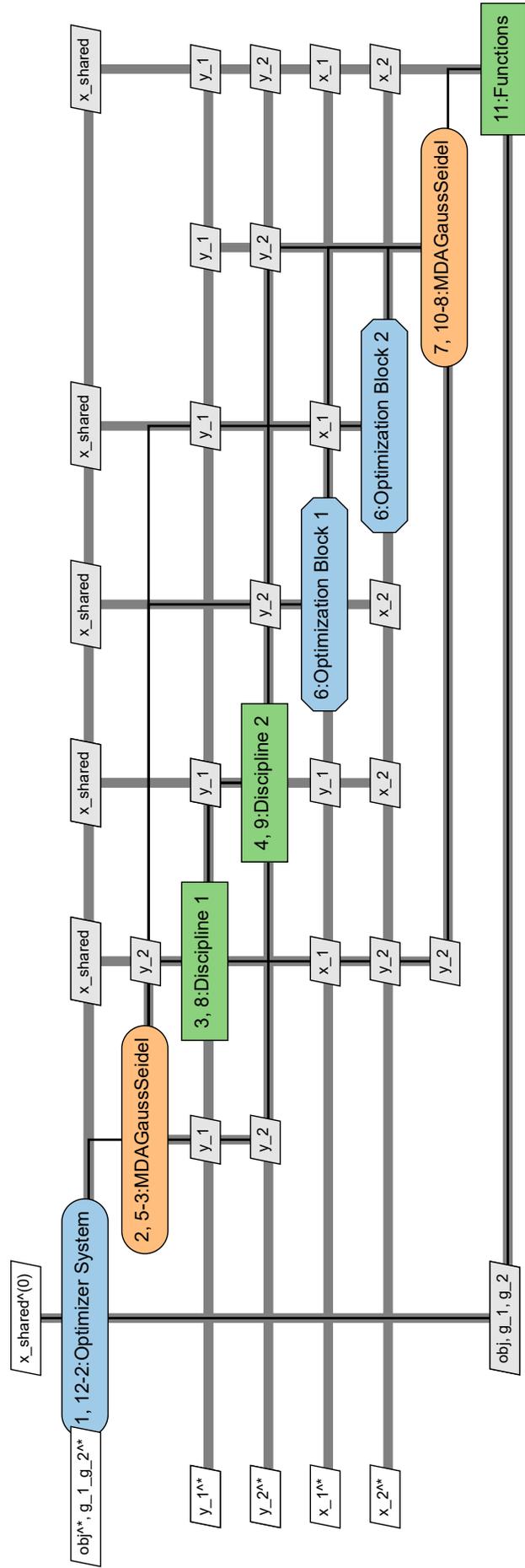


Figure 3.3: XDSM of the BL-IRT architecture

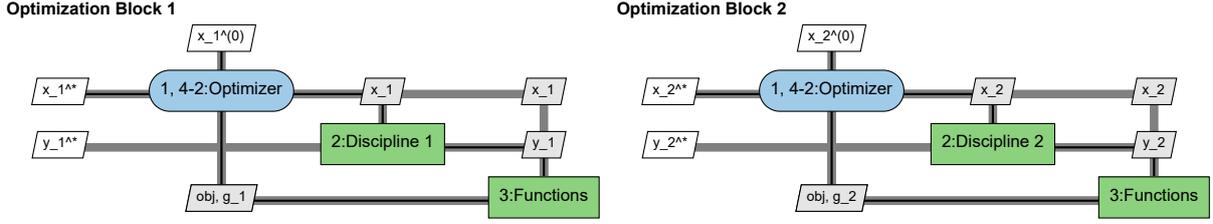


Figure 3.4: XDSM of disciplinary block optimizations executed in parallel within the BL-IRT architecture

3.4.1 The Block Coordinate Descent algorithm

Once the local variables x have been partitioned into p component vectors $x_i \in \mathbb{R}^{n_i}$, a widely used approach to solve (\mathcal{P}_{low}) is the block coordinate descent (**BCD**) method [16], which is also known as the Block Gauss-Seidel (**BGS**) method [64, 110, 111, 113] or alternating optimization (**AO**) method [17, 18]: at a given iteration k , for a particular value of i , the BCD algorithm computes a solution x_i^{k+1} of the following optimization problem:

$$\begin{aligned} \min_{t \in \mathcal{X}_i} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \\ \text{s.t. } g_i(t, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \leq 0 \end{aligned} \quad (\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$$

where: $x_{1:i-1}^{k+1} = (x_1^{k+1}, \dots, x_{i-1}^{k+1})$ and $x_{i+1:p}^k = (x_{i+1}^k, \dots, x_p^k)$. In other words, at each iteration of this iterative algorithm, the objective function is minimized with respect to a single block (or component), x_i , of variables while the rest of the blocks, $x_{\neq i}$, are held fixed. The iterative scheme loops over each block successively until convergence. The BCD method is assumed to be well-defined in the sense that every block (\mathbf{PB}_i) has a unique optimal solution. The Jacobi version of this algorithm, where each block optimization are solved in parallel instead of sequentially, can be seen as an in-between between the BL-IRT updates Equation (3.4.2) and the BCD algorithm but will not be considered. The main reason for this choice is that the method we're looking for should be as robust as possible for solving (\mathcal{P}_{low}) , and Jacobi-based methods are known to have poorer convergence properties than their Gauss-Seidel counterparts, as well as being less efficient at satisfying the constraints [15].

The proposed BCD algorithm for solving the lower optimization problem (\mathcal{P}_{low}) is described by Algorithm 2. It should be noted that Algorithm 2 will be referred to as the BCD-MDF algorithm, as each sub-optimization problem $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ is solved using a MDF architecture. The bi-level architecture using the BCD-MDF algorithm will be referred to as BL-BCD-MDF (Bi-level - Block Coordinate Descent - MDF architecture) and is illustrated in Figure 3.5 which depicts a generic XDSM diagram for two strongly coupled disciplines. The disciplinary optimizations are represented within Figure 3.6.

Algorithm 2: Block Coordinate Descent - (BCD-MDF) algorithm

Data: $x_0, x^0, \varepsilon_{var}, \varepsilon_{fun}$
Result: x^*
 $k, i = 0, 1;$
 $f^0 = f(x_0, x^0, \Psi(x_0, x^0));$
while $\frac{\|x^{k-1} - x^k\|}{\|x^k\|} > \varepsilon_{var}$ *OR* $\frac{\|f^{k-1} - f^k\|}{\|f^k\|} > \varepsilon_{fun}$ **do**
 $x^{k+1} = x^k;$
 for $i \in [1, p]$ **do**
 Optimize i^{th} Block: compute the solution x_i^{k+1} solution of:

$$\min_{t \in \mathcal{X}_i} \left\{ f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \text{ s.t. } g_i(t, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \leq 0 \right\};$$

 Update the i^{th} local data: $x^{k+1}[i] = x_i^{k+1};$
 end
 Compute the objective value: $f^{k+1} = f(x_0, x^{k+1}, \Psi(x_0, x^{k+1}));$
 $k = k + 1;$
end
 $x^* = x^k;$

Convergence issues for the block GS method have been widely explored under suitable convexity assumptions, both in the unconstrained and constrained case [16, 64, 91, 110, 111, 113, 149] and the reference therein. In [64], Grippo and Sciandrone consider the separable case i.e. where the feasible set is the Cartesian product of p closed convex sets. They prove that the block GS method is globally convergent for $p = 2$, and that for $p > 2$ convergence still holds provided that f is pseudoconvex, or componentwise strictly quasiconvex with respect to $p - 2$ components.

Definition 3.4.1 (Pseudoconvex function). A continuously differentiable function $f : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ defined on a (nonempty) convex open set \mathcal{X} is said to be pseudoconvex if and only if:

$$\forall (x, y) \in \mathcal{X}^2, \forall t \in (0, 1) : \quad \nabla f(x)^T (y - x) \geq 0 \implies f(y) \geq f(x) \quad (3.4.3)$$

Definition 3.4.2 (Quasiconvex function). A function $f : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ defined on a convex subset \mathcal{X} is said to be quasiconvex if and only if:

$$\forall (x, y) \in \mathcal{X}^2, \forall t \in (0, 1) : \quad f(tx + (1 - t)y) \leq \max \{f(x), f(y)\} \quad (3.4.4)$$

Both pseudoconvexity and quasiconvexity assumptions are relaxations of the classical convexity assumption:

$$\text{convexity} \implies \text{pseudoconvexity} \implies \text{quasiconvexity}. \quad (3.4.5)$$

Note that in the nonconvex case it is proved [64, 119], using counterexamples, that the GS method can cycle indefinitely without converging to a critical point if $p \geq 3$, even if the objective function is componentwise convex but not strictly quasiconvex with respect to each component.

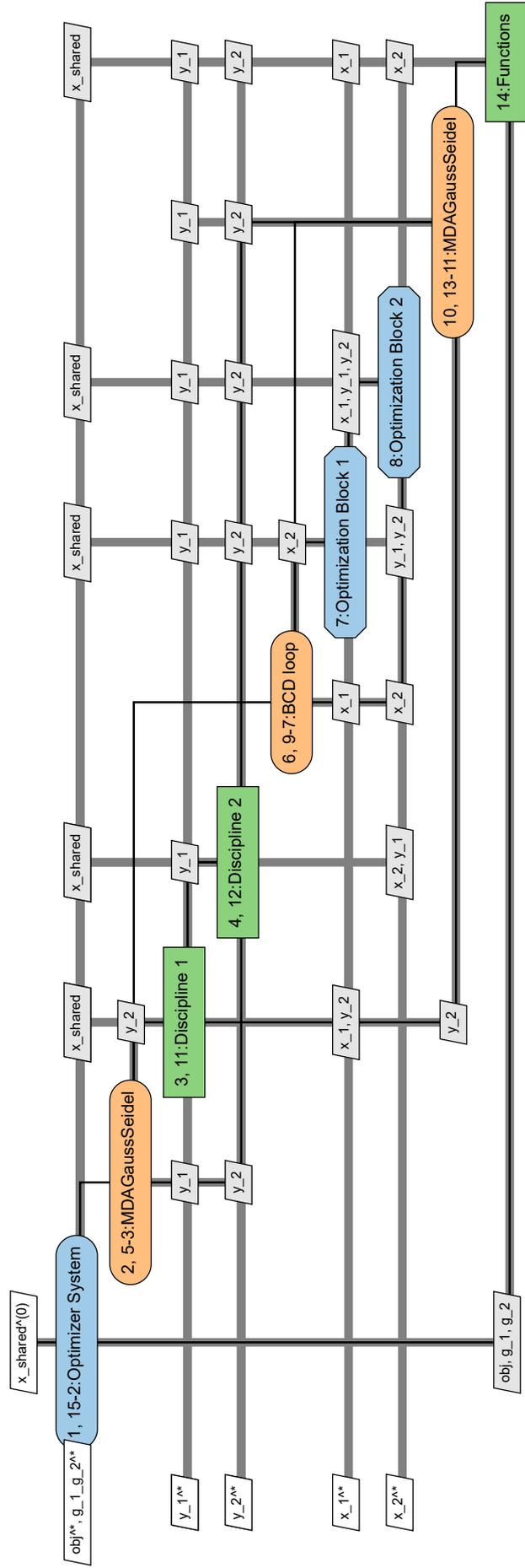


Figure 3.5: XDSM of the BL-BCD-MDF architecture

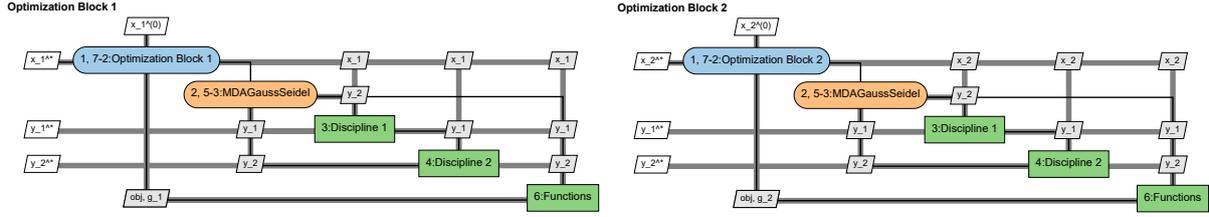


Figure 3.6: XDSM of the MDF block optimizations within the BCD loop of the BL-BCD-MDF architecture

However, special cases ensure convergence for non-convex and non-pseudo-convex objective functions [13], for example when the objective function is quadratic [91] or has a unique minimum in each coordinate block x_i [89]. This last assumption implies that there is a unique disciplinary solution for each shared variable x_0 , which is the most likely assumption in MDO practice.

In the non-separable case, where there is overlap between some of the variables between the blocks, as is the case for the (\mathcal{P}_{low}) problem due to the couplings introduced by the MDA, there are few results in the literature. One way of making the problem separable is to include the coupling constraints in the function to be minimized, using a well-chosen penalty: Tosserams et al. propose in [146] a method based on augmented Lagrangian relaxation and block coordinate descent. In [159], Xu and Yin suggest to incorporate the constraints into the objective using the indicator function, but this solution requires precise computation of the projection operator on the set of constraints, which can be as difficult as solving the original problem or simply not possible in practice.

Without further assumptions, we can easily check that, by construction, any fixed point of the BCD-MDF algorithm is necessarily a feasible point for the lower optimization problem (\mathcal{P}_{low}), but unfortunately not necessarily a solution of (\mathcal{P}_{low}) as explained in [19]:

Proposition 3.4.1. Let \bar{x} be a fixed point for the BCD-MDF algorithm which means that each block \bar{x}_i is solution of $\mathbf{PB}_i(\bar{x}_{\neq i})$. Then \bar{x} is a feasible point of \mathcal{P}_{low} .

In order to guarantee that the solutions of the lower optimization problem are indeed fixed points of the BCD-MDF algorithm (and vice versa), additional assumptions must be made regarding the block decomposition itself. First of all, it is assumed that every block optimization has a unique minimizer.

Definition 3.4.3 (Unique Block Minimizer). The optimization problem (\mathcal{P}_{low}) is said to verify the Unique Block Minimizer (UBM) property if for every block coordinate i and for every parameterization vector $x_{\neq i} = (x_{1:i-1}, \dots, x_{i+1:p})$, the optimization problem $\mathbf{PB}_i(x_{\neq i})$ has a unique minimizer and no local minimum point.

This is a reasonable assumption for MDO problems when distributed architectures are being considered and when convergence guarantees are sought. Indeed, it emphasizes that each disciplinary domain may be considered in an autonomous manner, finding their optimal design provided that any parameterization is utilized. This property can be regarded as a relaxation of the strict convexity of f and g_i over the entire domain on each block.

3.4.2 Convergence analysis of the BCD-MDF algorithm

In this section, we propose a series of assumptions designed to ensure the convergence of the BCD-MDF algorithm. It is important to note that, while a block decomposition may be employed, it does not guarantee that a fixed point in the BCD-MDF algorithm is, in fact, a solution to the lower optimization problem (\mathcal{P}_{low}). For the block decomposition to be meaningful, the blocks should depend primarily on the optimized block exerting only a minor influence on the constraints via the coupling variables. There are several potential ways of expressing this hypothesis.

First-Order Separability

A first approach is to assume that the constraints in each sub-blocks optimization are independent of the couplings in the neighborhood of the solution x^* of the lower optimization problem:

Definition 3.4.4 (First Order Separability). Let x^* be the MDF solution of (\mathcal{P}_{low}) at which the constraints are assumed to be qualified in the sense that the gradients of the active constraints at x^* are linearly independent. The optimization problem (\mathcal{P}_{low}) is said to have the First Order Separability (FOS) property at x^* if for every block coordinate i , and for every parameterization vector $x_{\neq i}$, the effects of the other blocks $k = 1, \dots, p$, $k \neq i$, upon the optimization of the i^{th} block are negligible in the following sense:

$$\forall i \in \llbracket 1, p \rrbracket, \forall k \neq i, \frac{\partial g_k}{\partial y}(x_k^*, \Psi(x^*)) \frac{\partial \Psi}{\partial x_i}(x^*) = 0. \quad (3.4.6)$$

In other words, each block x_i^* of a Karush-Kuhn-Tucker point x^* of the MDF problem (\mathcal{P}_{low}) is in turn a Karush-Kuhn-Tucker point of $\mathbf{PB}_i(x_{\neq i}^*)$. Under this assumption, exploiting the optimality conditions associated to each optimization problems at stake, we prove that the solution of the lower optimization problem is a fixed point of the BCD-MDF algorithm.

Proposition 3.4.2. Assume that the lower optimization problem (\mathcal{P}_{low}) has a unique solution x^* and satisfies the **Unique Block Minimizer** property. Assume also that (\mathcal{P}_{low}) has the **First Order Separability** property at x^* . Then x^* is a fixed point of the BCD-MDF algorithm i.e. for all $i = 1, \dots, p$,

$$x_i^* = \arg \min_{t \in \mathcal{X}_i} \left\{ f(x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \quad \text{s.t.} \quad g_i(t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \leq 0 \right\}. \quad (3.4.7)$$

Proof Let x^* be the unique solution of the lower optimization problem (\mathcal{P}_{low}). Let $i \in \{1, \dots, p\}$. Let us prove that x_i^* is solution of:

$$\min_{t \in \mathcal{X}_i} \left\{ f(x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \quad \text{s.t.} \quad g_i(t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \leq 0 \right\}. \quad (3.4.8)$$

Let \mathcal{L} be the Lagrangian function associated to the optimization problem (\mathcal{P}_{low}):

$$\mathcal{L}(x, \lambda) = f(x, \Psi(x)) + \sum_{i=1}^p \lambda_i^\top g_i(x_i, \Psi(x)) \quad (3.4.9)$$

where $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^{n_{c_1}} \times \dots \times \mathbb{R}^{n_{c_p}}$ denotes the vector of Lagrange multipliers, and \mathcal{L}_i the Lagrangian function associated to (3.4.8):

$$\mathcal{L}_i(t, \mu) = f(x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) + \mu^\top g_i(t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \quad (3.4.10)$$

where: $\mu \in \mathbb{R}^{n_i}$. Since x^* is the unique solution of (\mathcal{P}_{low}) whose constraints are assumed to be qualified at x^* , there exist Lagrange multipliers $\lambda \in \mathbb{R}^{n_{c_1}} \times \dots \times \mathbb{R}^{n_{c_p}}$ such that: $\nabla \mathcal{L}(x^*, \lambda) = 0$, and thus: $\nabla_{x_i} \mathcal{L}(x^*, \Psi(x^*), \lambda) = 0$. Applying now the FOS assumption at x^* , a straightforward computation provides:

$$\nabla_{x_i} \mathcal{L}_i(x_i^*, \lambda_i) = \nabla_{x_i} \mathcal{L}(x^*, \lambda) = 0. \quad (3.4.11)$$

and: $\nabla_{x_i}^2 \mathcal{L}_i(x_i^*, \lambda_i) = \nabla_{x_i}^2 \mathcal{L}(x^*, \lambda)$.

Furthermore, let $d \neq 0_{\mathbb{R}^n}$ be a feasible direction for (\mathcal{P}_{low}) at x^* , then for every constraints g_i and every component $j \in \llbracket 1, n_{c_i} \rrbracket$ such that $(g_i)_j(u_i, \Psi(x_{1:i-1}^*, x_i^*, x_{i+1:p}^*)) = 0$ we have:

$$\nabla_x (g_i)_j(x_i^*, \Psi(x^*))^\top d = 0 \quad (3.4.12)$$

As the FOS assumption holds at x^* , we also know that for all $k \neq i$, and every active component $j \in \llbracket 1, n_{c_k} \rrbracket$ of g_k at x^* :

$$\nabla_{x_i} (g_k)_j(x_k^*, \Psi(x^*)) = 0 \quad (3.4.13)$$

Hence d is a feasible direction of (\mathcal{P}_{low}) at x^* if and only if all its components d_i ($i \in \llbracket 1, p \rrbracket$) are feasible directions for their respective block $\mathbf{PB}_i(x_{\neq i}^*)$.

Putting all together and considering the case $i = i_0$, as x^* is the solution of (\mathcal{P}_{low}) , we have that for all feasible direction d_{i_0} of $\mathbf{PB}_{i_0}(x_{\neq i_0}^*)$,

$$d_{i_0}^\top \nabla_{x_{i_0}}^2 \mathcal{L}(x^*, \lambda) d_{i_0} = d_{i_0}^\top \nabla_{x_{i_0}}^2 \mathcal{L}_{i_0}(x_{i_0}^*, \lambda_{i_0}) d_{i_0} \geq 0. \quad (3.4.14)$$

Hence $x_{i_0}^*$ is a constrained solution of $\mathbf{PB}_{i_0}(x_{\neq i_0}^*)$ which admits two different solutions and contradicts the [Unique Block Minimizer](#) assumption. \square

The First Order Separability assumption is not always satisfied in practice, as it is a relatively strong assumption that is not easily verified beforehand. However, FOS acts as a sufficient assumption to ensure that the required solution is a fixed point for BCD-MDF. One might inquire whether a relaxed assumption exists, assuming that the effect of the blocks $x_{\neq i}$ on the constraints of a particular block i ($\mathbf{PB}_i(x_{\neq i})$) (throughout the MDA function Ψ) is negligible in comparison to the direct effect of the blocks' own disciplinary variables x_i around the solution x^* . In other words, consider cases where the first-order perturbation of block i resulting from the removal of $g_{\neq i}$, given by $\left\| \sum_{k \neq i} \lambda_k^* \nabla_{x_i} g_k(x_k^*, \Psi(x^*)) \right\|$ is either very small or negligible prior to the first-order lagrangian of the considered block $\|\nabla_{x_i} \mathcal{L}_i(x_i^*, \lambda_i)\|$. This is equivalent to the assumption that for each subproblem $\mathbf{PB}_i(x_{\neq i})$, near its solution and with respect to its block of variable x_i , the cross derivatives of the constraints of other blocks $g_{\neq i}$, through the coupling variables, are negligible in comparison to the total derivatives of f and g_i .

These relaxed assumptions, which are not considered here, may result in situations where each block does reach its corresponding block of the joint optima, i.e. $x_i^*(x_{\neq i}^*) = x_i^*$, the i^{th} block of the MDF optimal solution of $(\mathcal{P}_{low}(x_0))$, with a sufficient precision. This may be a promising direction for improvement of the following results.

Local convergence of the BCD-MDF algorithm

Let us now state one of the main theoretical results of this chapter, namely the local convergence of the BCD-MDF algorithm in the non-linear, coupled, and constrained case, extending the results given by *Ortega and al.* in [111, 113]:

Proposition 3.4.3. Let x^* be the unique solution of the lower level problem (\mathcal{P}_{low}) at which the constraints are assumed to be qualified in the gradients of the active constraints at x^* are linearly independent. Let λ^* be the associated Lagrange multiplier.

Assume that strict complementary slackness holds (component-wise) i.e. for any $i = 1, \dots, p$,

$$\lambda_i^* > 0 \text{ when } g_i(x_i^*, \Psi(x^*)) = 0, \quad (3.4.15)$$

and that the second order sufficient KKT conditions holds at x^* i.e. that for every direction $d \neq 0$ such that:

$$\nabla g_i(x^*, \Psi(x^*))^\top d = 0 \quad \text{for all } i \text{ such that } g_i(x_i^*, \Psi(x^*)) = 0, \quad (3.4.16)$$

we have: $d^\top \nabla^2 \mathcal{L}(x^*, \lambda^*) d > 0$ where \mathcal{L} denotes the Lagrangian associated to the problem \mathcal{P}_{low} .

Assume that the Unique Block Minimizer assumption and the First Order Separability at x^* holds. Then there exists a neighborhood $V(x^*, \lambda^*)$ of (x^*, λ^*) such that for every $(x^0, \lambda^0) \in V(x^*, \lambda^*)$, the sequence of iterates generated by the BCD-MDF algorithm converges q -linearly to (x^*, λ^*) .

In order to show the local convergence result, we first need to introduce the notion of eigenset, strongly quadratic form sign equivalent (SQFSE) matrices, and a key theorem, all of coming from [110].

Definition 3.4.5 (Eigenset). Let $A \in \mathcal{M}_n(\mathbb{R})$ for some integer $n > 0$. A subset E of \mathbb{C}^n is said to be an eigenset [110] of A if E consists of only eigenvectors (hence nonzero) of A , with at least one eigenvector corresponding to each distinct eigenvalue of A .

Definition 3.4.6 (SQFSE). Let S be an arbitrary non-empty subset of \mathbb{C}^n , the complex n -space. Then two $n \times n$ hermitian matrices P and Q are said to be strongly quadratic form sign equivalent (SQFSE) [110] on S if for all $x \neq 0$ in S the following condition is satisfied

$$(x^T P x)(x^T Q x) > 0 \quad (3.4.17)$$

The key theorem for the proof [110, Theorem 2.6.] takes the following form:

Theorem 3.4.1. Let $H \in \mathcal{M}_n(\mathbb{R})$, $n \in \mathbb{N}^*$, be an hermitian matrix and have a block splitting of the form $H = D - L - U$ with L being a block strictly lower-triangular matrix, U a block strictly upper-triangular matrix, and D a block diagonal matrix. Assume that $D - L$ is non-singular and defines $\nabla T = [D - L]^{-1}U$.

If $x^T D x \neq 0$ for all x on some eigenset E of ∇T then $\rho(\nabla T) < 1$ if and only if

$$(x^T D x)(x^T H x) > 0 \quad \text{for all } x \in E, \quad (3.4.18)$$

i.e. D and H are strongly quadratic form sign equivalent (SQFSE) on E .

Since the proof of Proposition 3.4.3 is a bit long, a sketch of the proof is given before the full proof, describing the main ideas and how to proceed.

Sketch of proof of Proposition 3.4.3

1. First, it is shown that the BCD-MDF algorithm, under the said assumptions, takes the form of an at least C^1 operator, denoted T , around the solution w^* . Hence there exists a neighborhood of w^* such that for all w^k :

$$w^{k+1} = T(w^k). \quad (3.4.19)$$

2. Then, by a first order Taylor extension of T around the solution, successive iterations are shown to be locally convergent towards w^* if and only if its gradient's at x^* is a contraction mapping. Hence the BCD-MDF algorithm converges q -linearly to (x^*, λ^*) if and only if

$$\rho(\nabla T(w^*)) < 1. \quad (3.4.20)$$

3. Next, \mathcal{G} the non-linear system of equations composed of each block first order KKT conditions is introduced:

$$\mathcal{G}(W, V) = \begin{bmatrix} \mathcal{G}_1(W_1, V_{2:p}) \\ \mathcal{G}_2(W_2, W_1, V_{3:p}) \\ \mathcal{G}_3(W_3, W_{1:2}, V_{4:p}) \\ \vdots \\ \mathcal{G}_p(W_p, W_{1:p-1}) \end{bmatrix} \quad (3.4.21)$$

where for all $i \in \llbracket 1, p \rrbracket$:

$$\mathcal{G}_i(W, V) = \nabla \mathcal{L}_i(W_i, W_{1:i-1}, V_{i+1:p}) = \begin{bmatrix} \nabla_{x_i} \mathcal{L}_i(W_i, W_{1:i-1}, V_{i+1:p}) \\ g_i(W_i, \Phi(W_{1:i-1}, W_i, V_{i+1:p})) \end{bmatrix}. \quad (3.4.22)$$

By construction, we have

$$\mathcal{G}(w^{k+1}, w^k) = 0. \quad (3.4.23)$$

4. As

$$\mathcal{G}(T(w^k), w^k) = 0, \quad (3.4.24)$$

there exists a neighborhood $N(w^*)$ of w^* such that

$$\forall w \in N(w^*) \quad \mathcal{G}(T(w), w) = 0 \quad (3.4.25)$$

5. It implies that within this neighborhood $\nabla T(w^*)$ can be rewritten as:

$$\nabla T(w^*) = - \left[\frac{\partial \mathcal{G}}{\partial W}(w^*, w^*) \right]^{-1} \frac{\partial \mathcal{G}}{\partial V}(w^*, w^*) = [L(w^*)]^{-1} U(w^*), \quad (3.4.26)$$

where $L(w^*)$ is a block lower triangular matrix and $U(w^*)$ a block strictly upper triangular matrix.

6. To show that the spectral radius of $\nabla T(w^*)$ is strictly inferior to 1, we'll rely on Theorem 3.4.1, i.e. we will show that there exist an eigenset of $\nabla T(w^*)$ such that, for all vector W in this eigenset, we have

$$(W^T D W)(W^T H W) > 0, \quad (3.4.27)$$

where $D = L(w^*)^T + U(w^*)$ is the block diagonal matrix composed of the Hessian of the Lagrangian of $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ at w^* and $H = L(w^*) - U(w^*)$ is the Hessian of the Lagrangian of (\mathcal{P}_{low}) at w^* :

- (a) First we characterize all the eigenvectors of $\nabla T(w^*)$.
- (b) Given an eigenvector $W = [x_1, \lambda_1, \dots, x_p, \lambda_p]$ of $\nabla T(w^*)$, we use this characterization to show that, necessarily, for all $i \in \llbracket 1, p \rrbracket$, x_i is a feasible direction for $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$. Consequently, x is a feasible direction for (\mathcal{P}_{low}) by construction.
- (c) Taking advantage of the fact that H and all D_i $i \in \llbracket 1, p \rrbracket$ are, by assumption, positive-definite along their respective feasible directions, we can easily show by direct computation that both $W^T D W$ and $W^T H W$ are strictly positive quantities.
- (d) Therefore, by Theorem 3.4.1:

$$\rho([L(w^*)]^{-1} U(w^*)) < 1. \quad (3.4.28)$$

7. Finally $T(w^*)$ is a contraction mapping around w^* which proves the local convergence of the BCD-MDF algorithm towards w^* .

Let us now describe all these steps in more detail:

Proof: Let x^* be the unique solution of the lower level problem (\mathcal{P}_{low}) and λ^* be the associated Lagrange multiplier. In this proof, we use the following block notation:

$$w^* = (w_1^*, \dots, w_p^*)$$

where $w_i^* = (x_i^*, \lambda_i^*)$, $i = 1, \dots, p$.

1st step: Defining the BCD-MDF operator. First we introduce the BCD-MDF operator i.e. the operator defining a complete iteration of the BCD-MDF algorithm (see Algorithm 2). More precisely, let $w_i^k = (x_i^k, \lambda^{i,k})$ be the unique primal-dual pair associated to $\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k)$ (which is consistent according the Unique Block Minimizer (UBM) assumption). Any complete iteration k of the BCD-MDF algorithm is of the form:

$$w^{k+1} = T(w^k) = S_p \circ S_{p-1} \circ \dots \circ S_2 \circ S_1(w^k) \quad (3.4.29)$$

where the mapping $S_i : \mathbb{R}^n \times \mathbb{R}^{n_c} \mapsto \mathbb{R}^n \times \mathbb{R}^{n_c}$ describes the result of the i^{th} sub-optimization of the BCD-MDF algorithm. Roughly speaking, S_i will update the i^{th} block as the solution to the problem \mathbf{PB}_i parameterized by $x_{\neq i}$, leaving the other blocks unchanged:

$$\forall j \in \mathbb{N}, (S_i(w))_j = w_j \quad \text{if} \quad j \neq i. \quad (3.4.30)$$

According to Fiacco's theorem (see Proposition 3.3.2) at x^* with multiplier λ^* , we easily check that each mapping S_i is well-defined and continuously differentiable in a neighborhood of $w^* = (w_1^*, \dots, w_p^*)$. By composition, the BCD-MDF operator is at least of class C^1 around w^* . Note that in the considered neighborhood of w^* the set of active constraints for each block is therefore constant. This is a direct consequence of Fiacco's theorem for the existence of continuously differentiable functions S_i . By construction, the set of active constraints for (\mathcal{P}_{low}) in this neighborhood is also constant, more precisely it is equal to the union of the active constraints of each $(\mathbf{PB}_i(x_{\neq i}))$ in the considered neighborhood.

2nd step: the BCD-MDF algorithm, a fixed-point algorithm. Assuming that the UBM property and the first order separability hold, Proposition 3.4.2 ensures that x^* is actually

a fixed point of the BCD-MDF operator T i.e. $T(w^*) = w^*$. Near the solution w^* , the Taylor expansion at w^k gives:

$$T(w^k) - T(w^*) = \nabla T(w^*)(w^k - w^*) + o(w^k - w^*) \quad (3.4.31)$$

where $\nabla T(w^*)$ denotes the Jacobian matrix of the operator T . Since w^* is a fixed point for the operator T :

$$w^{k+1} - w^* = \nabla T(w^*)(w^k - w^*) + o(w^k - w^*) \quad (3.4.32)$$

the sequence $\{w^k\}_{k \in \mathbb{N}}$ is locally convergent (towards w^*) if the spectral radius of the Jacobian matrix $\nabla T(w^*)$ satisfies: $\rho(\nabla T(w^*)) < 1$.

Since the T operator is defined implicitly, the next step is to seek an analytical characterization of this operator so that its spectral radius can be calculated. Let us now introduce the Lagrangian function \mathcal{L}_i associated to the problem $\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k)$:

$$\begin{aligned} \mathcal{L}_i(x_i, \lambda^i; x_{\neq i}) &= f(x_{1:i-1}, x_i, x_{i+1:p}, \Psi(x_{1:i-1}, x_i, x_{i+1:p})) \\ &+ (\lambda^i)^T g_i(x_i, \Psi(x_{1:i-1}, x_i, x_{i+1:p})) \end{aligned} \quad (3.4.33)$$

where $x_{\neq i} = (x_{1:i-1}, x_{i+1:p})$ denotes the vector parameterizing \mathbf{PB}_i . According to Proposition 3.4.3, the unique solution x_i^{k+1} of $\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k)$ and its associated Lagrange multiplier λ_i^{k+1} are fully characterized by the Karush-Kuhn-Tucker optimality conditions:

$$\mathcal{G}(w^{k+1}, w^k) = 0. \quad (3.4.34)$$

where the i blocks of $\mathcal{G}(w^{k+1}, w^k)$ are defined by:

$$\mathcal{G}_i(w^{k+1}, w^k) = \begin{bmatrix} \frac{\partial}{\partial x_i} \mathcal{L}_i(x_i^{k+1}, \lambda_i^{k+1}; x_{1:i-1}^{k+1}, x_{i+1:p}^k) \\ g_i(x_i^{k+1}, \Psi(x_{1:i-1}^{k+1}, x_i^{k+1}, x_{i+1:p}^k)) \end{bmatrix}, \quad i = 1, \dots, p. \quad (3.4.35)$$

In other words, the fixed point iterations of the BCD-MDF algorithm are of the form:

$$w^{k+1} = T(w^k) \iff \mathcal{G}(w^{k+1}, w^k) = 0, \quad (3.4.36)$$

hence for all $k \in \mathbb{N}$:

$$\mathcal{G}(T(w^k), w^k) = 0.$$

According to Proposition 3.3.2, there exists a neighborhood \mathcal{V}^* of w^* in which the operator \mathcal{G} is actually at least of class C^1 and:

$$\forall w \in \mathcal{V}^*, \quad \frac{d}{dw}(\mathcal{G}(T(w), w)) = \partial_1 \mathcal{G}(T(w), w) \nabla T(w) + \partial_2 \mathcal{G}(T(w), w),$$

where, to lighten the notations, $\partial_j \mathcal{G}$, $j \in \{1, 2\}$, denotes the partial derivative of \mathcal{G} with respect to its j th block of variables $w \in \mathbb{R}^{n+n_c}$.

Observe now that by construction, we have: $\frac{d\mathcal{G}}{dw}(T(w^*), w^*) = 0$, hence:

$$\nabla T(w^*) = - [\partial_1 \mathcal{G}(T(w^*), w^*)]^{-1} \partial_2 \mathcal{G}(T(w^*), w^*) \quad (3.4.37)$$

$$= - [\partial_1 \mathcal{G}(w^*, w^*)]^{-1} \partial_2 \mathcal{G}(w^*, w^*) \quad (3.4.38)$$

Denoting by $L(w^*) = \partial_1 \mathcal{G}(w^*, w^*)$ and $U(w^*) = -\partial_2 \mathcal{G}(w^*, w^*)$ a more compact form is obtained :

$$\nabla T(w^*) = [L(w^*)]^{-1} U(w^*) \quad (3.4.39)$$

and the iterative process converges towards w^* if $\rho([L(w^*)]^{-1} U(w^*)) < 1$.

3rd step: Computing the spectral radius of $\nabla T(w^)$.* Let us start by defining more explicitly the operators L and U . By definition the operator $L(w^*)$ is actually block lower triangular since each block optimization does not depend on the block variables w that have not yet been updated. Similarly, the operator $U(w^*)$ is strictly upper triangular since each block optimization only depends these same variables. More precisely:

$$L(w^*) = \partial_1 \mathcal{G}(w^*, w^*) = \begin{bmatrix} D_1 & 0 & 0 & \dots & 0 \\ L_{2,1} & D_2 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & D_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{p,1} & L_{p,2} & L_{p,3} & \dots & D_p \end{bmatrix} \quad (3.4.40)$$

and

$$U(w^*) = -\partial_2 \mathcal{G}(w^*, w^*) = - \begin{bmatrix} 0 & L_{1,2} & L_{1,3} & \dots & L_{1,p} \\ 0 & 0 & L_{2,3} & \dots & L_{2,p} \\ 0 & 0 & 0 & \dots & L_{3,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (3.4.41)$$

where for any $(i, j) \in \llbracket 1, p \rrbracket^2$, $j \neq i$:

$$D_i = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_i^2}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_i \partial x_i}(w_i^*; w_{\neq i}^*) \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_i \partial \lambda_i}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_i^2}(w_i^*; w_{\neq i}^*) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_i^2}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_i \partial x_i}(w_i^*; w_{\neq i}^*) \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_i \partial \lambda_i}(w_i^*; w_{\neq i}^*) & 0 \end{bmatrix} \quad (3.4.42)$$

$$L_{i,j} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial x_i}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_j \partial x_i}(w_i^*; w_{\neq i}^*) \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial \lambda_i}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_j \partial \lambda_i}(w_i^*; w_{\neq i}^*) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial x_i}(w_i^*; w_{\neq i}^*) & 0 \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial \lambda_i}(w_i^*; w_{\neq i}^*) & 0 \end{bmatrix}.$$

Observe now that by the [First Order Separability](#), we have for all i ,

$$\frac{\partial \mathcal{L}}{\partial x_i}(x^*; \lambda^*) = \frac{\partial \mathcal{L}_i}{\partial x_i}(w_i^*; x_{\neq i}^*) \quad (3.4.43)$$

so that:

$$\forall j \neq i, L_{i,j} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x_j \partial x_i}(w^*) & 0 \\ 0 & 0 \end{bmatrix}, \quad D_i = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x_i^2}(x^*; \lambda^*) & \frac{\partial^2 \mathcal{L}}{\partial \lambda_i \partial x_i}(x^*; \lambda^*) \\ \frac{\partial^2 \mathcal{L}}{\partial x_i \partial \lambda_i}(x^*; \lambda^*) & 0 \end{bmatrix}, \quad (3.4.44)$$

where \mathcal{L} denotes the Lagrangian function associated to the lower level problem (\mathcal{P}_{low}):

$$\mathcal{L}(x; \lambda) = f(x, \Psi(x)) + \sum_{k=1}^p \lambda_k^T g_k(x_k, \Psi(x)). \quad (3.4.45)$$

Consider now the matrix H defined as

$$H = L(w^*) - U(w^*) = \begin{bmatrix} D_1 & L_{2,1}^T & L_{3,1}^T & \cdots & L_{p,1}^T \\ L_{2,1} & D_2 & L_{3,2}^T & \cdots & L_{p,2}^T \\ L_{3,1} & L_{3,2} & D_3 & \cdots & L_{p,3}^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{p,1} & L_{p,2} & L_{p,3} & \cdots & D_p \end{bmatrix}, \quad (3.4.46)$$

by construction H is hermitian, and denote as H^* the bordered Hessian of (\mathcal{P}_{low}) at w^* , i.e. the Hessian matrix of (3.4.45) with respect to x and λ . We then observe that H^* is obtained after a row-column reordering of the matrix H . More precisely, reordering the basis of H , which follows the coordinates $[x_1, \lambda_1, \dots, x_p, \lambda_p]$, into the new system of coordinates $[x_1, \dots, x_p, \lambda_1, \dots, \lambda_p]$ yields the matrix H^* . Hence, by defining P as the $2p \times 2p$ block permutation matrix, P is defined as

$$P_{i,j} = \begin{cases} I_{n_i \times n_i} & \text{if } i = 0[2] \text{ and } j = p + \frac{i}{2} \\ I_{n_{c_i} \times n_{c_i}} & \text{if } i = 1[2] \text{ and } j = \frac{i+1}{2} \\ 0 & \text{Else} \end{cases}, \quad (3.4.47)$$

and it follows that

$$H^* = P^T H P. \quad (3.4.48)$$

To show that $\rho(\nabla T(w^*)) < 1$, we will establish the connection between the bordered Hessian of (\mathcal{P}_{low}) at w^* , i.e. H^* , and the block decomposition of its permutation H .

Now let us note that

$$L(w^*)^T + U(w^*) = \begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_p \end{bmatrix} \equiv D. \quad (3.4.49)$$

To apply Theorem 3.4.1, given our notation, the goal of this part of the proof is to show that there exists an eigenset of $\nabla T(w^*)$ such that H and $D = L(w^*)^T + U(w^*)$ are SQFSE. Since these eigensets are difficult to characterize in our case, we will consider the most general case, i.e. we will show that H and D are SQFSE on the whole space of eigenvectors of $\nabla T(w^*)$.

First, for the sake of readability, let us introduce new notations for the block decomposition. For all $i \in \llbracket 1, p \rrbracket$, let

$$D_i = \begin{bmatrix} A_i & B_i^T \\ B_i & 0 \end{bmatrix}, \quad (3.4.50)$$

and for all $j \neq i$,

$$L_{i,j} = \begin{bmatrix} C_{i,j} & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.4.51)$$

In other words, A_i is the Hessian of the Lagrangian of $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ at x^* , \mathcal{L}_i , with respect to x_i , B_i is the Jacobian of the active constraints among g_i at w^* , and $C_{i,j}$ are the cross derivatives of the Lagrangian of (\mathcal{P}_{low}) at w^* with respect to the block x_i and x_j . We

also know, that each matrix D_i is the bordered Hessian of $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ at x^* , which is by nature indefinite. An easy way to see this property is to pick a vector $d \in \mathbb{R}^{n_i+n_{c_i}}$ such that the only non-zero elements are on the λ_i coordinates, which necessarily leads to $d^T D_i d = 0$. Following the new notations, we also have a new expression for H^* :

$$H^* = P^T H P = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad (3.4.52)$$

with

$$A = \begin{bmatrix} A_1 & C_{2,1}^T & \cdots & C_{p,1}^T \\ C_{2,1} & A_2 & \cdots & C_{p,2}^T \\ \vdots & \vdots & \ddots & \vdots \\ C_{p,1} & C_{p,2} & \cdots & A_p \end{bmatrix}, \quad (3.4.53)$$

and

$$B = \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_p \end{bmatrix}. \quad (3.4.54)$$

Let us characterize the eigenvectors of $\nabla T(w^*)$.

Let $W = [w_1, \dots, w_p] = [x_1, \lambda_1, \dots, x_p, \lambda_p]$ be an eigenvector of $\nabla T(w^*)$ associated to an eigenvalue μ . Then:

$$U(w^*)W = \mu L(w^*)W \quad (3.4.55)$$

$$\iff \forall i \in \llbracket 1, p \rrbracket \quad - \sum_{k=i+1}^p L_{i,k} w_k = \mu \left(- \sum_{k=1}^{i-1} (L_{i,k} w_k) + D_i w_i \right). \quad (3.4.56)$$

A straightforward calculation shows that

$$\forall i \in \llbracket 1, p \rrbracket \quad D_i w_i = \begin{bmatrix} A_i x_i + B_i^T \lambda_i \\ B_i x_i \end{bmatrix}, \quad (3.4.57)$$

and that

$$\forall (i, k) \in \llbracket 1, p \rrbracket \times \llbracket 1, p \rrbracket, \quad k \neq i, \quad L_{i,k} w_k = \begin{bmatrix} C_{i,k} x_k \\ 0 \end{bmatrix}. \quad (3.4.58)$$

There is two possible cases for the value of μ : either $\mu = 0$ or either μ is nonzero. Let us first consider the case $\mu = 0$. By reconsidering (3.4.55), we have $W \in \ker(U(w^*))$, which combined with (3.4.49) and (3.4.46) yields directly:

$$W^T H W = W^T (L(w^*) - U(w^*)) W \quad (3.4.59)$$

$$= W^T L(w^*) W \quad (3.4.60)$$

$$= W^T (D^T - U^T(w^*)) W \quad (3.4.61)$$

$$= W^T D W + (U(w^*) W)^T W \quad (3.4.62)$$

$$= W^T D W. \quad (3.4.63)$$

Hence for $\mu = 0$,

$$(W^T D W)(W^T H W) = (W^T D W)^2 > 0. \quad (3.4.64)$$

Assume now that the eigenvalue μ is nonzero, so the second block line of (3.4.56) implies that

$$\forall i \in \llbracket 1, p \rrbracket \quad B_i x_i = 0, \quad (3.4.65)$$

which is equivalent to

$$\forall i \in \llbracket 1, p \rrbracket \quad x_i \in \ker(B_i). \quad (3.4.66)$$

Hence $x = [x_1, \dots, x_p] \in \ker(B)$ by construction.

Let us now study the sign of $(W^T DW)(W^T HW)$. First we calculate $W^T DW$:

$$W^T DW = \begin{bmatrix} w_1^T & \dots & w_p^T \end{bmatrix} \begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_p \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix} = \sum_{i=1}^p w_i^T D_i w_i, \quad (3.4.67)$$

with

$$\forall i \in \llbracket 1, p \rrbracket \quad w_i^T D_i w_i = x_i^T A_i x_i + x_i^T B_i^T \lambda_i + \lambda_i^T B_i x_i. \quad (3.4.68)$$

Since for all $i \in \llbracket 1, p \rrbracket$, we have $x_i \in \ker(B_i)$, then

$$\forall i \in \llbracket 1, p \rrbracket \quad w_i^T D_i w_i = x_i^T A_i x_i > 0, \quad (3.4.69)$$

as each D_i $i \in \llbracket 1, p \rrbracket$ are positive-definite along the feasible direction of $(\mathcal{P}_{B_i}(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ at w^* . So we have shown that

$$W^T DW > 0. \quad (3.4.70)$$

We then observe via a straightforward calculation that $W^T HW = x^T Ax$. Since $x \in \ker(B)$, i.e. since x is an admissible direction for (\mathcal{P}_{low}) , we necessarily have $x^T Ax > 0$ and

$$W^T HW > 0. \quad (3.4.71)$$

Finally, by (3.4.64) for $\mu = 0$, and by (3.4.70) and (3.4.71) for μ being nonzero, it follows for all eigenvectors W of $\nabla T(W^*)$

$$(W^T DW)(W^T HW) > 0. \quad (3.4.72)$$

Hence D and H are **SQFSE** on the set of all eigenvectors of $\nabla T(w^*)$, and according to Theorem 3.4.1, it follows that $\rho(\nabla T(w^*)) < 1$. This concludes on the local convergence of the BCD operator towards w^* q-linearly. \square

Discussion on the local convergence result The previous result, presented in Proposition 3.4.3, provides sufficient conditions to solve (\mathcal{P}_{low}) exactly. In addition, if Fiacco's theorem holds for (\mathcal{P}_{low}) (cf. Proposition 3.3.2), it makes all functions of interest in $(\mathcal{P}_{B_i-level})$, namely the objective and the constraints, at least C^1 functions with respect to the shared variables x_0 . Consequently, general gradient-free algorithms, even those whose convergence properties are enhanced by the C^1 property, may be employed to solve $(\mathcal{P}_{B_i-level})$ and obtain the same result as solving (\mathcal{P}_{MDF}) .

However, this result is highly theoretical and several limitations must be acknowledged. First and foremost, this can be seen as an extension of the results given in [64] by

Grippe and Sciandrone in the non-separable case, therefore stronger assumptions are made and may be challenging to verify, particularly the [Unique Block Minimizer](#) and the [First Order Separability](#) assumptions, these two hypothesis are made without further investigations.

While the [Unique Block Minimizer](#) assumption may seem reasonable for the considered applications, and can be tested to some extent by multi-starting the block optimizations, this is not necessarily the case for the second, the [First Order Separability](#) hypothesis, which assumes, among other things, that the separation of constraints does not affect each block solution. In other words, it implies that at convergence, the reunion of all solutions of $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$, $x_i^*(x_{\neq i}^*)$, is strictly equal (or at least very close) to the joint solution x^* of (\mathcal{P}_{low}) . It is likely that the [First Order Separability](#) assumption is too strict for the results we have shown in this manuscript, and that it may not even be necessary. Finding better, less restrictive assumptions is one of the important areas of improvement for the work presented.

In addition to these theoretical considerations, the demonstrated properties are also highly susceptible to practical implementation and setup. Indeed, this implies the optimal resolution of all sub-optimization problems, including all block optimizations and (\mathcal{P}_{low}) (through the BCD loop). In practice, however, this may not be the case due to numerical errors or sub-optimal hyper-parameters (such as all the tolerances). If these issues typically necessitate expertise on the specific problem and implementation to be resolved, one might inquire whether there are potential avenues for action on a more abstract level, involving modifications to the underlying architecture.

3.5 Variants of the BCD-MDF algorithm

In consideration of the [First Order Separability](#) assumption and to a lesser extent the [Unique Block Minimizer](#) hypothesis, it is possible to modulate the block decomposition in order to lighten the assumptions. The newly proposed architectures, which will be presented in this section and referred to as variants, illustrate the adaptability of the original bi-level architecture. Two of these variants only impact the resolution of the lower-optimization problem (\mathcal{P}_{low}) . The first variant focuses on improving each block's robustness by providing surrogates of other blocks' complicating constraints. The second variant mitigates the utilization of costly MDAs within each block optimization, albeit at the cost of worst convergence properties. Finally a more macro variant is proposed that impacts the entire bi-level architecture. Once again, the objective is to lighten the [First Order Separability](#) assumption, but this time by directly addressing the coupling variables by bringing them up to the system level, thereby eliminating them from the lower optimization problem.

3.5.1 With linear approximation of the constraints

If the [First Order Separability](#) hypothesis does not hold, meaning that some constraints depend too much on the other block variables, a linearization of these constraints can be added to the corresponding blocks to enforce it. In this case, the iteration scheme

become:

$$x_i^{k+1} = \arg \min_{t \in \mathcal{X}_i} \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \\ s.t. \quad g_i(t, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \leq 0 \\ \tilde{g}_j(t) \leq 0 \quad j \in \mathcal{S}_i \end{array} \right\} \quad (3.5.1)$$

Where \mathcal{S}_i is the set of index of constraints which do not respect the [First Order Separability](#) assumption along i^{th} block optimization and $\tilde{g}_j : \mathbb{R}^{n_i} \mapsto \mathbb{R}^{n_{c_j}}$ the corresponding linear approximation of constraint j along x_i .

This variant suggests that the overall BCD algorithm process becomes more complex as the linear approximation must be updated with greater frequency. Furthermore, if the considered constraints exhibit significant non-linearity, it is reasonable to anticipate that the BCD loop will require a greater number of iterations to achieve convergence. These functions approximate the constraints of the other disciplines in the current block, in order to avoid their violation.

Nevertheless, this variant retains the quality of the previously demonstrated properties (e.g. the feasibility of the fixed point and the local convergence) on the BCD. These proofs remain valid under the same set of assumptions, with the replacement of Equation ([PB_i\(x_{≠i}\)](#)) by Equation (3.5.1). In particular, the hypothesis in the [First Order Separability](#) assumption can be relaxed, as each constraint in each \mathcal{S}_i is guaranteed to satisfy this assumption.

The architecture resulting from this novel variant of the BL-BCD-MDF algorithm is designated BL-BCD-MDF-LC (Bilevel - Block Coordinate Descent - MDF - Linearized Constraints architecture). It should be noted that the linear approximations of the constraints can be equivalently replaced by surrogates that maintain first-order consistency (i.e. the value of the functions and their gradients) at the points under consideration.

If first-order surrogates are not feasible in practice, it may indicate that the disciplinary separation is not physically pertinent. In such a scenario, the effect of the couplings within block j may dominate the impact of x_i on the disciplinary optimization of block i . In this case, the corresponding blocks would need to be merged and optimized together in (x_i, x_j) . This approach would effectively eliminate the discrepancy between blocks i and j . In the most extreme case where all the blocks are too interdependent, merging all the blocks leads to the original MDF architecture, which is convergent towards x^* by assumption. However, this necessitates the fully coupled derivatives with respect to all the local design variables.

3.5.2 A weakly coupled variant

The BCD-MDF algorithm addresses the issue of coupling by running MDAs in each sub-optimization. This approach ensures that couplings are coherent and at equilibrium at each step of the overall process. However, depending on the problem's structure, solving an entire MDA at each iteration of a sub-optimization algorithm may not be feasible in practice for various reasons. Firstly, obtaining the coupling variables at convergence through a MDA is typically time-consuming. Consequently, looping on block optimizations sequentially that rely heavily on such algorithms may be prohibited. Secondly, from an architectural viewpoint, some blocks may not be given access to all other coupling functions, or at least to the coupled derivative of the MDA with respect to their block of variables.

As a result, we propose a weakly-coupled version of the previous BCD-MDF algorithm, releasing the couplings constraints in each sub-optimization at the price of degraded convergence properties for the overall procedure.

Consider the following (\mathcal{IDF}) reformulation of (\mathcal{P}_{low}):

$$\min_{x, y, y^T} f(x, y) \quad \text{s.t.} \quad g(x, y) \leq 0 \quad \text{and} \quad y_i - \phi_i(x_i, y_{\neq i}^T) = 0 \quad \forall i \in \llbracket 1, p \rrbracket \quad (3.5.2)$$

where ϕ_i is the coupling function computed by discipline i and in which the dependency on the shared variables x_0 have been avoided. This optimization problem is equivalent to the previous optimization problem (\mathcal{P}_{low}) by equivalence of the MDF and IDF architectures.

The proposed variant of the BCD-MDF algorithm does not longer solves the whole coupling vector by solving a complete MDA, but now only compute its own block of coupling variables. Similarly as the design vector x , the coupling vector y is decomposed by blocks and updated by sequentially running each sub-optimization. The weak BCD procedure (BCD-WK) is described as:

$$x_i^{k+1} = \arg \min_{t \in \mathcal{X}_i} \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \\ \text{s.t.} \quad g_i(t, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \leq 0 \end{array} \right\} \quad (\text{BCD-WK})$$

$$\text{s.t.} \quad y_i^{k+1} = \phi_i(x_i^{k+1}, y_{1:i-1}^{k+1}, y_{i+1:p}^k)$$

The weak BCD procedure can be viewed as the merge of both the design variables loop and the MDA loop, updating both of them at the end of each sub-optimization, with the expectation that the coupling vector will converge as long as the local variables are converging towards x^* . It is important to note that the block of coupling variables, y_i , is not considered a design variable by the optimizer. This is due to the availability of the coupling function, ϕ_i , which is employed directly to compute the functions of interest, thereby ensuring the individual feasibility of discipline i by design. The optimal value y_i^{k+1} is thus recovered at the conclusion of the block optimization process, as it has already been computed at the corresponding optimal design x_i^{k+1} .

This architecture will be referred as BL-BCD-WK (Bi-Level - Block Coordinate Descent - Weakly coupled architecture). A generic XDSM diagram on a two strongly coupled disciplines problem is given by Figure 3.7. The disciplinary optimizations are represented within Figure 3.8.

Similarly as the MDF counterpart, the BCD-MDF algorithm, adding linearization (or if available surrogates verifying the first-order consistency) of the constraints to certain blocks when the [First Order Separability](#) assumption isn't verified is highly recommended. The resulting architecture follows the same convention of naming and is referred, for the whole bi-level architecture, as BL-BCD-WK-LC (Bi-Level - Block Coordinate Descent - Weakly coupled - Linearized Constraints architecture).

The BCD-WK algorithm is outlined in detail in Algorithm 3. To illustrate the versatility of the proposed methodologies, the preceding variant, which incorporates linearized constraints on specific blocks, is also included.

Algorithm 3: Block Coordinate Descent - Weakly coupled (BCD-WK) algorithm

Data: $x_0, x^0, y^0, \varepsilon_{var}, \varepsilon_{fun}$

Result: x^*

$k, i = 0, 1;$

$f^0 = f(x_0, x^0, y^0);$

while $\frac{\|x^{k-1} - x^k\|}{\|x^k\|} > \varepsilon_{var}$ *OR* $\frac{\|y^{k-1} - y^k\|}{\|y^k\|} > \varepsilon_{var}$ *OR* $\frac{\|f^{k-1} - f^k\|}{\|f^k\|} > \varepsilon_{fun}$ **do**

$x^{k+1}, y^{k+1} = x^k, y^k;$

for $i \in [1, p]$ **do**

 Optimize i^{th} Block: ;

if *FOS holds for i^{th} Block* **then**

$x_i^{k+1} =$

$$\arg \min_t \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \\ s.t. \quad g_i(t, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \leq 0 \end{array} \right\}$$

end

else

$x_i^{k+1} =$

$$\arg \min_t \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \\ s.t. \quad g_i(t, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \leq 0 \\ \quad \tilde{g}_j(t) \leq 0 \quad j \in \mathcal{S}_i \end{array} \right\}$$

end

$y_i^{k+1} = \phi_i(x_i^{k+1}, y_{1:i-1}^{k+1}, y_{i+1:p}^k);$

 Update the i^{th} local data: $x^{k+1}[i], y^{k+1}[i] = x_i^{k+1}, y_i^{k+1};$

end

 Compute the objective value: $f^{k+1} = f(x_0, x^{k+1}, y^{k+1});$

$k = k + 1;$

end

$x^*, y^* = x^k, y^k;$

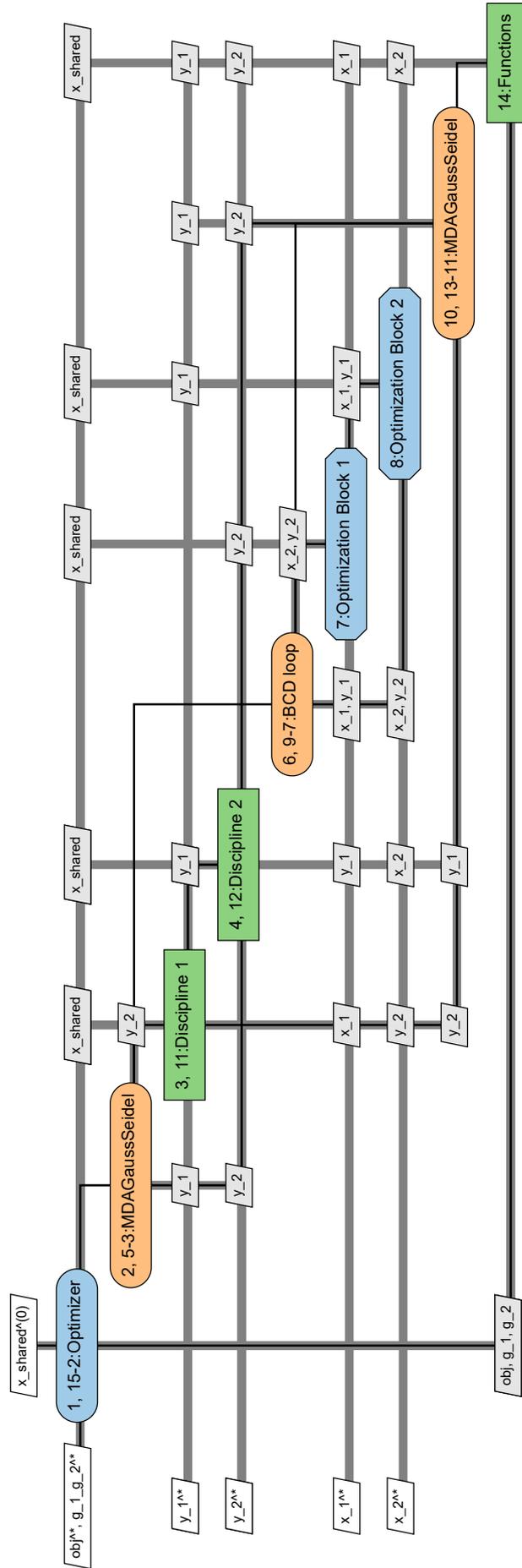


Figure 3.7: XDSM of the BL-BCD-WK architecture



Figure 3.8: XDSM of the disciplinary block optimizations within the BCD loop of the BL-BCD-WK architecture

3.5.3 Adding target values for difficult couplings

In case the previous variants are not sufficient to guarantee a satisfactory block decomposition for the purpose of solving the lower-level optimization problem, it is possible to intervene in the bi-level decomposition itself. Suppose that there is a subset of coupling variables that threatens the coherence of the block decomposition. This may be due to the failure of either the [First Order Separability](#) or the [Unique Block Minimizer](#) assumptions. The management of this subset of coupling variables can be handled by the system problem. Similar to the IDF approach, copies of these couplings, also known as target values, can be added to the system-level optimizer and held fixed for the lower-level optimization problem. The set of indices of these couplings is denoted as \mathcal{C}_0 , and the following bi-level decomposition is obtained:

$$\begin{aligned} \min_{x_0 \in \mathcal{X}_0, y_0^T} \quad & f(x_0, x^*(x_0, y_0^T), \tilde{\Psi}(x_0, x^*(x_0, y_0^T), y_0^T)) \\ \text{s.t.} \quad & g^{up}(x_0) \leq 0 \end{aligned} \quad (3.5.3)$$

where the lower-optimization problem is reformulated as follows:

$$x^*(x_0, y_0^T) = \arg \min_{x \in \mathcal{X}} \left\{ \begin{array}{l} f(x_0, x, \tilde{\Psi}(x_0, x, y_0^T)) \\ \text{s.t. } g(x_0, x, \tilde{\Psi}(x_0, x, y_0^T)) \leq 0 \\ y_{0,i}^T - \phi_i(x_0, x_i^*(x_0, y_0^T), \tilde{\Psi}_{\neq i}(x_0, x^*(x_0, y_0^T), y_0^T)) = 0 \quad \forall i \in \mathcal{C}_0 \end{array} \right\} \quad (3.5.4)$$

where $\tilde{\Psi}$ is a slight modification of the complete MDA Ψ where all couplings in \mathcal{C}_0 are held fixed at y^T . In other words $\tilde{\Psi}(x_0, x, y^T)$ is the unique solution of the following system of equations :

$$\begin{aligned} \forall i \in \mathcal{C}_0 \quad & y_i - y_i^T = 0 \\ \forall i \in \llbracket 1, p \rrbracket \setminus \mathcal{C}_0 \quad & y_i - \phi_i(x_0, x_i, y_{\neq i}) = 0 \end{aligned} \quad (3.5.5)$$

This modification alleviates the block decomposition assumptions by making each block independent from the i^{th} block optimization $\forall i \in \mathcal{C}_0$. This is due to the i^{th} disciplinary analysis ϕ_i being fixed leading to $\frac{\partial \tilde{\Psi}}{\partial x_i} = 0$ for each block optimization.

Similarly to the previous variants, the choice to introduce target variables, and which ones, is the result of a compromise. On one side, adding targets for coupling variables enforce each block's autonomy, making it easier to obtain the solution of (\mathcal{P}_{low}) and continuity,

differentiability of the x^* mapping. On the other side, each added targets makes the system level optimization problem heavier similarly as in the classical IDF architecture: retrieving multidisciplinary feasibility at the end of the optimization may be difficult and take, in average, more system-level iteration, including more resolution of (\mathcal{P}_{low}).

3.6 Numerical experiments

In this section, numerical experiments are carried out on small test cases to illustrates the theoretical results. Three bi-level architectures are considered: the BL-IRT architecture, the BL-BCD-MDF architecture and one of its variants, the BL-BCD-WK architecture. A fourth architecture, the MDF architecture, is also present to have a point of comparison with a classical monolithic architecture.

First the Sobieski’s Super-Sonic Business Jet (SSBJ) test case [139] (cf. Appendix A) is considered. First of all, the noise reduction on the system function is studied. Next a benchmark is considered to study the relative robustness of all these architectures on the choice of the starting point.

Finally, a slightly modified Sellar optimization problem [135] (cf. Appendix B.1) is considered. A second block of local variables has been added to the optimization problem to allow a two-block decomposition, and all local variables, coupling variables, and constraints have been vectorized to be scalable on demand. Its particular simplicity allows for an affordable scalability study comparing a gradient-free approach for both monolithic and distributed architectures when faced with high-dimensional local variables, couplings, and constraints.

3.6.1 Discrepancy reduction and local convergence comparisons on SSBJ

In this section, the Sobieski’s Super-Sonic Business Jet (SSBJ) test case [139] is considered. This test case, which is summarized within Appendix A, was originally designed for the development of the BLISS architecture. Consequently, this test case is a primary choice for the testing of more general bi-level architectures, such as those described in this chapter.

Discrepancy measurement

From the perspective of the system level, the objective function of the previous BL-IRT architecture is not a function in practice, in the sense that its response is not deterministic with respect to the choice of the shared variables x_0 . In fact, evaluating the objective, f , triggers the lower optimization problem’s resolution, which was previously assumed to be highly dependent on the previous value of the local and coupling variables. It has been postulated that an approximated solution of \mathcal{P}_{low} will have a significant impact on the subsequent optimization, as it parameterizes each block optimization process, each of which is executed in parallel only once. The objective of this initial application is to quantify this discrepancy (or equivalently this noise) on the BL-IRT system objective and ascertain whether the BCD-based bi-level architectures are capable of reducing it to a tolerable level from the optimization algorithm perspective.

To quantify the discrepancy, a set of n randomly generated values for the shared variables x_0 are employed as sample points using the classical Latin Hypercube Sampling (LHS) method [105]. Subsequently, the objective function, f , is evaluated at each of the sample points in a fixed sequence, referred to as the forward order $\{(x_0)_k^{fd}\}_{k \in [1..n]}$. Upon evaluating the objective function, the lower optimization problem is resolved, and a sequence is generated $\{(x^*)_k^{fd}\}_{k \in [1..n]}$ that is concealed from the system optimizer. This sequence is defined by $(x^*)_k^{fd} = x^*((x_0)_k^{fd}, (x^*)_{k-1}^{fd}, (y^*)_{k-1}^{fd})$, where $x_{(k-1)}^*$ and $y_{(k-1)}^*$ are the previous values taken by the local and coupling variables, respectively, which are obtained from the evaluation of f at x_0^{k-1} .

Concurrently, the identical process is executed in reverse order, designated as the backward order $\{(x_0)_k^{bd}\}_{k \in [1..n]}$. Similarly, a sequence $\{(x^*)_k^{bd}\}_{k \in [1..n]}$ of local variables is generated. By definition, $(x_0)_k^{fd} = (x_0)_{n+1-k}^{bd} \quad \forall k \in [1..n]$. However, the same is not true for the local variables, as their values depend on the path taken by their previous evaluations. Consequently, the noise introduced into the function f by the process can be quantified using the following formula:

$$E_k = \log_{10} \left(\frac{\|f((x_0)_k^{fd}, (x^*)_k^{fd}) - f((x_0)_{n+1-k}^{bd}, (x^*)_{n+1-k}^{bd})\|}{\|f((x_0)_k^{fd}, (x^*)_k^{fd})\|} \right) \quad (3.6.1)$$

Figure 3.9 shows the distribution of the relative discrepancy error computed on 500 randomly generated values of x_0 , using the aforementioned LHS algorithm, for the three bi-level architectures investigated in this paper.

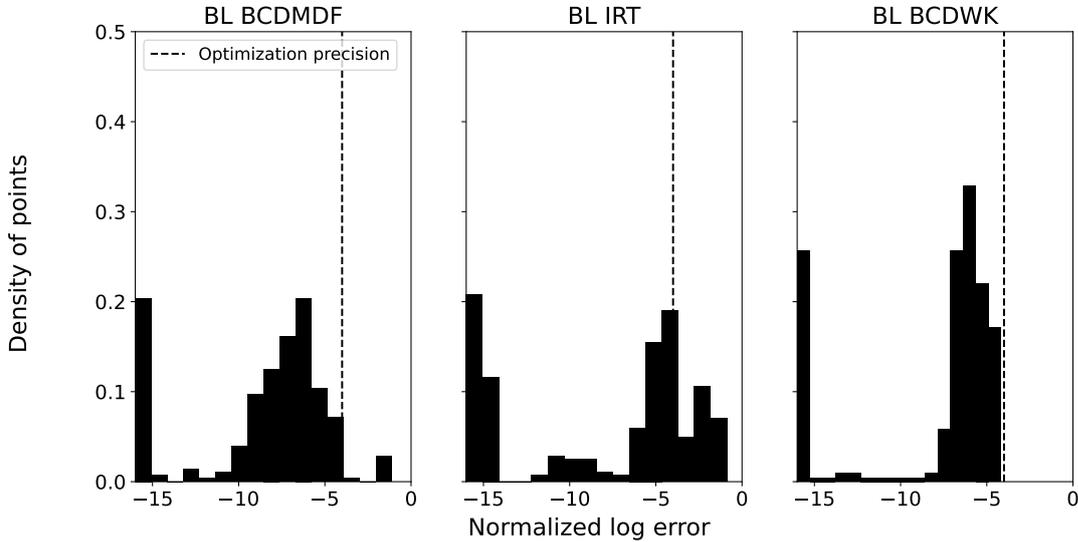


Figure 3.9: Measure of the discrepancy of the objective function for a randomly generated x_0 for SSBJ.

Figure 3.9 demonstrates that the BCD-based architectures exhibit a lower discrepancy in the objective function compared to the IRT bi-level architecture. The latter exhibited approximately 20% of the 500 randomly generated shared design values that generated a noise level exceeding the required precision (of 10^{-4} on both the objective and design) on the lower-problem resolution. The BL-BCD-MDF architecture was unable to reduce

the discrepancy below the requested precision for all the considered design points, with approximately 5% of cases resulting in failure. The remaining errors that exceed the required precision are attributable to the settings of some hyperparameters that prevent a proper convergence towards the unique minimum of (\mathcal{P}_{low}). In contrast, the BL-BCD-WK architecture demonstrated success in this specific application. The most probable explanation for its superior performance relative to BL-BCD-MDF is that, in contrast to the previous approach, the BCD-WK algorithm is now free to explore a wider range of design variables, including both the local x and coupling y variables, which allows it to produce a more robust solution based on the initial parameterization.

The application was done with all recycling mechanics, namely warm-start of the MDAs, BCD loop and block optimization, being enabled for all the considered architectures. Deactivating all these mechanisms allows the precision to be fully controlled by the tolerances given to the architecture. Indeed, this forces $(x^*)_k^{fd} = (x^*)_{n+1-k}^{bd} \quad \forall k \in [1..n]$ as $(x^*)_k^{fd} = x^*((x_0)_k^{fd}, (x^*)_0, (y^*)_0)$, where the latter terms are now deterministic if the values that compose $(x_0)_k^{fd}$ are already known.

It is important to note that this remark is only relevant when considering BCD-based bi-level architectures. Indeed, in contrast to the other bi-level architectures, the recycling mechanics, particularly warm starting each block optimization, is essential for any numerical optimization at the system level. Without the warm start of each block between each system iteration, each block optimization would ignore the evolution of other blocks' variables. One of the direct consequences is that the discrepancy problem is harder to control for the functions of interest to the IRT bi-level architecture than for the BCD-based functions. Since the warm-start mechanism cannot be disabled, the discrepancy cannot be fully controlled and is therefore inherent in the design of the BL-IRT architecture.

A first optimization on a suitable starting point

The objective is now to maximize the range of the SSBJ test case. The gradient-free algorithm COBYLA [119, 120, 121]) is employed to solve the system level optimization, and the state of the art SQP gradient-based SNOPT [58] algorithm, wrapped in pyoptspase [157], is utilized to solve the blocks optimizations.

The selection of COBYLA is based on the premise that gradients with respect to the shared variables x_0 are not available at the system level. Moreover, as the g_2 constraint computed by the Aerodynamics discipline only depends on the shared variables x_0 , the system level optimization problem is constrained. Note that, in particular, the SSBJ's system constraint (i.e. g_2) matches the definition of g^{up} given in Equation ($\mathcal{P}_{Bi-level}$). Finally, based on experience, the first-order approximations of the functions of interest utilized by COBYLA render the bi-level convergence more robust to the noise introduced by discrepancy errors.

As the considered problem is relatively small and that the optimal values for all considered functions and design variables are already known (see Table A.1 within Appendix A), it is possible to study beforehand if the chosen application is well-suited for the presented bi-level architectures.

In consideration of the hypothesis, at the shared variables' optimal value x_0^* , the MDF solution for \mathcal{P}_{low} is well-obtained in $x^* = [x_1^*, x_2^*, x_3^*]$, verifies LICQ and the First and Second order KKT necessary conditions for problem \mathcal{P}_{low} . Both the Aerodynamics and

the Propulsion blocks satisfy the [Unique Block Minimizer](#) assumption in the vicinity of x_0^* . The solutions obtained by each block correspond to the coupled optimal primal-dual pair of MDF, thereby validating the [3.4.4](#) assumption. The block Structure’s optimal solution reaches the upper bound of x_1 , rather than merely approaching it. Consequently, slight alterations in x_1^* result in the activation of the upper bound, accompanied by a non-zero Lagrange multiplier. Although this has a negligible impact on the objective (the error being approximately 10^{-2}).

Considering the initial design point, recalled in [Table A.1](#) within [Appendix A](#) and that is considered as a "good" starting point, a first optimization is carried out. [Figure 3.10](#) gather all the objective values during the aforementioned optimization, for each block’s perspective.

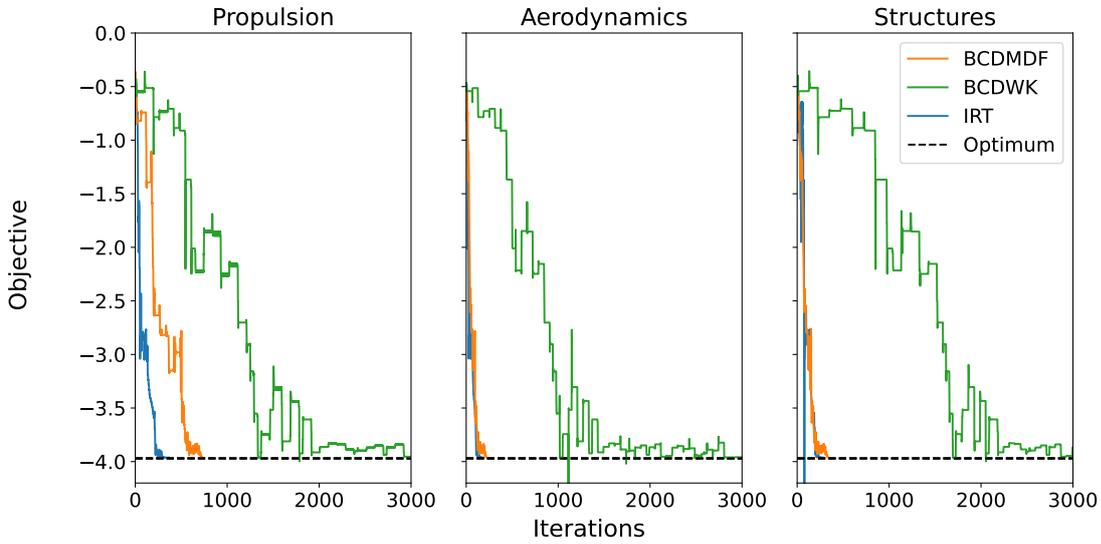


Figure 3.10: Block evolution for all considered bi-level architectures, the represented objective being scaled with a factor $-1e^{-3}$

From this initial optimization, several observations can be made. First, all architectures were able to reach the MDF optima with the corresponding optimal objective being $f^* = 3963.88$. Both BL-IRT and BL-BCDMDF’s optimization curves exhibited similar overall behavior, with the latter demonstrating the expected over-cost due to the BCD loop. The bi-level architecture that relies on the BCDWK algorithm encountered greater challenges in reaching the optimal solution. Frequently, retrieving a multidisciplinary feasible design led to a deterioration in the objective. Moreover, it can be observed that the number of discipline calls for both the Aerodynamics and Structure disciplines is similar for both the BL-IRT and BL-BCDMDF architectures. This indicates that both blocks converged rapidly to their optimal values, suggesting that the BCD loop did not result in additional discipline calls for either of them. Conversely, the propulsion block exhibited a longer convergence time, indicating that this discipline’s optimization is particularly influenced by extra variables, in contrast to the two others.

Robustness in the choice of the starting point

To illustrate the effect of the discrepancy reduction on convergence properties, a complete optimization benchmark is performed on the SSBJ test case with the three bi-level architectures, with the MDF architecture being present for comparison with a monolithic architecture. In order to address the potential bias introduced by the algorithm performance on the starting point, and to generate performance profiles that are the main standard methodology for comparison of optimization algorithms [44], the computational cost and objective values are computed on 150 random starting points $(x_0)^0$.

The true optima for SSBJ being at $f^* = 3963.88$, three targets are defined for the performance profiles at respectively 3000, 3900 and 3957 the later, if reached, being considered as fully converged considering a gradient-free approach is used (ie 0.2% error).

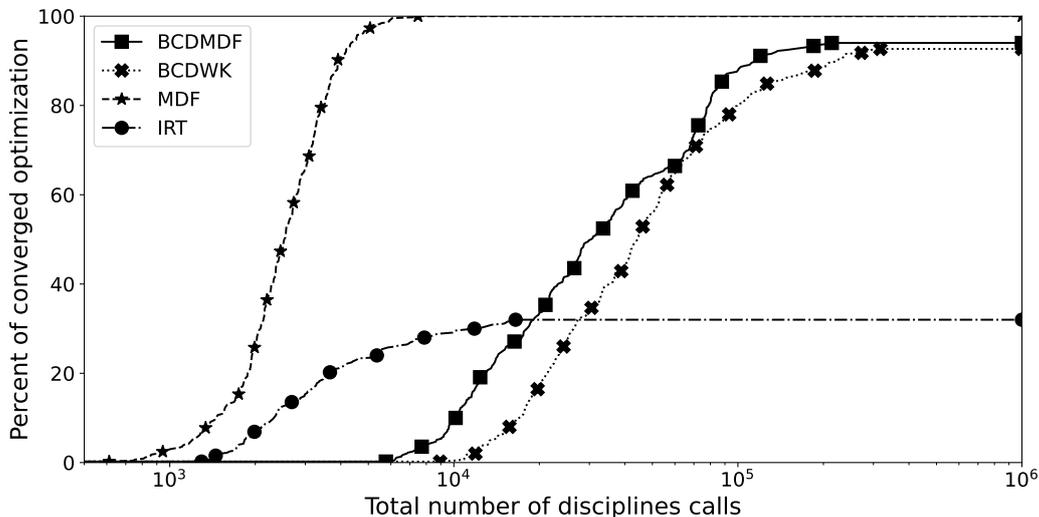


Figure 3.11: Percents of targets reached relating to the total number of disciplines calls for each considered architectures, BCD loop converge at 10^{-3}

Results from Figure 3.11 illustrates that both BL-BCD-MDF and BL-BCD-WK reached at least 90% of the above defined targets while BL-IRT reached around 30%. Note that the MDF architecture reached the SSBJ optima for every starting points as expected.

The BL-IRT architecture's low percentage of targets reached and high dependency to the starting point can be attributed, at least to some extent, to the fact that the first $n_0 + 1$ iterations of the COBYLA algorithm are dedicated to the sampling of the functions of interest around the starting point. Consequently, the approximations are built on the points where the discrepancy error is high, due to the fixed distance between these points.

The average 10% of targets that have not been reached by BCD-based architectures can be attributed to numerical issues related to a conflict between the optimizer's stopping criteria and requested tolerances for each aspect of the bi-level architectures. Experiments have demonstrated that non-converging starting points can be converted into converging ones when tolerances for either the block optimization or the BCD loop are modified, although this may result in previous convergent starting points no longer reaching the

optimum. An excessively wide BCD loop tolerance may result in the local variables x remaining unchanged, despite a low impact on the block optimization. This can lead to the system optimizer prematurely halting the optimization process. Conversely, a BCD loop tolerance that is excessively restrictive may be impossible to converge in practice due to the effects of couplings, i.e. the [First Order Separability](#) assumption does not hold for every value of x_0 . By repeating the failed starting points for both BCD-based architectures with another BCD loop tolerances (between 10^{-3} and 10^{-8}), one can reduce the impact on the convergence. These results are indicated with the suffix **corr** in Figure 3.12.

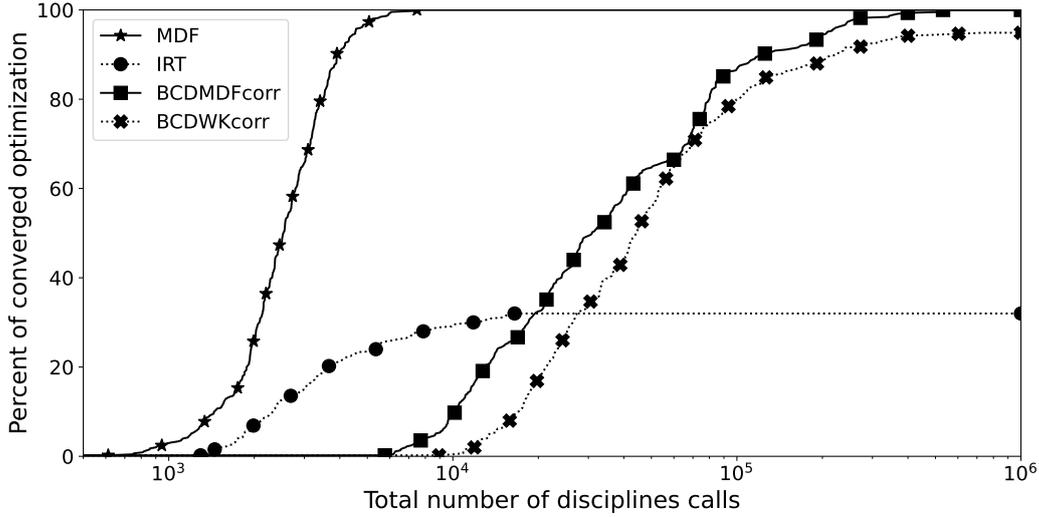


Figure 3.12: Percents of targets reached relating to the total number of disciplines calls for each considered architectures, BCD loop converge between 10^{-3} and 10^{-8}

In the case of BL-BCD-MDF, all optimizations reached the optimal solution, once the tolerances were adjusted for each of them. However, BL-BCD-WK failed to reach the final target for some starting points, regardless of the choice made regarding the BCD loop tolerance. The specific challenge for BL-BCD-WK is that the BCD loop simultaneously controls the couplings and the local design variables convergence. It is typical that coupling variables require a higher level of accuracy than design variables. Consequently, achieving a high convergence accuracy for optimizations is more challenging than for couplings by a MDA.

With regard to the computational cost of utilizing the newly proposed bi-level architectures, the BCD variant necessitates 10 times more discipline calls than the bi-level IRT architecture. However, this is justified by its enhanced robustness. In comparison to gradient-free MDF, the cost may be multiplicative by 100 to 1000. However, the SSBJ test case is a small MDO problem that was selected for its relevance in terms of block decomposition. It is not representative of the dimensions of real MDO problems, in which hundreds of design variables and thousands of constraints are involved. In the latter, a derivative-free MDF resolution is expected to fail, in contrast to bi-level strategies using disciplinary gradients such as in [55]. A further benchmark is then proposed in order to address the question of scalability with respect to the number of local variables, coupling variables, and constraints. This is discussed in the next section.

Another drawback of the presented benchmark problem is the high cost of the different MDAs used within the block optimizations; better performances can be expected by using more optimized MDAs that are specific to each disciplinary optimization. These modifications, in line with the justifications for the block decomposition in the first place, would allow to use solvers that converge both the design and the couplings, while avoiding to run all the disciplines in each block optimization.

This benchmark illustrates the capabilities of the newly proposed BCD-based architectures. The discrepancy error reduction allows for a much wider range of convergence, but at the cost of significantly higher computational complexity. It also shows that a dynamic update of the tolerances for the BCD loop and the nested block optimization should be present to avoid unnecessary block optimization. This should lead to further research.

3.6.2 Scalability study

A scalable two-block decomposition of the Sellar problem

The original Sellar problem [135], also described with Appendix B.1, has been slightly modified to allow for a block decomposition in the local variables. A dependency on a newly created variable x_2 has been added to the objective function and the y_2 analysis function. This modification has been made in order to create a symmetry in both blocks, and does not change the optimum in the shared variables x_0 . The obtained two-blocks Sellar problem is defined as follow:

$$\begin{aligned}
& \text{minimize } f(x_{0,1}, x_{0,2}, x_1, x_2, y_1, y_2) = x_1^2 + x_2^2 + x_{0,2} + y_1^2 + e^{-y_2} \\
& \quad \text{with respect to } x_{0,1}, x_{0,2}, x_1, x_2, y_1, y_2 \\
& \quad \text{subject to :} \\
& \quad -10 \leq x_{0,1} \leq 10 \\
& \quad 0 \leq x_{0,2} \leq 10 \\
& \quad 0 \leq x_1 \leq 10 \\
& \quad 0 \leq x_2 \leq 10 \\
& \quad c_1(y_1) = 3.16 - y_1^2 \leq 0 \\
& \quad c_2(y_2) = y_2 - 24 \leq 0 \\
& \quad y_1 - y_1(x_{0,1}, x_{0,2}, x_1, y_2) = 0 \\
& \quad y_2 - y_2(x_{0,1}, x_{0,2}, x_2, y_1) = 0
\end{aligned} \tag{3.6.2}$$

With the disciplinary analysis being given by:

$$\textbf{Discipline 1: } y_1(x_{0,1}, x_{0,2}, x_1, y_2) = \sqrt{x_{0,1}^2 + x_1 + x_{0,2} - 0.2y_2} \tag{3.6.3}$$

$$\textbf{Discipline 2: } y_2(x_{0,1}, x_{0,2}, x_2, y_1) = |y_1| + x_{0,1} + x_{0,2} - x_2 \tag{3.6.4}$$

The unique MDF solution of the block Sellar problem Equation (3.6.2), outside of the introduction of the block x_2 , stays unchanged. This is illustrated in Table 3.1.

The coupling solution and the total derivatives with respect to all the design variables (x_0, x_1, x_2) , near the MDF solution, for the two-block Sellar are detailed within Appendix B.2.1

	Sellar	Block Sellar
Objective	$f^* = 3.1833$	$f^* = 3.1833$
Shared variables	$[x_{0,1}^*, x_{0,2}^*] = [0.0, 1.977639]$	$[x_{0,1}^*, x_{0,2}^*] = [0.0, 1.977639]$
Local variables	$x_1^* = 0$	$[x_1^*, x_2^*] = [0, 0]$
Coupling variables	$[y_1^*, y_2^*] = [0.8, 1.8]$	$[y_1^*, y_2^*] = [0.8, 1.8]$

Table 3.1: Optimal design for Sellar and Block Sellar

We now propose a new extension of the aforementioned problem by adding an additional parameter that controls the size of all the local variables, the coupling variables, and the constraints. Each local variable, constraint and coupling variable are vectorized to be now considered as an n -dimensional vector. This modified n -dimensional block Sellar problem is described by the following set of equations:

$$\begin{aligned}
\text{minimize } f(x_{0,1}, x_{0,2}, x_1, x_2, y_1, y_2) &= \frac{\|x_1\|^2}{n} + \frac{\|x_2\|^2}{n} + x_{0,2} + \frac{\|y_1\|^2}{n} + e^{-\bar{y}_2} \\
&\text{with respect to } x_{0,1}, x_{0,2}, x_1, x_2, y_1, y_2 \\
&\text{subject to :} \\
&-10 \leq x_{0,1} \leq 10 \\
&0 \leq x_{0,2} \leq 10 \\
&0 \leq x_{1,i} \leq 10 \quad i = 1, \dots, n \\
&0 \leq x_{2,i} \leq 10 \quad i = 1, \dots, n \\
&c_{1,i}(y_{1,i}) = 3.16 - y_{1,i}^2 \leq 0 \quad i = 1, \dots, n \\
&c_{2,i}(y_{2,i}) = y_{2,i} - 24 \leq 0 \quad i = 1, \dots, n \\
&y_{1,i} - y_{1,i}(x_{0,1}, x_{0,2}, x_{1,i}, y_{2,i}) = 0 \quad i = 1, \dots, n \\
&y_{2,i} - y_{2,i}(x_{0,1}, x_{0,2}, x_{2,i}, y_{1,i}) = 0 \quad i = 1, \dots, n
\end{aligned} \tag{3.6.5}$$

$$\textbf{Discipline 1: } y_{1,i}(x_{0,1}, x_{0,2}, x_{1,i}, y_{2,i}) = \sqrt{(x_{0,1})^2 + x_{1,i} + x_{0,2} - y_{2,i}} \tag{3.6.6}$$

$$\textbf{Discipline 2: } y_{2,i}(x_{0,1}, x_{0,2}, x_{2,i}, y_{1,i}) = |y_{1,i}| + x_{0,1} + x_{0,2} - x_{2,i} \tag{3.6.7}$$

This transformation preserves the global minimum of the one-dimensional version (in \mathbb{R}^{2n} for the local variables) and the optimal objective value ($f^* = 3.1833$) and constraints. The global optimal design for the MDF architecture is obtained in $x_0^* = [0.0, 1.977639]$ and $x_1^* = x_2^* = 0_{\mathbb{R}^n}$.

A scalable study

In consideration of the aforementioned hypothesis, at the shared variables' optimal value x_0^* , the MDF solution for \mathcal{P}_{low} is well-obtained at $x^* = [x_1^*, x_2^*]$. This solution is an over-constrained optimum, as all lower bounds are active, as well as the constraint c_1 . Consequently the LICQ is not verified. Nevertheless, the MFCQ is satisfied and a Lagrange multiplier vector that verifies both the first- and the second-order KKT necessary conditions for (\mathcal{P}_{low}) can be readily identified. Both blocks verify the Unique

Block Minimizer assumption in the vicinity of x_0^* . Moreover, the solution yielded by each block corresponds to the coupled optimal primal-dual pair of MDF, thereby validating the **First Order Separability** assumption.

Figure 3.13 depicts the average total number of disciplines required to reach the final target, which is fixed at 3.19 with $n \in [1, 5, 20, 50]$. The true optimal value is found to be at $f^* = 3.1833$. It should be noted that the original Sellar problem has two coupling variables, two local variables, and two constraints. Consequently, for a specific value of n , the abscissa represents the total number of local variables, coupling variables, and constraints that are all equal to $2 * n$. A total of 150 distinct starting points were utilized in the testing process.

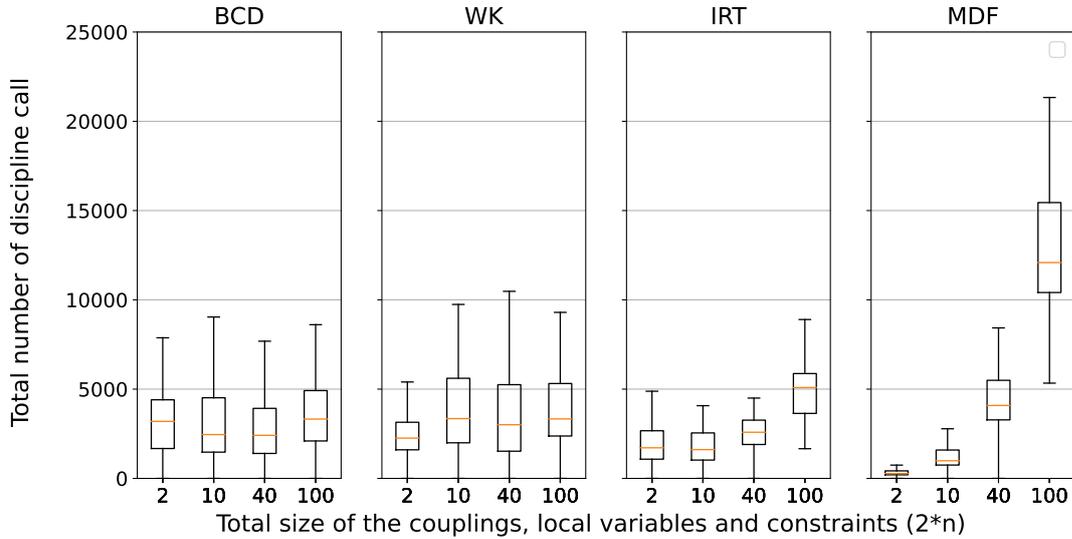


Figure 3.13: Scalability study on Sellar

The results indicate that the average number of disciplines executed to reach the final target remains relatively consistent for BCD-based architectures. This is attributed to the relatively low dimensions of the sub-problems addressed with gradient-based optimizers. Conversely, the MDF architecture exhibits the expected curse of dimensionality for gradient-free approaches. This results in a rapid increase in computational cost with the problem size.

The variance for BCD-based architectures is significantly higher than that of the two others in the number of discipline executions for low-dimensional problems ($2 * n \leq 40$). These variations are attributed to the overall BCD-loop convergence, which could be enhanced through further work on the algorithms. For higher-dimensional problems, exceeding 40 variables, the trend is reversing between MDF and BL-BCD-MDF architectures.

In general, bi-level architectures are more efficient than gradient-free MDF in terms of both mean and variance of CPU cost when the number of local design variables exceeds 40, which is always the case for the type of applications targeted (e.g. aircraft aerostructure optimization).

Chapter 4

A multi-fidelity framework

Summary - This chapter covers topics related to extending multi-fidelity methods, which aim to obtain a high-precision (or high-fidelity) solution to a problem by taking advantage of other models of lower precision but lower computational cost, commonly known as low-fidelity models, to multidisciplinary design optimization (MDO). We address one of the distinctive characteristic of the composite nature of MDO problems: given a large number of disciplines, each with several levels of fidelity, we face a combinatorial explosion of the number of possible fidelity models, and the interest of integrating each of them into the multi-fidelity process is not known *a priori*, in particular due to the presence of coupling variables. The need then arises to roughly sort this very large number of models in terms of cost and precision in order to select the most relevant models for integration into a multi-fidelity process. In this context, two sensitivity criteria, one based on coupled adjoint and the other on gradient alignment, are proposed for the rapid classification of low-fidelity models. These criteria are tested on simple cases, demonstrating their ability to efficiently approximate the error of these models. A multi-fidelity refinement approach is used to compare the impact of different model choices on the convergence speed, showing that the presented criteria offer an interesting alternative to more precise but also more expensive methods, while still providing an interesting acceleration of the considered multi-fidelity approach. In addition, a small application is made to illustrate the capabilities of the implementation within GEMSEO and how the presented multi-fidelity methodologies can be intertwined with the bi-level architecture presented in Chapter 3.

Résumé - Ce chapitre couvre les sujets liés à l'extension des méthodes multifidélité, qui visent à obtenir une solution de haute précision (ou haute fidélité) à un problème en tirant parti d'autres modèles de moindre précision, mais de moindre coût de calcul, communément appelés modèles de basse fidélité, à l'optimisation multidisciplinaire (MDO). Nous abordons l'une des caractéristiques distinctives de la MDO, qui est liée à la nature composite des problèmes traités : étant donné un grand nombre de disciplines, chacune avec plusieurs niveaux de fidélité, nous sommes confrontés à une explosion combinatoire du nombre de modèles de fidélité possibles et l'intérêt d'intégrer chacun d'entre eux dans le processus multifidélité n'est pas connu a priori, en particulier en raison de la présence de variables de couplage. Il est donc nécessaire de trier grossièrement ce très grand nombre de modèles en termes de coût et de précision afin de sélectionner les modèles les plus pertinents pour l'intégration dans un processus multifidélité. Dans ce contexte, deux critères de sensibilité, l'un basé sur l'adjoint couplé et l'autre sur l'alignement du gradient, sont proposés pour la classification rapide des modèles de basses fidélité. Ces critères sont testés sur des cas simples, démontrant leur capacité à approximer efficacement l'erreur de ces modèles. Une approche de raffinement multi-fidélité est utilisée pour comparer l'impact des différents choix de modèles sur la vitesse de convergence. Elle montre que les critères présentés offrent une alternative intéressante à des méthodes plus précises, mais aussi plus coûteuses, tout en fournissant une accélération intéressante de l'approche multi-fidélité considérée. En outre, une petite application est présentée pour illustrer les capacités de l'implémentation dans GEMSEO, ainsi que la manière dont les méthodologies multi-fidélité peuvent être imbriquées aux architectures à bi-niveaux présentées dans le précédent chapitre.

4.1 Introduction

As models involving physics become more accurate in their predictions and results, they tend to become increasingly complex. This directly leads to a significant increase in the computational cost of these models, which may prevent numerical experiments that rely heavily on their use. As MDO typically involves the integration of multiple coupled physical phenomena, for instance with the integration of CFD/CSM solvers in the optimization process, it is particularly sensitive to these limitations.

As these models, which have been qualified as high-fidelity models, are still necessary to perform high-quality optimization, research have been conducted on strategies to reduce the computational cost while taking advantage of these models' high accuracy. Multi-fidelity approaches [46, 116] represent a set of methodologies designed to enhance the efficiency and the precision of high-computational numerical problems. These methods rely on the incorporation of one or multiple low-fidelity models, of varying cost and approximation qualities, that emulates the high-fidelity model with much less time, but at the price of a lower accuracy. The objective is to achieve the same outcomes as if only the high-fidelity model were employed, while reducing the reliance on high-fidelity executions, typically by reducing the number of high-fidelity model calls.

Following the general success of multi-fidelity in mono-disciplinary optimization, it is reasonable to expect that the extension of such approaches to multidisciplinary design optimization could be beneficial. Initial attempts combine different modeling levels [115, 135, 137, 136, 140, 161], such as performing CFD and CSM solvers for both the aerodynamics and structures with analytical models for other physics/components. This is typically done in overall aircraft design [161]. Several classical multi-fidelity approaches can also be directly applied within a MDO framework with few adaption from the mono-disciplinary case [161].

However, despite recent efforts in this direction, the multi-fidelity MDO literature remains sparse. In particular, most of the proposed methodologies, even if often successful, typically do not take into account MDO-specificities. This indicates that further MDO-specific improvements can be expected for generic multi-fidelity frameworks. Some promising works that do address these special features do exist. Some of these acknowledge the discipline decomposition, as evidenced in [6, 29, 54] where Bayesian approximation of the error induced by each disciplinary fidelity level are constructed. This allows for the dynamic increase of the fidelity level of the discipline that contributed most to the error. Another example is provided by *Wang and al.* [156], who propose a multi-fidelity framework that considers a change in the overall MDO architecture as a change in fidelity and provides a suitable switching criterion.

In particular, this chapter focuses on one of the MDO-specific features when considering multi-fidelity approaches: the combinatorial explosion of the number of possible fidelity models. By considering multiple disciplinary models and several fidelity levels for each one of them, it is possible to create combinations of these disciplinary fidelity models. Each combination yields a new fidelity level of the overall optimization problem. As all these models are coupled models of different disciplinary fidelity models, it is not known, *a priori*, which coupled fidelity models should be incorporated within a multi-fidelity framework to maximize its efficiency.

This observation calls for dedicated methodologies for assessing the suitability of a specific

fidelity model in relation to the other fidelity models. In the traditional approach, each fidelity model is evaluated on two criteria: the cost of execution and its accuracy relative to the high-fidelity model. This naturally leads to the construction of the well-known (*cost, error*) Pareto front, which characterizes for each fidelity model if it is Pareto dominant or not. A Pareto-dominant model is of interest as it is guaranteed to be at least as good as any other model, in terms of both speed and accuracy.

However, considering the large amount of fidelity models, the construction of the said Pareto front is challenging, especially as the estimation of the error for every fidelity models is costly. A direct consequence is that strategies that heavily rely on statistics to discriminate fidelity models may meet their limits. To the best of our knowledge, there exist two already existing works that tackle this specific challenge within a MDO context.

The first one is the work by *Charayron and al.* [28] in which a multi-fidelity Bayesian optimization process is considered. To rank the fidelity models emerging from the aforementioned combination of disciplinary fidelity models, a (*cost, error*) Pareto front is constructed. Both criteria are estimated by evaluating each model on a set of sampling points issued from a unique design of experiments. The mean cost is then computed, as well as a criterion measuring and aggregating the mean error on both the objective and constraints. The selection of fidelity models from this Pareto front yielded promising outcomes in the optimization of a drone design comprising 19 design variables, with 5 fidelity levels and 100 sampling points for the design of experiments. These interesting results, however, are unlikely to be transposable to our targeted applications, which encompass several thousand design variables and non-linear constraints, as well as more than one hundred fidelity models. In particular, we cannot rely on a design of experiments for the ranking of the different fidelity levels.

The second work, which is more closely related to our case of application, is the one by *Wu and al.* [158] that employs a comparable approach to fidelity ranking within a gradient-based sequential multi-fidelity framework. The approach presented by the authors addresses the potential combinatorial explosion through the use of two distinct steps:

- The first step involves the construction of multiple (*cost, error*) Pareto fronts, one for each discipline and for each considered output. For each discipline, each fidelity model is executed once on a specific design point, and the outputs of interest, such as coupling variables or constraints, are compared to the corresponding high-fidelity outputs. Any fidelity model that produces a non-Pareto optimal output is excluded from the subsequent multi-fidelity optimization process. The multi-fidelity method is then initiated with the combination of all the lowest fidelity models.
- The second step happens at the end of each fidelity optimization. Once a switching criterion, based on the first-order KKT conditions, is met, another combination of disciplinary fidelity models must be selected for the subsequent optimization. To accomplish this, an error propagation is performed on the current solution to identify the next combination with the highest normalized error reduction. Each disciplinary output is considered a multivariate normal random variable, with each output of a single discipline being considered highly correlated. However, the correlation between different disciplines, as implied by the couplings, is not considered. Given

that the considered system functions of interest are analytic and relatively simple, the Monte Carlo method is employed to perform the error propagation, with an average of 10^6 models executions for the considered application. All errors for a specific combination are then aggregated together using the Lagrangian multipliers as weights for the constraints errors. The error reduction is normalized by the cost of the fidelity combination. This process is repeated until the combination of all high-fidelity models is considered and fully optimized.

The presented methodology addresses the same typology of problems that we aim to solve with multi-fidelity approaches, making it of great interest. Our position on the presented work is that the first step might have deleted interesting disciplinary fidelity models, which are suboptimal when considered on their own, but might also be useful when combined with other disciplinary fidelity models. Furthermore, we do the hypothesis that if the initial ranking of fidelity models is sufficiently accurate, typically through a more exhaustive analysis than that proposed in [158], the subsequent optimization process may be avoided, thereby conserving significant computational resources.

This chapter is thus devoted to a methodology for approximating the quality of a large number of fidelity levels within the context of MDO, while avoiding models intrusiveness. The results are expressed in the form of two sensitivity criteria, which provide an approximation of the error induced by each model. These criteria were designed to be directly applicable to generic multi-fidelity approaches. In particular, the criteria were developed with the intention of facilitating the down-selection of the most suitable fidelity models for use within the multi-fidelity refinement framework proposed in [108]. Given the considerable number of models that require ranking, it is not an option to employ highly computationally intensive methods. Instead engineering tools and methodologies based on local information, meaning that they are computed at a single design point, are considered. This implies that these two criteria will undoubtedly yield an imperfect ranking of the different fidelity combinations. Consequently, non-dominant models can be selected for inclusion in the multi-fidelity framework. This is a price we are willing to pay for several reasons. First, given the number of models and their potentially high cost of execution, an exact ranking of all fidelity models would be more costly than running the high-fidelity model as a standalone. Second, global approximations of model errors could lead to penalization of overall poor but locally interesting models. Finally, despite the approximations, the resulting ranking is expected to preserve the overall model hierarchy. This implies that two fidelity models will have the same relative ranking according to both the criteria and the true classification. In particular, since we intend to rely on the refinement approach, convergence to the high-fidelity optimum of the MDO problem is guaranteed as soon as the combination of all high-fidelity models is integrated into the multi-fidelity refinement framework.

The chapter is structured as follows: first, a review of the multi-fidelity approaches is provided, with a focus on numerical optimization. The existing multi-fidelity MDO framework and applications are presented at this point, followed by the observation that the number of fidelity models and their selection is rarely addressed. Next, the theoretical framework is presented, describing the methodology and the need for a fast approximation of the error associated with a certain level of fidelity. Two criteria are presented as a response to the need of a fast ranking of numerous fidelity models, both of which are designed to be low-cost and mostly non-intrusive estimators, as derivatives, or

at least an approximation, are required. Both criteria are then tested on small test cases, among them a perturbed Sellar [135] problem, an ordering problem on SSBJ [139] and a multi-point profile optimization problem are considered. For all of these applications, different model selections are considered and compared, in terms of convergence speed, with the multi-fidelity refinement approach proposed in [108]

4.2 State of the art: multi-fidelity in MDO

4.2.1 The multi-fidelity approaches

Multi-fidelity approaches rely on the utilization of a combination of a reference model, the so-called high-fidelity (HF) model, and a set of low-fidelity (LF) models. The former is considered to be sufficiently precise for the targeted application but also too costly, in terms of both the computation time and the necessary computational resources, to be executed too many times. The latter, the so-called low-fidelity models, are far less expensive than the high-fidelity one, but also less accurate. Multi-fidelity approaches are outer-loop strategies that incorporate both type of models in order to identify a solution of the high-fidelity model while reducing the restitution time of the overall numerical experiment. The distinction between single- and multi-fidelity outer-loop applications is illustrated in Figure 4.1.

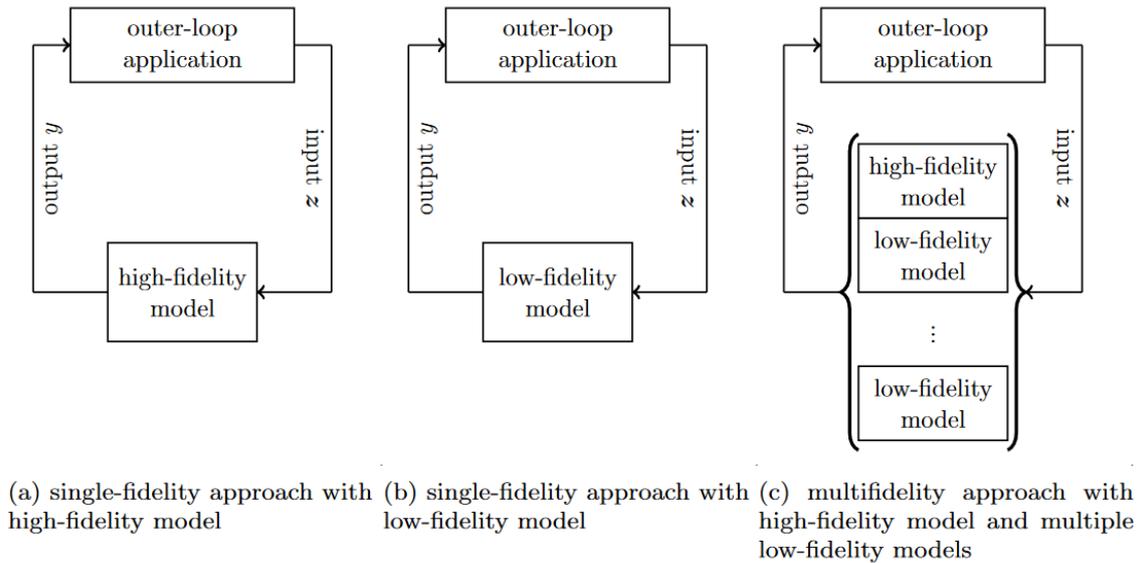


Figure 4.1: From single fidelity towards multi-fidelity approaches [116]

A substantial corpus of literature exists on multi-fidelity approaches. Two surveys, among others, namely [116] and [46], provide an overview of such methods across several numerical domains. In particular, [116] describes multi-fidelity approaches as outer-loops using a HF model and one or more LF models. These approaches have the particular interest of regrouping major areas of research in numerical science, namely inference, uncertainty quantification, and optimization. Furthermore, they determine a useful categorization of multi-fidelity approaches, which we will summarize right after. In contrast, [46] focuses

on surrogate modeling, i.e. substitution models based on sampling points of an original model, and in particular multi-fidelity surrogates models, which are surrogates built on samples coming from both HF and LF models, and their construction.

This section presents a summary of the main concepts and categorizations presented in [116]. Readers are encouraged to consult the original paper for a more detailed and exhaustive review.

Model Management

As defined by the survey conducted in [116], a multi-fidelity method is primarily defined by the model management considered. Model management is the process of defining the manner in which the various models interact with one another, it encompasses the determination of which models are executed, when they are executed, and which data are exchanged. It must balance model evaluations among the models (i.e. to decide which model to evaluate when) and must guarantee the same accuracy in the outer-loop result as if only the high-fidelity model were used. Three methods of model management, illustrated in Figure 4.2, are described by the survey conducted in [116]:

- **The adaptation method:** entails the updating of low-fidelity models with information from the high-fidelity model as the computation proceeds. The Efficient Global Optimization (EGO) algorithm [75] is a well-known example of this approach. This model management adaptively constructs a low-fidelity model by interpolating the objective function corresponding to the high-fidelity model with Gaussian process regression. Another example is the model correction approach [76].
- **The fusion method:** entails the evaluation of high and low-fidelity models, after which the information from all outputs is combined. For instance, the control variate frameworks [69], in which low-fidelity models serve as control variates for the estimation of statistics on random variables of the high-fidelity model, are part of this paradigm. Another example could be the co-kriging techniques [31, 50, 98], that extend the classical kriging techniques to the multivariate case by relying on the variance and covariance of several models. For example, in numerical optimization when the high-fidelity derivatives are accessible, it is common to learn both the output of interest and its derivative [49, 50].
- **The filtering method:** entails invoking the high-fidelity model following subsequent to the evaluation of the low-fidelity models. This may entail evaluating the high-fidelity model exclusively if the low-fidelity model is deemed inaccurate, or if the candidate point meets certain criteria based on the low-fidelity evaluation. Good representatives of this category are the refinement approach [108] and the Trust-Region Model Management (TRMM) algorithm [5]. The refinement approach employs an optimization process at each fidelity level, from the lowest to the highest. The objective is to identify a more optimal starting point for the subsequent level, thereby reducing the reliance on that level. The Trust-Region Model Management (TRMM) algorithm represents an adaptation of the classical trust-region SQP algorithm, wherein the local quadratic model is replaced by a generic low-fidelity model. Both of these methods are subsequently elucidated in greater detail.

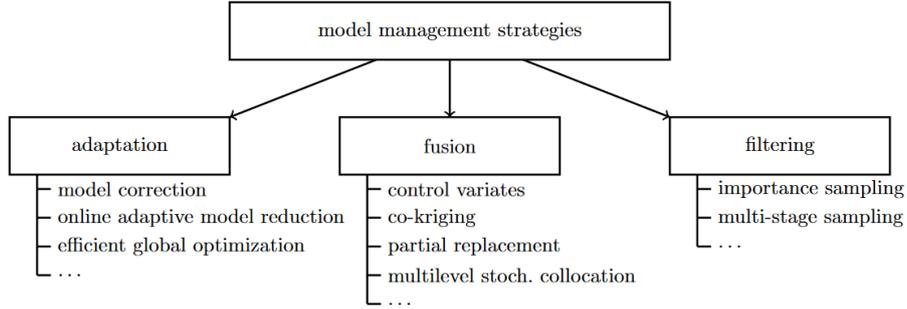


Figure 4.2: A categorization of MF model management [116]

Multi-fidelity approaches also encompass multilevel methods, such as multi-grid methods [65], which describe a natural hierarchy between fidelity models by varying some parameters. Multi-fidelity methods are more generic in that they do not necessarily rely on models that can be generated on request by varying a parameter, such as a discretization. The quality and costs of these models are typically determined by already-known rates, which are not necessarily accessible in more generic multi-fidelity processes.

Model management techniques studied in [46] highly rely on Multi-fidelity surrogate models, also known as data-fit models, and are consequently classified as belonging to the fusion category in contrast to the definition of multi-fidelity hierarchical models where models are invoked on some criterion and no surrogates are built on several fidelity models. The latter category encompasses both the adaptation and filtering methods. Additionally, in this same review, it is posited that multi-fidelity arises only when at least two physical models are involved in the process. Consequently, well-known methods, such as The Efficient Global Optimization (EGO) algorithm [75] (cf. Section 4.2.2), which employs a single physics-based model for optimization, are not regarded as a multi-fidelity approach in the sens of [46].

Methods that merely replace the high-fidelity model with a low-fidelity model, such as model reduction, are rejected by both surveys [46, 116] as multi-fidelity approaches. This is because they lower the accuracy of the solution, by typically providing error bounds, when multi-fidelity methods should establish accuracy and convergence guarantees.

Low-fidelity models

In addition to model managements, multi-fidelity approaches also rely on one or several low-fidelity models. [116] describes three primary sources of low-fidelity models :

- **Simplified models** : are created by applying expertise and in-depth knowledge to create a simpler model. This can be done by defining an early stopping criterion or a natural problem hierarchy such as RANS and Euler equations [70], by using linear approximations of the considered models [117], or even by using coarser meshes [85].
- **Projection based reduced models** : utilize the mathematical structure of the problem, such as a low-dimensional subspace. Proper Orthogonal Decomposition (POD), Krylov subspace, or reduced basis methods are representative of this category. See [14] for a detailed survey.
- **Data-fit models** : a sample of inputs and outputs is employed to generate an

approximation of the model. These models, also known as response surface models or surrogates models, can be further delineated into several categories, including interpolation and regression models, Kriging/Cokriging [50, 98], Support Vector Machine (SVM) [68, 107] and Radial Basis Functions (RBF) [49].

The construction of low-fidelity models is contingent upon the nature of the problem. In some instances, the construction is straightforward and intuitive, such as a coarser grid for PDE resolution or Krylov subspaces. However, in other cases, a data-fit method is necessary to build low-fidelity models with sufficient accuracy. For non-linear models, Radial Basis Functions (RBF) are a popular choice due to their simplicity, while Kriging and Cokriging [50, 98] as Bayesian estimators are more accurate but require a good a priori estimation to be effective [31].

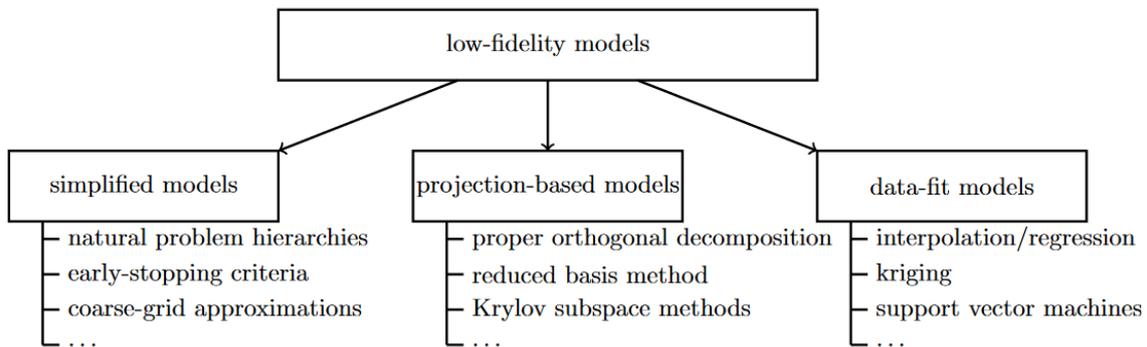


Figure 4.3: Categorization of Low-fidelity models [116]

4.2.2 Multi-fidelity methods in numerical optimization

One of the main areas where multi-fidelity approaches are particularly effective is numerical optimization. This is particularly true for high-dimensional problems but also when computationally intensive PDE solvers are necessary, in CFD or structure, for instance. A multi-fidelity method is deemed to be efficient if it is capable of converging to the HF solution with the same degree of accuracy that would be achieved by employing solely the high-fidelity model, while simultaneously reducing the CPU time required (typically by lowering the number of high-fidelity model evaluations).

As in classical optimization, in which there is a unique fidelity level, multi-fidelity optimization approaches can be categorized into global or local methods.

Global methods are employed to identify the optimal design within the entire design space. As they do not typically necessitate the gradient and frequently operate with black-box models, Bayesian optimization techniques are frequently utilized for the construction of surrogate models. These methods seek a minimum with respect to an adaptive refinement of a low-fidelity model throughout the optimization search, utilizing information from high-fidelity model evaluations. The Efficient Global Optimization (EGO) algorithm [75] is based on this principle: during the optimization process, it builds a kriging model of the objective, and the next sample point is selected by maximizing an infill criterion, such as the expected improvement function, as shown in Figure 4.4. This approach is the most popular when searching for a global multi-fidelity optimization algorithm. EGO can be made globally convergent and does not require high-fidelity model derivatives [74].

However, EGO is sensitive to the initial points used to build the kriging model [74, 96]. This implies that a fairly exhaustive search around the initial points might be necessary before a more global search can begin. Building a kriging model can therefore be time consuming, depending on the model dimension, and may be prohibitive in practice.

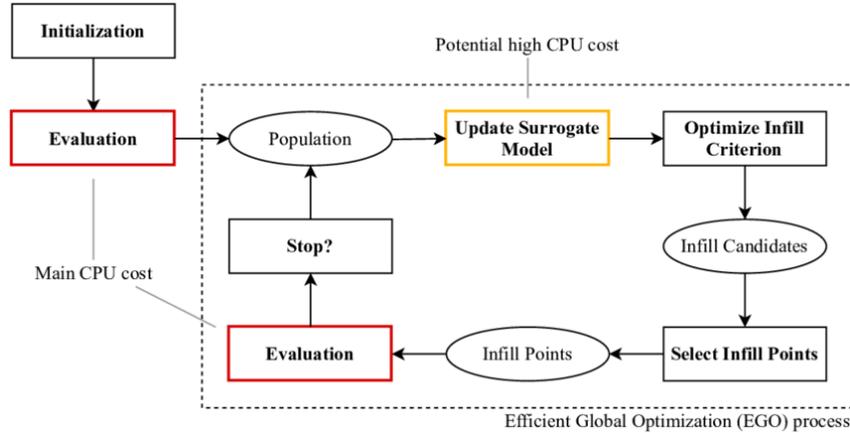


Figure 4.4: EGO optimization workflow [25]

Variants of EGO exist, including a multi-objective version [126], and an EGO algorithm that fuses information from several kriging models [6]. Other global methods include gaussian process-based optimization [60] and multi-fidelity optimization methods based on pattern search [124].

Local methods, in contrast, seek solutions that are locally optimal, i.e. the best point in a restricted part of the design space. Consequently they may yield to a worse solution than their global counterparts when the considered problem has multiple local minima and they generally rely heavily on gradient information. However, these methods typically require far fewer model evaluations than the global ones, which makes them increasingly attractive as the dimensions of the optimization problem increase. Indeed, when considering more than a thousand design variables and non-linear constraints, most derivative-free algorithms will fail, and having access to the derivatives typically accelerates computation time significantly [34, 145]. They also handle non-linear constraints much better than heuristics [99].

Among the set of local multi-fidelity methods, the Trust-Region Model Management (TRMM), introduced in [5], is one of the most frequently utilized. This is an enhanced version of the Trust Region (TR) method that can use generic low-fidelity models instead of the classical quadratic model. In practice, each iterate is chosen after the optimization of an approximation model with a particular trust region radius. The overall workflow is summed up by Figure 4.5. The primary distinction between the classical Trust Region algorithm (which employs a quadratic approximation) and the TRMM lies in the fact that the accuracy of the solution is no longer solely contingent upon the trust region radius. Instead, it is also influenced by the fidelity level utilized for the approximation. If the optimization was successful and the predicted improvement is considered sufficiently accurate, the trust region radius can be increased or the fidelity level of the approximation model can be decreased to accelerate convergence. In the event that the accuracy of the approximation is deemed insufficient, the fidelity level can be increased or the radius can be decreased.

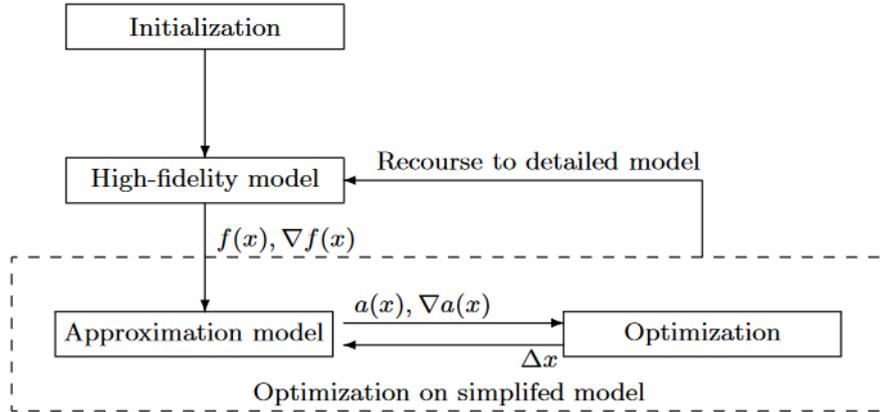


Figure 4.5: TRMM outer loop [5]

It has been demonstrated that the method is globally convergent towards a local minimum of the HF model under the simple assumption that the low-fidelity model satisfies first-order consistency at each HF evaluated point [4]. This assumption is not difficult to obtain in practice, as it is based on the equality of the objective function and its gradient of the low-fidelity model with that of the high-fidelity model at the center of the trust region. In the case of an unconstrained optimization problem, the zero-order consistency, equality of the objective functions at the center of the trust region, may be sufficient to ensure convergence [43].

There are several variants of the TRMM algorithm. [35] and [45] designed a multi-objective version, [4] modifies TRMM to include strategies from other optimization methods (Lagrangian, Sequential Quadratic Programming (SQP)), while [129] extends the TRMM to handle variable parametrization problems (multi-fidelity problems where each model has a different parametrization). A recursive version (with an arbitrary number of fidelity levels), introduced in [61] as the Recursive Multiscale Trust-Region (RMTR) method, has been used for Adjoint-Based CFD applications in [109] in conjunction with an asynchronous validation criterion. This framework permits recursive infinity norm trust region optimization, allowing the approximation model to be either a quadratic model or a already specified lower-fidelity model with a possible variable parametrization through full rank linear prolongation and restriction. As gradients are available for all the considered models, a first-order criterion based on gradient alignment is employed. The asynchronous validation, illustrated in Figure 4.6, permits optimization to continue while ensuring the validity of a given design point. For the considered Adjoint-Based CFD applications the results demonstrate that this trust region approach exhibits comparable effectiveness to a purely hierarchical multilevel refinement approach [109]. The latter approach, which optimizes successively from the lowest to the highest fidelity level, is further considered and investigated in [108].

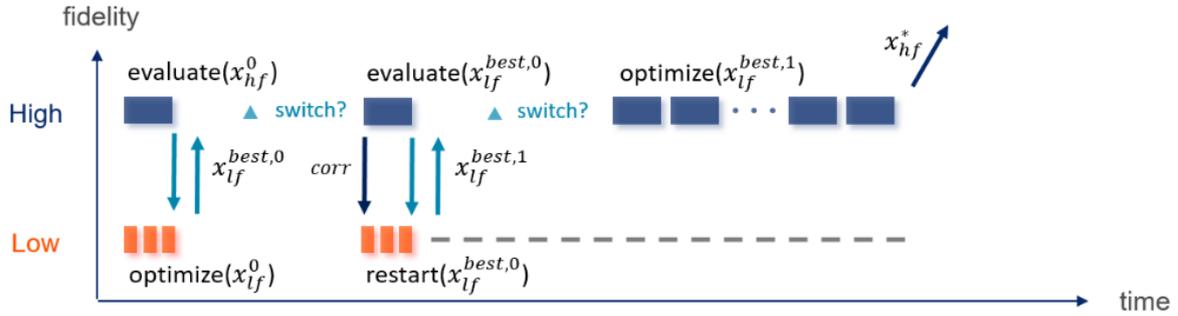


Figure 4.6: Asynchronous validation for recursive TRMM, figure from [109]

[98] combines both approaches to develop a provably convergent multi-fidelity optimization method that employs Cokriging Bayesian model calibration and first-order consistent trust regions. In the unconstrained case, without high-fidelity derivatives, [97] demonstrates that the same results can be achieved despite the high number of function evaluations required given the lack of gradient information.

MDO problems are typically complex and time-consuming to solve, which makes the multi-fidelity approaches promising. However, although the multi-fidelity domain is widely studied, the previous results are presented in the context of the classic monodisciplinary problem and rarely explore the specific aspects of MDO. This is despite the fact that there is a wealth of potential in exploiting MDO architectures, disciplinary separation or the handling of coupling variables. The subsequent section is devoted to an examination of MDO-specific features and their exploitation in the construction of multi-fidelity processes.

4.2.3 Multi-fidelity applied to MDO

The preceding works were general multi-fidelity approaches designed for general monodisciplinary optimization problems. While these references are directly applicable to MDO (e.g. at the disciplinary level), they do not address MDO-specific multi-fidelity strategies that can be developed. In the following section, we present work that takes into account the organization of the disciplines, the coupling variables, and the different MDO architectures that have been developed to date.

The MDO community initially attempted to employ Surrogates-Based Optimization (SBO) methods to address MDO problems. The fundamental concept is to substitute high-fidelity models and costly components to surrogate models. This was the case for the majority of distributed architectures, as the multitude of models involved and the substantial volume of information exchanges render the replacement of costly models or those that are intensively executed a necessity. For instance, [135] and [137] utilize response surface-based models for CSSO (Concurrent Subspace Optimization), which is summarized in Figure 4.7, and CO (Collaborative Optimization) architectures respectively. BLISS-2000 [140] follows this philosophy of design by nature, as every communications between system and disciplinary sub-problems are made through a set of response surface-based models that approximate their optimized coupling value with respect to a block of local variable. Those three distributed architecture are discussed in more detail in Section 3.2.1. Some approaches address monolithic architectures by employing surrogates and comparing them.

Among them, there is the MDF architecture where costly components such as MDA are replaced using Kriging, Gaussian processes, neural networks or polynomial response surfaces [136, 115]. Most of these approaches rely on surrogates that must be kept up to date during optimization with high-fidelity model evaluations. For these reasons, the creation and updating of the surrogate must be integrated into the optimization framework, which adds cost during optimization and organizational issues in enforcing the training method. As an answer, [48] proposed to use so-called "one-shot" surrogates, which are built beforehand within their adjoint based optimization framework. Since they are not kept up to date, these surrogates are less reliable, but since they are built before optimization begins, the training cost is already paid. It also allows for more modular approaches where any surrogate of interest can be plugged directly into the optimization framework, which is very valuable when considering distributed architectures.

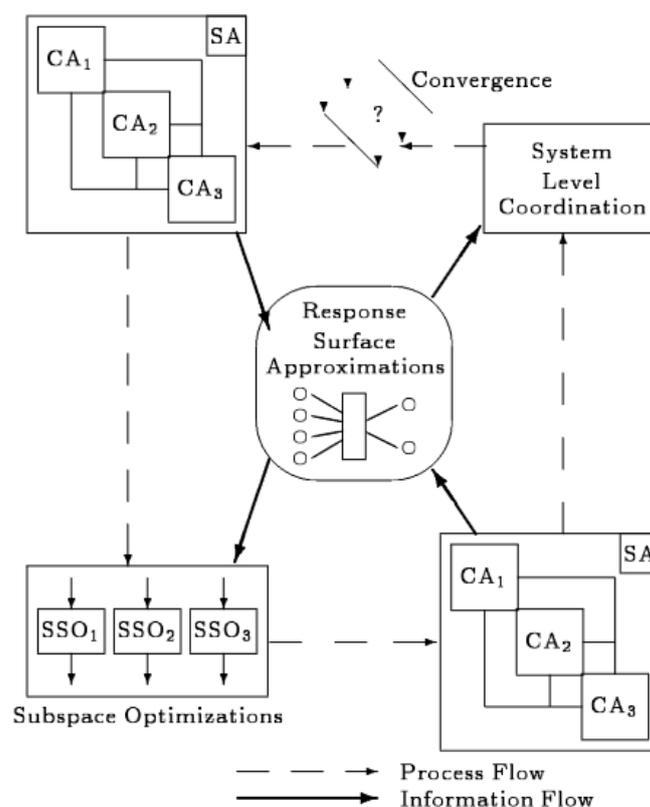


Figure 4.7: CSSO framework using a neural network based surrogate for couplings [135]

The aforementioned SBO approaches demonstrate the potential for low-fidelity models to be integrated into complex MDO processes, thereby reducing the time required for computation. However, as they do not align with the aforementioned multi-fidelity methods, they will not be further discussed. Alternatively, they may be regarded as analogous to SBO methods, in that they entail the substitution of models with a meticulously selected approximation. Moreover, these methods are highly reliant on surrogate models [161] to serve as proxy and merely replacing costly components. This is not an ideal choice in our case as their construction is often complex and generally require high expertise and prior knowledge on the original model subject to the approximation [49, 50, 6, 161]. In particular, this is all the more true for high-fidelity and high-dimensional models, as all surrogates

are subject to the curse of dimensionality to some extent, despite recent advances in this area [7]. Therefore, for the targeted applications, such as the pylon optimization in [55] where 840 design variables and 200 000 nonlinear constraints are considered, building surrogates may not be the best choice, simplified models, for instance using an early stopping criterion or a change in the architecture, will be preferred instead.

The literature on more in-depth multi-fidelity frameworks that address multiple fidelity levels and model management is relatively scarce, with few examples emerging lately. Bayesian approaches are a popular choice [46]. [6] proposed a probabilistic quantification of the model discrepancy, which they utilized to solve a MDO problem with fixed fidelity levels for each model. At a certain point in the optimization process, an estimated optimal design is evaluated in terms of its performance and the variance of its constraints. If the variance is deemed to be excessive, further investigation is conducted using Sobol indices, which reveal the models with the greatest influence on the error. These models are then replaced by the next fidelity level. In order to improve the method, the approach is extended by [29] to encompass interdisciplinary couplings. However this approach only considers feedforward effects. More recently, [54] employed Cokriging by Linear Model of Coregionalisation (LMC) in order to address feedback effects and reduce uncertainties in the coupling variable. However, these approaches do not account for the computational cost of each fidelity, nor do consider the effect of discipline errors on constraints.

[156] presents a multi-fidelity MDO framework with a switching mechanism that enables the fidelity level of the model to be incrementally increased during optimization. This filtering framework has the distinctive feature of allowing both the models and the MDO architecture to be updated from the lowest fidelity level to the highest whenever a criterion is met as illustrated in Figure 4.8. The switching mechanism is derived from the adaptive model switching (AMS) [106] approach, which determines whether the uncertainty associated with the current model output outweighs the latest improvement in the relative fitness function. The proposed variant of AMS no longer necessitates a population-based algorithm and instead generates a distribution of the relative fitness function improvements throughout the optimization iterations. However, the methodology is highly reliant on surrogate modeling and error quantification, and it is illustrated on a Battery Thermal Management System (BTMS) [155] with only 4 disciplines and 17 optimization variables. Consequently, it is unlikely that such results can be extended to large-scale problems.

Some rare works (See e.g. [93, 96]) employ multi-fidelity approaches in a distributed context. For instance, [93] presents a CO architecture that utilizes a corrected low-fidelity model at the disciplinary level. This low-fidelity model is computed offline and is an aggregation of low- and high-fidelity level simulations. The exploitation of parallelism in conjunction with low-fidelity models can result in a significant reduction in restitution time. However, due to the distributed architecture, which is known for exhibiting poor or non-existent convergence properties, these methods must be employed with caution, as they rarely reach the HF optimum.

[122] proposes a novel approach to the MDO organization of an aero-structure problem. Their hypothesis is that the optimization problem can be simplified by focusing on the structure discipline alone. To this end they introduced an Individual Disciplinary Optimization (IDO) process, where only the structure runs high-fidelity simulations. The aerodynamic discipline is replaced by a low-fidelity version obtained by a combination of

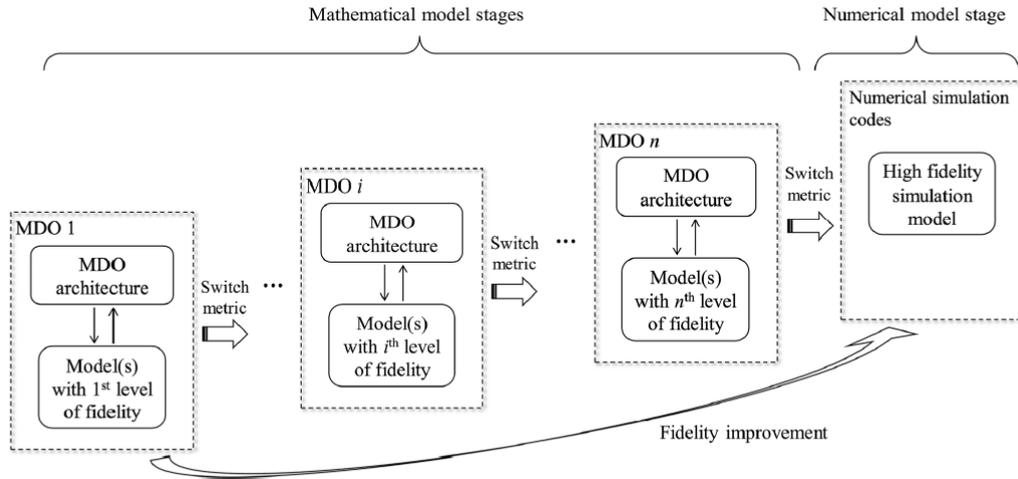


Fig. 1 Variable fidelity-based MDO architecture

Figure 4.8: Illustration of the Filtering Model Management [156]

Kriging and POD. The said process is shown in Figure 4.9. It should be noted that the authors do not consider multiple low-fidelity models.

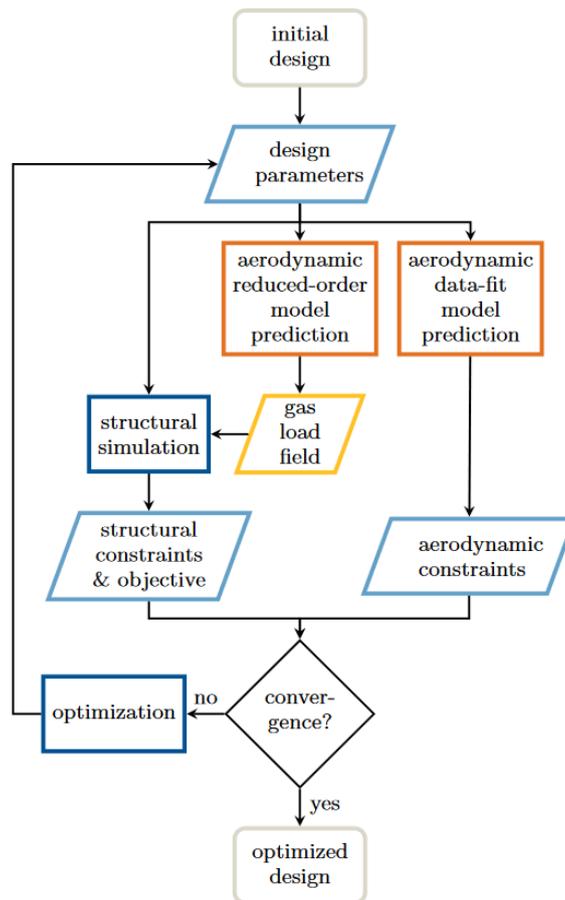


Figure 4.9: Flowchart of the IDO MF process [122]

In their study, [158] proposes a gradient-based sequential multi-fidelity approach based on successive single-fidelity MDF optimization derived from the multi-level optimization

acceleration technique [20, 92] in the mono-disciplinary case. The approach addresses the potential combinatorial explosion of the number of fidelity levels in MDO problems by first performing an error quantification for each model and selecting the fidelity levels that lie on the cost-error Pareto front. A criterion based on the first-order KKT conditions indicates when the current system fidelity level is no longer relevant. Error propagation is performed on the current solution to select the next system fidelity level with the highest normalized error reduction. It should be noted that the error propagation takes into account the correlation between the outputs of the same discipline but not the ones between different disciplines implied by the couplings.

Considerations on the number of low-fidelity models

It is notable that the optimal number of fidelity levels for a model to perform a multi-fidelity optimization is seldom addressed. The number of fidelity levels is typically fixed by the authors (typically two or three)[93, 96, 122] or left to the reader to consider more general multi-fidelity approaches with an arbitrary number of levels [108, 156, 158]. This complex issue may become essential in the future as the sources for generating fidelity levels become more numerous. This is particularly true when dealing with large-scale MDO problems. It is anticipated that a large number of fidelity levels could impede the entire process, either by selecting non-relevant fidelity levels or by complicating the architecture to an unprofitable level of time consumption. Furthermore, the selection criteria to be computed also typically scale poorly with the number of design and coupling variables. Conversely, an insufficient number of fidelity levels may result in the under-utilization of the full capacity of multi-fidelity approaches, with the consequence that high-fidelity models are over-exploited.

Related to the previous observation, provided fidelity models are typically considered to be all useful and already ranked from the lowest to the highest fidelity. This may not be true in practice, considering the numerous possible fidelity models and their interactions through couplings. This problem, already highlighted and partly addressed in [158], is still open and the following is dedicated to a first step toward an answer.

4.3 Down-selecting fidelity models for MDA

The composite nature of MDO presents a multitude of opportunities for the creation of novel fidelity models. If it is common in practice to simply replace certain components, such as disciplinary codes, with surrogate models in order to generate new fidelity models. In this context MDO offers a plethora of new implementation and engineering choices. Among these, hyperparameters are more numerous than in the mono-disciplinary case, allowing the generation of new fidelity models with a combination of changes to algorithm tolerances (e.g. optimizer, constraints, or MDA convergence) or more disciplinary-specific (such as grid precision) and others. Furthermore, more drastic changes can be considered, such as a complete modification of the architecture.

In order to narrow down the possibilities while still treating MDO-specific fidelity creation, several fidelity models for each discipline have been considered, as well as the effect of their combination on the multidisciplinary resolution fidelity.

4.3.1 Fidelity levels for MDA

As previously stated, MDAs are iterative algorithms that, at each iteration, typically execute all the disciplines' analysis until the coupling variables have converged, i.e. until the coupling variables verifies the consistency constraints (2.1.2). Consequently, MDAs are highly time-consuming and represent the main cost in the MDA-based architectures such as the MDF architecture and its derivatives. Hence, in a multi-fidelity framework for MDO where MDAs are mandatory the creation and the study of lower-fidelity models of MDAs is of paramount importance for performance. A straightforward method for generating an MDA fidelity level is to rely on the lower-fidelity levels of the disciplines and, thus on their respective disciplinary analysis approximations.

For the purposes of this study, p disciplines are defined whose respective coupling functions are computed for all possible designs as solution of:

$$y_i = \phi_i(x_0, x_i, y_{\neq i}) \quad \forall i \in \llbracket 1, p \rrbracket \quad (4.3.1)$$

Assume that each coupling function ϕ_i possesses $F_i - 1$ low-fidelity levels and denote the j^{th} low-fidelity level of discipline i as $\tilde{\phi}_{i,j}$. Assume that, outside of the high-fidelity model, the origins of the low-fidelity models are unknown *a priori*, hence there is no clear hierarchy between them. For the sake of consistency, the high-fidelity model for discipline i will be referenced either as ϕ or as $\tilde{\phi}_{i,F_i}$.

In this framework, it is possible to define a fidelity level for the MDA, Ψ , as a combination of p fidelity level choices, one for each discipline. Therefore, a fidelity level for the MDA is uniquely identified by a vector \mathcal{V} of p integers with the i^{th} component lying in $\llbracket 1, F_i \rrbracket$ ($\forall i \in \llbracket 1, p \rrbracket$).

Given a specific value of $\mathcal{V} \in \prod_{i \in \llbracket 1, p \rrbracket} \llbracket 1, F_i \rrbracket$, the low-fidelity MDA $\tilde{\Psi}_{\mathcal{V}}$ is defined implicitly as the continuously differentiable function that associates a design point (x_0, x) to the solution of the following system of equations in y :

$$\forall (x_0, x), \quad y = \tilde{\Psi}_{\mathcal{V}}(x_0, x) \iff \begin{bmatrix} y_1 - \tilde{\phi}_{1,\mathcal{V}_1}(x_0, x_1, y_{\neq 1}) \\ y_2 - \tilde{\phi}_{2,\mathcal{V}_2}(x_0, x_2, y_{\neq 2}) \\ \vdots \\ y_p - \tilde{\phi}_{p,\mathcal{V}_p}(x_0, x_p, y_{\neq p}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.3.2)$$

By convention, the highest fidelity MDA is defined as $\tilde{\Psi}_{\mathcal{V}^{HF}}$ (or alternatively by Ψ) with $\mathcal{V}^{HF} = [F_1, F_2, \dots, F_p]$.

Clearly, the number of potential MDA fidelity levels increases exponentially as the number of disciplines, p , and fidelity levels F_i ($i \in \llbracket 1, p \rrbracket$) increase, following the classical Cartesian product of p sets. Let $\#F_{MDA}$ the number of MDA fidelity levels that can be generated, i.e; the total number of possible coupling function combinations, it follows that:

$$\#F_{MDA} = \prod_{i=1}^p F_i \quad (4.3.3)$$

By further adding the possibility of "removing" a discipline from the MDA (i.e. the disciplinary analysis is not done during the process, hence outputs of the considered

discipline are held fixed), which can be seen as another fidelity level indexed as 0, and after removing the MDA with no disciplines at all (and therefore of poor interest), it follows that:

$$\#F_{MDA} = \left[\prod_{i=1}^p F_i + 1 \right] - 1 \quad (4.3.4)$$

The considerable number of potential fidelity levels indicates that the majority of them are contraindicated. This is particularly evident in the context of multi-fidelity frameworks, which typically assume that the different models involved are significantly disparate, both in terms of cost and/or precision. Moreover, all fidelity disciplinary models are coupled, which results in hard-to-predict coupling effects on the fidelity of the resulting MDA. Consequently, it is not evident whether a particular fidelity combination would be considered a higher or lower-fidelity level than another combination, with the exception of the high-fidelity MDA. This observation raises the question of how to rank and select the most relevant models for an effective multi-fidelity optimization.

To this aim, the proposed methodology approximate the two primary factors to be considered in evaluating the quality of a fidelity level : its cost (i.e. the time required to perform the multidisciplinary analysis) and its precision (i.e. the distance between the response and the output of the high-fidelity model). It is evident that a fidelity level of interest, denoted by $\tilde{\Psi}_{\mathcal{V}}$, where \mathcal{V} is the vector identifying which fidelity is used for each discipline, must be situated on the Pareto front $(cost(\tilde{\Psi}_{\mathcal{V}}), error(\tilde{\Psi}_{\mathcal{V}}))$ illustrated in Figure 4.10 and defined by the cost and error metrics, in order to be considered as an interesting model to be incorporated in a multi-fidelity strategy. This is not the first occurrence of a (cost, error) Pareto front to classify different fidelity models, see [116] for general multi-fidelity frameworks, or [28, 158] for more MDO-oriented examples.

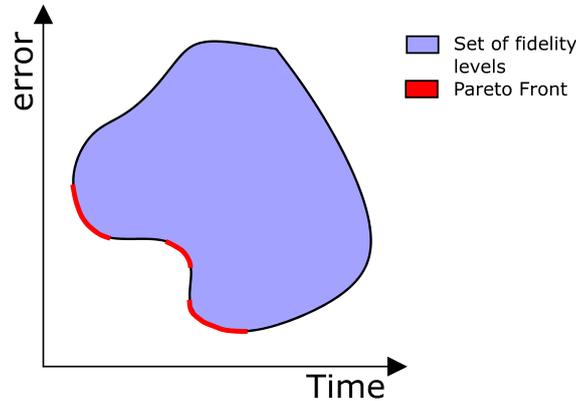


Figure 4.10: Modelization of a (cost, error) Pareto front

In order to construct the cost-error Pareto front and therefore select the optimal fidelity levels to retain, it is necessary to compute the time consumption and the error introduced by each fidelity level. This typically implies the use of costly statistics to estimate how well a low-fidelity model approximates the high-fidelity one. In [28] each fidelity model is evaluated on a set of points issued from a Design of Experiment. An error aggregating both objective and constraints inconsistencies is made. In [158] this Pareto front is utilized for eliminating suboptimal disciplines fidelity models, where the error is supposed to be independent of the design variables (x_0, x) . Nevertheless, it is important to acknowledge

that the selected models are not necessarily employed in the multi-fidelity optimization process. The optimization strategy, using a refinement approach, must determine which fidelity model must be optimized after each low-fidelity optimization. To this end, an error propagation is performed towards system objective and constraints and subsequently, a Monte Carlo approach is employed to estimate the next fidelity level that exhibits the best error reduction - cost ratio.

As previously highlighted, as the number of possible fidelity levels increases for each discipline, the number of possible fidelity levels for the coupling resolution model grows exponentially. Consequently, this combination forces a reconsideration of the use of costly statistics to rank each fidelity model. Instead, this study aims to explore and propose criteria based on local information, trading off the accuracy of the estimations with the capacity to treat a large number of models, sharing computational costs as much as possible.

Let $\tilde{\Psi}_{\mathcal{V}}$ be an approximation model of the high-fidelity MDA Ψ , where all high-fidelity disciplinary analysis ϕ_i have been replaced by their low-fidelity approximations $\tilde{\phi}_{i,\mathcal{V}_i}$. It is possible to obtain a fast approximation of the cost of $\tilde{\Psi}_{\mathcal{V}}$ by calculating the cost of an iteration of the fidelity level times the average number of iterations n_{HF}^* of the high-fidelity MDA:

$$\text{cost}(\tilde{\Psi}) = n_{HF}^* \times \text{cost}_{\text{iter}}(\tilde{\Psi}_{\mathcal{V}}) \quad (4.3.5)$$

The selection of the MDA algorithm influences both the cost of an iteration and the average number of iterations required to reach convergence. It is generally anticipated that n_{HF}^* will be lower when Gauss-Seidel MDAs are considered than when Jacobi MDAs are employed. This is due to the slower rate of convergence of the Jacobi algorithm in comparison to the Gauss-Seidel one [113]. With regard to the function $\text{cost}_{\text{iter}}$, which quantifies the cost of a single iteration of the specified MDA, the computational methodology varies slightly depending on the selected algorithm. This is illustrated in (4.3.6) and (4.3.7) which pertain to MDA Jacobi and MDA Gauss-Seidel, respectively. It should be noted that these costs are theoretical restitution time. In particular for the MDA Jacobi, this restitution time is typically restrained by the CPU cost and the number of available computational units.

$$\text{MDA Jacobi: } \text{cost}_{\text{iter}}(\tilde{\Psi}_{\mathcal{V}}) = \max_{i \in [1,p]} (\text{cost}(\tilde{\phi}_{i,\mathcal{V}_i})) \quad (4.3.6)$$

$$\text{MDA Gauss-Seidel: } \text{cost}_{\text{iter}}(\tilde{\Psi}_{\mathcal{V}}) = \sum_{i=1}^p \text{cost}(\tilde{\phi}_{i,\mathcal{V}_i}) \quad (4.3.7)$$

As the approximation of the error induced by each model is typically a more challenging task, necessitating numerous model evaluations, good approximations of all the terms involved in the previous equations are readily available at no additional cost after the aforementioned error approximation.

With regard to the discrepancy between models, as just hinted, the error introduced by a fidelity level is more challenging to quantify. For a specific design point (x_0, x) , the error associated with the coupling resolution can be determined by converging both the

low- and high-fidelity MDAs and comparing their outputs: $\tilde{\Psi}_{\mathcal{V}}(x_0, x) - \Psi(x_0, x)$. Without even considering statistics over the set of design variables (x_0, x) , given the multitude of fidelity levels and the necessity to perform a complete MDA for each one of them, i.e. compute $\tilde{\Psi}_{\mathcal{V}}(x_0, x)$ for every possible combination \mathcal{V} , it is not conceivable to hope for an estimation of the errors in a reasonable amount of time.

The subsequent section is devoted to numerical tools or criteria that have been devised with the objective of providing an estimate of the error introduced by each fidelity level in a reasonable amount of time.

4.3.2 Two criteria for error estimation

Two criteria for a fast approximation of the error introduced by low-fidelity MDAs are presented hereafter. Considering the large number of fidelity models, these criteria are made to give a rough approximation of the true hierarchy between the different models. This comes from the assumption that even a rough approximation of the ranking, even if it leads to the consideration of sub-optimal models, is expected to yield some gain when using multi-fidelity approaches. Therefore, both criteria are based on local information at a specific design point (x_0, x) and they do not necessitate costly statistics to be computed similarly as in [158]. They estimate the magnitude of the error induced by each model, which is assumed to remain globally the same on the entirety of the research domain. The overall ranking of all models should not be significantly affected, thus maintaining the relative positioning of each model in terms of error.

A coupled adjoint based criterion

The first proposed criterion is based on the coupled adjoint (cf. Section 2.2.2). This criterion assumes that a change in fidelity for one or more disciplines only disturbs the high-fidelity optimization problem with a constant error on the multidisciplinary feasibility constraints. Therefore, by only considering an initial design (x_0, x) , the said criterion approximates the error on the functions of interest at the optimal design.

Settings Let us consider the following high-fidelity optimization problem:

$$\begin{aligned} & \min_{x_0 \in \mathcal{X}_0, x \in \mathcal{X}, y \in \mathbb{R}^m} f(x_0, x, y) \\ \text{s.t.} \quad & g(x_0, x, y) = 0 \\ & y_i - \phi_i(x_0, x_i, y_{\neq i}) = 0 \quad i \in \llbracket 1, p \rrbracket, \end{aligned} \tag{PHF}$$

where $\mathcal{X}_0 \subset \mathbb{R}^{n_0}$ and $\mathcal{X} \subset \mathbb{R}^n$ denote the design spaces of the shared and local variables respectively, $f : \mathcal{X}_0 \times \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}$ the objective, and $g : \mathcal{X}_0 \times \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_c}$ the constraints. The functions $\phi_i : \mathbb{R}^{n_0+n_i} \rightarrow \mathbb{R}^{m_i}$ denotes the coupling function associated with discipline i , $i = 1, \dots, p$. All these functions are considered to be at least twice continuously differentiable.

Making a change on the fidelity level of one or more disciplines, i.e. merely replacing those computational components by low-fidelity ones, has the effect of perturbing their respective outputs for the same input (x_0, x) . This impact is reflected in the coupling resolution, with the disturbed disciplinary analysis of discipline i , with fidelity j being denoted as $\tilde{\phi}_{i,j}$. Let \mathcal{V} be the p -dimensional vector such that for all $i \in \llbracket 1, p \rrbracket$, the chosen fidelity level for

discipline i is \mathcal{V}_i . Hence, the choice of \mathcal{V} defines the fidelity level for the MDA as a direct consequence of the replacement of the functions that compute the coupling variables in the previous optimization problem (P^{HF}). These new coupling functions $\tilde{\phi}_{i,\mathcal{V}_i}$ are replacing the high-fidelity ones, ϕ_i , leading to a newly defined low-fidelity problem:

$$\begin{aligned} & \min_{x_0 \in \mathcal{X}_0, x \in \mathcal{X}, y \in \mathbb{R}^m} f(x_0, x, y) \\ \text{s.t.} \quad & g(x_0, x, y) = 0 \\ & y_i - \tilde{\phi}_{i,\mathcal{V}_i}(x_0, x_i, y_{\neq i}) = 0 \quad i \in \llbracket 1, p \rrbracket \end{aligned} \quad (P_{\mathcal{V}}^{LF})$$

Similarly as in the MDF architecture, one can define a multidisciplinary analysis algorithm as an implicit continuously differentiable function from a set of coupling functions by using the Implicit function theorem (cf. Theorem 2.2.5).

Let $\Psi : \mathcal{X}_0 \times \mathcal{X} \rightarrow \mathbb{R}^m$ be the high-fidelity MDA function:

$$\forall (x_0, x) \in \mathcal{X}_0 \times \mathcal{X} \quad y = \Psi(x_0, x) \iff \forall i \in \llbracket 1, p \rrbracket \quad y_i - \phi_i(x_0, x_i, y_{\neq i}) = 0 \quad (4.3.8)$$

and let $\tilde{\Psi}_{\mathcal{V}} : \mathcal{X}_0 \times \mathcal{X} \rightarrow \mathbb{R}^m$ be the low-fidelity MDA function described by \mathcal{V} :

$$\forall (x_0, x) \in \mathcal{X}_0 \times \mathcal{X} \quad y = \tilde{\Psi}_{\mathcal{V}}(x_0, x) \iff \forall i \in \llbracket 1, p \rrbracket \quad y_i - \tilde{\phi}_{i,\mathcal{V}_i}(x_0, x_i, y_{\neq i}) = 0 \quad (4.3.9)$$

In other words, both Ψ and $\tilde{\Psi}_{\mathcal{V}}$ are twice continuously differentiable functions which give the coupling solution with respect to (x_0, x) for the disciplinary analysis system created from the ϕ_i 's (or $\tilde{\phi}_{i,\mathcal{V}_i^{HF}}$ or $\tilde{\phi}_{i,F_i}$ equivalently) and $\tilde{\phi}_{i,\mathcal{V}_i}$'s respectively.

By definition of the consistency constraints, they can be rewritten as:

$$\begin{aligned} & y_i - \tilde{\phi}_{i,\mathcal{V}_i}(x_0, x_i, y_{\neq i}) = 0 \quad i \in \llbracket 1, p \rrbracket \\ \iff & y - \tilde{\Psi}_{\mathcal{V}}(x_0, x) = 0 \\ \iff & y - \Psi(x_0, x) = \tilde{\Psi}_{\mathcal{V}}(x_0, x) - \Psi(x_0, x) \end{aligned} \quad (4.3.10)$$

Where $c_{\mathcal{V}} = \tilde{\Psi}_{\mathcal{V}}(x_0, x) - \Psi(x_0, x) \in \mathbb{R}^{n_y}$ denotes the error on the coupling resolution induced by the fidelity level change, defined by \mathcal{V} , in the disciplines and is assumed to be independent of (x_0, x) . Therefore the newly obtained optimization problem ($P_{\mathcal{V}}^{LF}$) can be rewritten as a parameterized version of (P^{HF}):

$$\begin{aligned} & \min_{x_0 \in \mathcal{X}_0, x \in \mathcal{X}, y \in \mathbb{R}^m} f(x_0, x, y) \\ \text{s.t.} \quad & g(x_0, x, y) = 0 \\ & y - \Psi(x_0, x) = c_{\mathcal{V}} \end{aligned} \quad (P_{c_{\mathcal{V}}}^{LF})$$

The parameter $c_{\mathcal{V}} = \tilde{\Psi}_{\mathcal{V}}(x_0, x) - \Psi(x_0, x)$ being the error on the consistency constraints introduced by the change of fidelity in the disciplines defined by \mathcal{V} . In particular, it should be noted that in the event where $c_{\mathcal{V}} = 0$ (which should happen only if $\mathcal{V} = \mathcal{V}^{HF}$), ($P_{c_{\mathcal{V}}}^{LF}$) reduces to (P^{HF}). The proposed criterion is designed to determine the impact of a potential change in discipline on the objective's precision at the solution (x_0^*, x^*) with minimal computational cost. In other words, given a combination of fidelity levels \mathcal{V}

and an estimation of $c_{\mathcal{V}}$, the criterion should provide an estimate of the impact on the optimum of $(P_{c_{\mathcal{V}}}^{LF})$.

As the consistency constraints, $y - \Psi(x_0, x) = c_{\mathcal{V}}$, are also continuously differentiable with respect to $c_{\mathcal{V}}$ by assumption, the following result is a direct application of the envelope theorem (cf. Theorem 2.2.7) to deduce the impact of a fidelity change on the optimal objective function value.

Impact on the objective value at the optimal design Let the following optimization problem

$$\begin{aligned} \min_{x_0 \in \mathcal{X}_0, x \in \mathcal{X}, y \in \mathbb{R}^m} \quad & f(x_0, x, y) \\ \text{s.t.} \quad & g(x_0, x, y) = 0 \\ & R(x_0, x, y; c) = 0 \end{aligned} \tag{P_{res}^{LF}}$$

with $R : \mathcal{X}_0 \times \mathcal{X} \times (\mathbb{R}^m)^2 \rightarrow \mathbb{R}^m$ being the consistency constraints on y , solved with an error c . This is equivalent to $(P_{c_{\mathcal{V}}}^{LF})$ (with $R(x_0, x, y; c) = y - \Psi(x_0, x) - c$) but consistency constraints are given in a residual form R which is clearly twice continuously differentiable with respect to all of its variables.

Let \mathcal{L} be the Lagrangian function of (P_{res}^{LF}) :

$$\mathcal{L}(x_0, x, y, \mu, \lambda; c) = f(x_0, x, y) + \mu^T g(x_0, x, y) + \lambda^T R(x_0, x, y; c) \tag{4.3.11}$$

where $\mu \in \mathbb{R}^{n_c}$ and $\lambda \in \mathbb{R}^m$ being the so-called Lagrange multipliers associated with g and R respectively.

For all value c in a small neighborhood of 0, assume that [Fiacco's theorem](#) holds, i.e. there exist 5 continuously differentiable functions x_0^*, x^*, y^*, μ^* and λ^* such that $(x_0^*(c), x^*(c), y^*(c), \mu^*(c), \lambda^*(c))$ is the optimal primal-dual pair, such as defined in [1st order KKT condition](#), for (P_{res}^{LF}) and define

$$\mathcal{L}^*(c) \equiv \mathcal{L}(x_0^*(c), x^*(c), y^*(c), \mu^*(c), \lambda^*(c); c) \tag{4.3.12}$$

the Lagrangian value at this solution. By definition, $\mathcal{L}^*(c)$ is given as follows:

$$\begin{aligned} \mathcal{L}^*(c) = & f(x_0^*(c), x^*(c), y^*(c)) \\ & + \mu^*(c)^T g(x_0^*(c), x^*(c), y^*(c)) \\ & + \lambda^*(c)^T R(x_0^*(c), x^*(c), y^*(c); c) \end{aligned} \tag{4.3.13}$$

\mathcal{L}^* is clearly continuously differentiable with respect to c and a direct derivation of [\(4.3.13\)](#) gives:

$$\begin{aligned} \frac{d\mathcal{L}^*}{dc}(c) = & \frac{\partial f}{\partial x_0} \frac{dx_0^*}{dc} + \frac{\partial f}{\partial x} \frac{dx^*}{dc} + \frac{\partial f}{\partial y} \frac{dy^*}{dc} \\ & + \mu^*(c)^T \left[\frac{\partial g}{\partial x_0} \frac{dx_0^*}{dc} + \frac{\partial g}{\partial x} \frac{dx^*}{dc} + \frac{\partial g}{\partial y} \frac{dy^*}{dc} \right] \\ & + \lambda^*(c)^T \left[\frac{\partial R}{\partial x_0} \frac{dx_0^*}{dc} + \frac{\partial R}{\partial x} \frac{dx^*}{dc} + \frac{\partial R}{\partial y} \frac{dy^*}{dc} + \frac{\partial R}{\partial c} \right] \end{aligned} \tag{4.3.14}$$

By reordering terms, (4.3.14) can be rewritten as follows:

$$\begin{aligned}
\frac{d\mathcal{L}^*}{dc}(c) &= \left[\frac{\partial f}{\partial x_0} + \mu^*(c)^T \frac{\partial g}{\partial x_0} + \lambda^*(c)^T \frac{\partial R}{\partial x_0} \right] \frac{dx_0^*}{dc} \\
&+ \left[\frac{\partial f}{\partial x} + \mu^*(c)^T \frac{\partial g}{\partial x} + \lambda^*(c)^T \frac{\partial R}{\partial x} \right] \frac{dx^*}{dc} \\
&+ \left[\frac{\partial f}{\partial y} + \mu^*(c)^T \frac{\partial g}{\partial y} + \lambda^*(c)^T \frac{\partial R}{\partial y} \right] \frac{dy^*}{dc} \\
&+ \lambda^*(c)^T \frac{\partial R}{\partial c}(c)
\end{aligned} \tag{4.3.15}$$

The first three terms between brackets in (4.3.15) correspond to the first order **1st order KKT condition** respectively, all of them being evaluated in $(x_0^*(c), x^*(c), y^*(c), \mu^*(c), \lambda^*(c); c)$, the optimal primal-dual pair. Therefore:

$$\frac{\partial \mathcal{L}}{\partial x_0}(x_0^*(c), x^*(c), y^*(c), \mu^*(c), \lambda^*(c); c) = \frac{\partial f}{\partial x_0} + \mu^*(c)^T \frac{\partial g}{\partial x_0} + \lambda^*(c)^T \frac{\partial R}{\partial x_0} = 0 \tag{4.3.16}$$

$$\frac{\partial \mathcal{L}}{\partial x}(x_0^*(c), x^*(c), y^*(c), \mu^*(c), \lambda^*(c); c) = \frac{\partial f}{\partial x} + \mu^*(c)^T \frac{\partial g}{\partial x} + \lambda^*(c)^T \frac{\partial R}{\partial x} = 0 \tag{4.3.17}$$

$$\frac{\partial \mathcal{L}}{\partial y}(x_0^*(c), x^*(c), y^*(c), \mu^*(c), \lambda^*(c); c) = \frac{\partial f}{\partial y} + \mu^*(c)^T \frac{\partial g}{\partial y} + \lambda^*(c)^T \frac{\partial R}{\partial y} = 0 \tag{4.3.18}$$

Consequently, (4.3.15) can be simplified as:

$$\frac{d\mathcal{L}^*}{dc}(c) = \lambda^*(c)^T \frac{\partial R}{\partial c}(c) = -\lambda^*(c), \tag{4.3.19}$$

with $\frac{\partial R}{\partial c}(c) = -I_{n_y}$ recalling that $R(x_0, x, y; c) = -c$ for all design variables (x_0, x, y) and parameters c .

Let $f^* : c \mapsto f(x_0^*(c), x^*(c), y^*(c))$, be the value function which associates for any c the unique, optimal value of $(P_{c,y}^{LF})$. Assume that f^* is continuously differentiable with respect to c , then, according to the envelope theorem (Theorem 2.2.7), for all possible value of c , it follows that:

$$\frac{\partial f^*}{\partial c}(c) = \frac{\partial \mathcal{L}^*}{\partial c}(c) = -\lambda^*(c). \tag{4.3.20}$$

Therefore the first order Taylor expansion can be applied to the value function with respect to c :

$$f^*(c) = f^*(0) + \frac{\partial f^*}{\partial c}(0)c + \epsilon(c) \quad \text{where } \lim_{c \rightarrow 0} \epsilon(c) = 0. \tag{4.3.21}$$

Denoting $\lambda^{HF} = \lambda^*(0)$, the high-fidelity optimal Lagrange multipliers at $(x_0^*(0), x^*(0), y^*(0))$, it follows that

$$f^*(c) - f^*(0) \approx -\lambda^{HF} c. \tag{4.3.22}$$

In other words, the dot product between the high-fidelity optimal Lagrange multipliers associated with the coupling constraints residuals, and the coupling error, gives a first

order approximation of the error on the objective function. However, as it stands, the said criterion necessitates to perform the high-fidelity optimization, to compute the optimal Lagrange multipliers, which depends, among other things, on the set of active constraints. As this criterion do not intend to complete any optimization to be computed, there is a need to approximate both λ^{HF} and c without being at the optimum. Instead, these terms should be computed at any design point.

Recall that at any design point the multidisciplinary constraint R admits a unique solution:

$$R(x_0, x, y; c) = 0 \iff y = \Psi(x_0, x) + c. \quad (4.3.23)$$

As this residual is enforced to be equal to zero at each design choice by the computational process, the adjoint equations state that there exists a unique vector λ^f (see Section 2.2.2), the so-called adjoint vector associated with objective function, solution of the linear system

$$\frac{\partial R^T}{\partial y} \lambda^f = \frac{\partial f^T}{\partial y} \quad (4.3.24)$$

which can be seen as the computation of the total derivative of the objective with respect to a perturbation of the residual, i.e. $\lambda^f = \frac{df}{dc}$ with $f : (x_0, x) \mapsto f(x_0, x, \Psi(x_0, x) + c)$.

Recall that c is assumed to preserve its magnitude on the whole space of research, or in other words that the fidelity error remains essentially the same and does not depend on the design variables. Therefore, the proposed methodology approximates the sensibility of the value function f^* by the total derivative of the objective at any design point:

$$\frac{\partial f^*}{\partial c}(c) \approx \frac{df}{dc}(x_0, x; 0) = \frac{\partial f}{\partial c} - (\lambda^f)^T \frac{\partial R}{\partial c} = -(\lambda^f)^T. \quad (4.3.25)$$

In practice, this is equivalent to make the approximation $(\lambda^f)^T \approx \lambda^{HF}$, where $(\lambda^f)^T$ is computed on any design point.

Indeed, this is a rough approximation, as the constraints g and their related Lagrange multipliers at the optimum $\mu^*(c)$ are completely neglected in the sense that this result implicitly implies that the set of active constraints remains the same at (x, x_0) and at $(x_0^*(c), x^*(c))$. This is obviously false for most optimization problems whenever inequality constraints are presents. This is, however a price we are willing to pay considering the high amount of models to rank, the hypothesis being that the resulting classification stays globally unchanged compared to the one we should have obtained with more computationally intensive approximations.

In some cases, this approximation is more reliable. For the following unconstrained case, for instance:

$$\begin{aligned} & \min_{x_0 \in \mathcal{X}_0, x \in \mathcal{X}, y \in \mathbb{R}^m} && f(x_0, x, y) \\ & \text{s.t.} && R(x_0, x, y; c) = 0 \end{aligned} \quad (4.3.26)$$

this is a known result [52] that, at the optimum $(x_0^*(0), x^*(0), y^*(0))$ of (4.3.26) for $c = 0$, both λ^{HF} , the optimal Lagrange multipliers vector, and $(\lambda^f)^T$, the adjoint vector associated with the objective function, are equal to $\frac{\partial f^*}{\partial c}$ where f^* is the value function of (4.3.26) that associates c to $f(x_0^*(c), x^*(c), y^*(c))$. In other words, they are equal to the first order approximations of the impact on the optimum for a perturbation of the residual:

$$f^*(c) - f^*(0) \approx \frac{\partial f^{*T}}{\partial c} (0)c = (\lambda^{HF})^T c = (\lambda^f)^T c. \quad (4.3.27)$$

Consequently, this approximation is expected to exhibit superior performance in the unconstrained case, even if a non-optimal design point is considered, and especially when considering that the error on the model stays globally of the same magnitude in the whole space of research.

Definition of the criterion Given previous results, a criterion to measure the accuracy of a given model defined by a fidelity combination \mathcal{V} is given by:

$$\epsilon^{\text{adj}} = - \sum_{i=1}^p \lambda_i^f [\tilde{\phi}_{i,\nu_i}(x_0, x_i, y_{\neq i}^{HF}) - y_i^{HF}] \quad (4.3.28)$$

where y^{HF} is the high-fidelity coupling solution at (x_0, x) , i.e. $y^{HF} = \Psi(x_0, x)$.

This criterion is inspired by previous adjoint correction based methods for grid fidelity estimation [12, 11, 152]. It assumes that most of the error on models comes from a deterioration of the coupling resolution and gives a first-order approximation of the error on the considered functions. Both λ^f and y^{HF} necessitate the execution of the high-fidelity model for a specific design value (x_0, x) , this computation is done a unique time for all fidelity models. To avoid the computation of the residual at convergence for each model, i.e. $\tilde{\Psi}_{\mathcal{V}}(x_0, x) - \Psi(x_0, x)$, it is approximated using the discrepancy between the high-fidelity solution and a unique disciplinary analysis computed on the high-fidelity solution y^{HF} .

Let us consider a design point (x_0, x) , a high-fidelity MDA Ψ , and list a low-fidelity disciplinary analysis $L_{\tilde{\phi}}$. Further assume that the adjoint equation (4.3.24) admits a unique solution, i.e. $\frac{\partial R^T}{\partial y}(x_0, x, \Psi(x_0, x))$ is non-singular. Algorithm 4 describes the overall process to compute the adjoint based criterion given in Equation (4.3.28) for each combination $\mathcal{V} \in \times_{i \in [1,p]} [1, F_i]$ of disciplinary fidelity:

Algorithm 4: Adjoint correction based criterion

Data: (x_0, x) , Ψ , $L_{\tilde{\phi}}$

Result: $[\epsilon_{\mathcal{V}}^{\text{adj}}]$

Compute HF solution : $y^{HF} = \Psi(x_0, x)$;

Compute λ^f as the solution of $\frac{\partial R^T}{\partial y}(x_0, x, y^{HF})\lambda^f = \frac{\partial f^T}{\partial y}(x_0, x, y^{HF})$;

do in parallel

foreach $\tilde{\phi}_{i,j} \in L_{\tilde{\phi}}$ **do**

 | Compute the couplings on HF solution: $\tilde{y}_{i,j} = \tilde{\phi}_{i,j}(x_0, x_i, y^{HF})$;

end

end

do in parallel

foreach $\mathcal{V} \in \times_{i \in [1,p]} [1, F_i]$ **do**

 | Compute adjoint correction based criterion : $\epsilon_{\mathcal{V}}^{\text{adj}} = - \sum_{i=1}^p \lambda_i^f [\tilde{y}_{i,\nu_i} - y_i^{HF}]$

end

end

This algorithm has two major costs, the first one is the necessity to converge the high-fidelity MDA Ψ to obtain the reference coupling values y^{HF} , the second is the computation of the coupled adjoint λ^f , that necessitates the solution of the related linear system. The second step of the algorithm is highly parallelized, each disciplinary fidelity model can be treated independently and only necessitate a unique execution to be evaluated at (x_0, x, y^{HF}) . Considering that the high-fidelity disciplines have already been executed to obtain both y^{HF} and λ^f , and recall that discipline i has F_i fidelity models, the total number of model execution for step 2 is given by:

$$n_{\text{calls}} = \left[\sum_{i=1}^p F_i \right] - p \quad (4.3.29)$$

A gradient alignment criterion

As the fully coupled adjoint is not guaranteed to be available, a second criterion, based on the gradient information is explored. A way to quickly estimate the error of a fidelity level is to compute the angle between the high-fidelity model's gradient, denoted as ∇f^{HF} , and the low-fidelity model's gradient ∇f^{LF} on a specific design point (x_0, x) and the high-fidelity's couplings $y^{HF} = \Psi(x_0, x)$. This estimator, which is referred to as the gradient alignment criterion is defined by:

$$\epsilon_f^{\text{align}} \approx 1 - \text{align}(\nabla f^{HF}, \nabla f^{LF}, x_0, x) \quad (4.3.30)$$

where

$$\text{align}(\nabla f^{HF}, \nabla f^{LF}, x_0, x) = \begin{cases} -1, & \text{if } \|\nabla f^{HF}(x_0, x, \Psi(x_0, x))\| = 0 \\ -1, & \text{if } \|\nabla f^{LF}(x_0, x, \Psi(x_0, x))\| = 0 \\ \frac{\nabla f^{HF}(x_0, x, \Psi(x_0, x)) \cdot \nabla f^{LF}(x_0, x, \Psi(x_0, x))}{\|\nabla f^{HF}(x_0, x, \Psi(x_0, x))\| \|\nabla f^{LF}(x_0, x, \Psi(x_0, x))\|}, & \text{Else} \end{cases} \quad (4.3.31)$$

The alignment operator, which was introduced in [108] as a validation criterion, assesses whether both fidelity levels suggest the same descent direction at (x_0, x) . The error, which lies by definition within the interval $[0, 2]$, is simply obtained by penalizing the discrepancy between the two descent directions. Purely aligned descent directions yields a zero error, while orthogonal ones result in an error of 1. In the event that both gradients point in opposite directions, or that the low-fidelity gradient vanishes, the upper bound, 2, is reached. It should be noted that, by definition of the alignment, the chosen design point (x_0, x) should not be on where the high-fidelity gradient vanishes.

The associated algorithm, which compute the gradient based criterion for each low-fidelity models, is given in Algorithm 5.

Algorithm 5: Gradient alignment criterion

Data: (x_0, x) , L_Ψ
Result: $[\epsilon_f^{align}]$
Compute HF solution : $y^{HF} = \Psi(x_0, x)$;
Compute HF gradients : $\nabla f^{HF} = \nabla f^{HF}(x_0, x, y^{HF})$;
do in parallel
 foreach $\tilde{\Psi} \in L_\Psi$ **do**
 Compute LF gradients : $\nabla f^{LF} = \nabla f^{LF}(x, y^{HF})$;
 Compute gradient alignment criterion : $\epsilon_f^{align} = 1 - \text{align}(\nabla f^{HF}, \nabla f^{LF})$
 end
end

Considering the cost of computing the alignment criterion, there is a unique costly operation : converging the high-fidelity MDA to obtain the high-fidelity coupling values, i.e. computing $y^{HF} = \Psi(x_0, x)$. The gradients for each models are computed separately (preferably in parallel) and the criterion computation follows directly at an affordable cost.

The alignment criterion (4.3.30) is defined for the objective function but constraints can also be considered, contrary to the adjoint-based criterion presented earlier, which is based on the envelope theorem (cf. Theorem 2.2.7). If multiple functions are accounted for, a natural way to describe the resulting Pareto front is to extend (4.3.30) to a multi-dimensional error, where each error is considered as a minimization criterion. In this case, each Pareto-optimal model is guaranteed to be Pareto-optimal for every considered functions at once. However, this is rarely done in practice as, if too many objective are taken into account, the obtained Pareto front is likely to be degenerated [72]. Instead, the errors on a set of functions $\mathcal{S} \subseteq \{f, g_i \ i = 1, \dots, p\}$ can be aggregated using a weighted sum:

$$\epsilon^{align} = \sum_{h \in \mathcal{S}} w_h \times \epsilon_h^{align} \quad \text{with} \quad w_h > 0 \quad \forall h \in \mathcal{S} \quad \text{and} \quad \sum_{h \in \mathcal{S}} w_h = 1 \quad (4.3.32)$$

The error, in this case, represents the weighted average discrepancy between gradient alignments for all considered functions.

This method is typically preferred to extending the Pareto front for its simplicity of implementation and visualization. However, the choice of the weight is problem specific and must be given *a priori*. Moreover, in the event where the Pareto front is not convex, certain Pareto optimal points may not be seen, depending on the choice for the weights.

In [158], a similar strategy considering all functions of interest, i.e. the objective and all the constraints, is used to find the next fidelity model. As the error propagation is done after the previous fidelity optimization, Lagrange multipliers can be accessed directly from the optimizer and used as weight, as they act as shadow prices, for scaling the constraints.

4.3.3 Pareto front and post treatment

Once the costs and errors associated with each model have been successfully approximated, a $(cost, error)$ Pareto front can be constructed. However, when only considering models on the aforementioned Pareto front, it is possible that not all models may be relevant for the optimization process. Therefore, a procedure is required to select the optimal fidelity levels on the Pareto front.

Once a desired number of fidelity levels is provided, a greedy algorithm is executed for the purpose of down-selecting relevant fidelity levels. Given a number of fidelity levels to be down-selected, the greedy algorithm comprises two steps :

- **Step 1:** The initial selection of models is conducted to ensure an even distribution along one of the two axis. Given the anticipated variability in cost estimates, the cost selection is typically based on the logarithmic transformation of the costs. Regardless of the requested number of fidelity levels, the high-fidelity model is always included in the selection. For instance, for a two fidelity models selection, the chosen models are the two extremes of the Pareto front. A three fidelity selection is illustrated by Figure 4.11 for both axis. Figure 4.11a illustrate a three models evenly spread selection along the cost axis, while Figure 4.11b shows the same selection but considering the error axis instead. It should be noted that the selection of the in-between low-fidelity models, i.e. not the lowest fidelity model nor the highest, can be greatly affected by the choice of the axis.

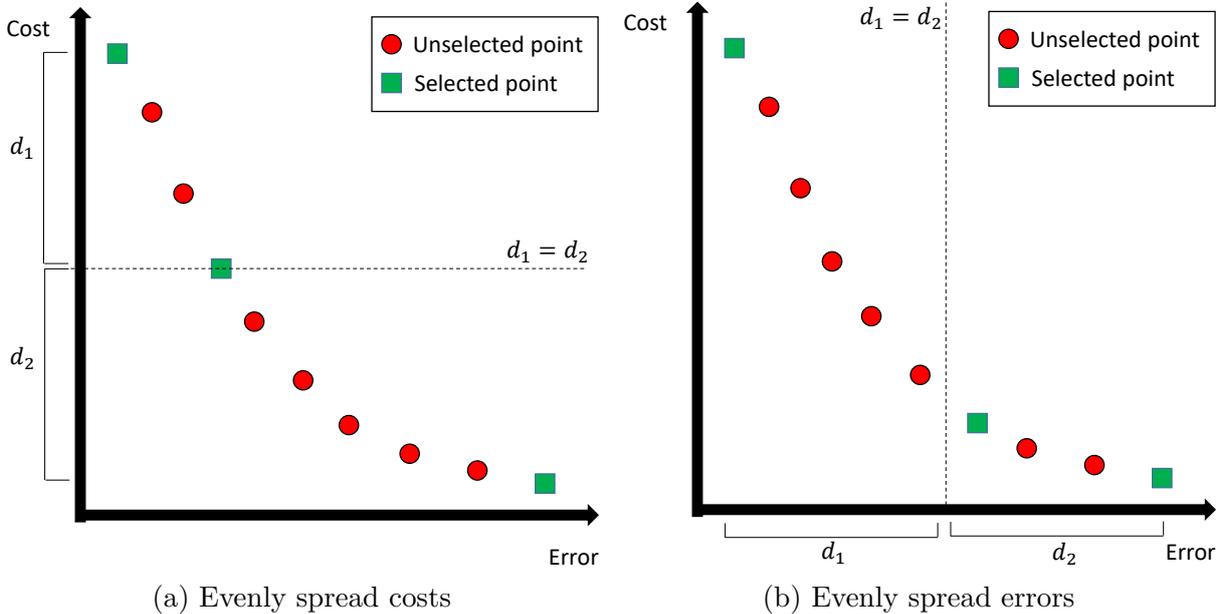


Figure 4.11: Evenly spread fidelity selection

- **Step 2:** The preceding step proposes models that may not be relevant for a multi-fidelity optimization process as it is often assumed that all selected fidelity models should be different enough in terms of cost and error. Therefore, if two selected models have a cost and/or an error that are considered to be too similar, one should be removed. A post-processing algorithm is dedicated to the elimination of redundant models. This is done with a minimal user-defined minimal slope, or a

threshold $\mathcal{T}_r > 0$, for one of the two criterion between two selected Pareto points. In its default configuration, the post-selection algorithm selects the Pareto point that minimizes the error over the cost. In other words, if two selected Pareto point P and P' are such that $cost(P) > cost(P')$ and $error(P) < error(P')$, then P should be preferred over P' if:

$$-\frac{error(P) - error(P')}{cost(P) - cost(P')} < \mathcal{T}_r \quad (4.3.33)$$

An example of the removal of an already selected Pareto point is illustrated in Figure 4.12. In this case, the closest neighbor under the threshold is preferred instead, as it minimizes the error.

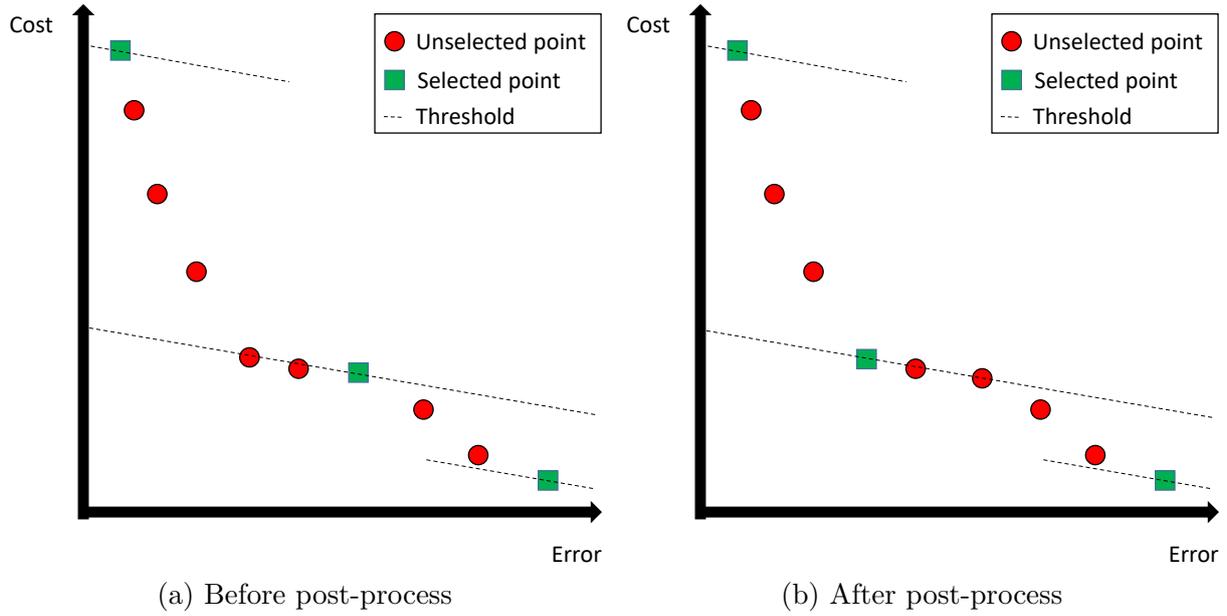


Figure 4.12: Example of the application of the post-treatment: the second selected point is removed from the selection and replaced with a similar point that minimizes the error.

4.4 Multi-fidelity methodologies validation

This section collects experiments and applications to test the proposed multi-fidelity methods.

First, the multi-fidelity framework, based on a refinement approach and in continuity with *Romain Olivanti's* work [109, 108], is introduced and implemented in a MDO context. This multi-fidelity approach is considered for all the following applications.

Then, several applications are performed to test the two sensitivity criteria presented in Section 4.3 with simple models where fidelity levels are easily generated to illustrate the proposed methodology in a multi-fidelity context, from the definition and ranking of fidelity models to complete multi-fidelity optimizations. The adjoint-based criterion is tested on a perturbed Sellar test case where the fidelity models are generated by varying a couple of parameters. The 441 fidelity models are ranked with the said criterion and the resulting choice of fidelity models is compared in terms of discipline calls reduction with other selections on the refinement multi-fidelity approach. The gradient alignment

criterion is tested on the SSBJ test case (see Appendix A) where low-fidelity levels are generated by reordering disciplines and neglecting backward effects.

Finally, to illustrate more complex fidelity construction and to make the connection with the previous chapter on bi-level architectures, a multi-fidelity application is performed on the SSBJ test case by considering bi-level MDO architectures as fidelity models. The goal of this relatively small application is to show the capabilities of the different implementations and how they can be intertwined to create more complex optimization processes.

4.4.1 A multi-fidelity refinement framework

Given an arbitrary number of already ranked fidelity levels, a generic multi-fidelity (or multilevel) refinement approach is considered. This framework optimizes each fidelity model sequentially, from the lowest to the highest fidelity level, each optimization being warm-started with the best point of the previous suboptimization. In consideration of the pre-established hierarchy, it is hoped that each suboptimization will approach the high-fidelity solution as the fidelity increases, thereby reducing the necessity for costly calls from subsequent levels. This model management strategy may be classified as a filtering approach, as suboptimizations are terminated according to criteria contingent upon the current low-fidelity model [116].

The choice of using the multi-fidelity refinement approach has been motivated by several factors. Firstly, the method is shown to be locally convergent as long as it is ensured that the process switches to the high-fidelity model, and that the algorithm that optimizes the high-fidelity model is itself locally convergent for this model. Moreover, the multi-fidelity refinement approach is a generic and non-intrusive method in that it makes no assumptions on the problem's structures, the nature of the low-fidelity models nor their number. Last but not least, the multi-fidelity refinement method is well-adapted to deal with high-dimensional and high-fidelity applications, it has been shown to be promising in multiple occasions, for instance for adjoint-based aerodynamic shape optimization [92, 109, 108] or for an aerostructural problem within [158].

The multi-fidelity framework based on the refinement approach has been implemented within a GEMSEO [53] multi-fidelity package. This implementation is based on an earlier multi-fidelity package originally developed by *Romain Olivanti's* [108] in Python2, compatible with GEMS V3, and designed to tackle multi-fidelity MDO problems. The original framework already allows for variable parameterization between levels through full rank linear prolongation and restriction. As gradients are considered to be available for all the fidelity models, a first-order criterion based on gradient alignment is employed to decide whether or not it is time to switch to the next fidelity level. The asynchronous validation mechanism permits the current optimization to continue while verifying the validity of a given design point.

The extension of this package to include sensitivity criteria and Pareto selection, and the porting of this code is now fully compatible with python3 and the latest version of GEMSEO, and benefits from all the latest features of GEMSEO in terms of workflow generation and data exchange.

4.4.2 Coupled adjoint criterion

A toy test case: a disturbed Sellar case

To test the coupled adjoint criterion, the Sellar [135] test case have been considered for its simplicity. This optimization problem and its guiding equations are given in more details within Appendix B.1. To quickly generate low-fidelity models of the two disciplines involved, we propose a multi-fidelity variant of the Sellar problem where each disciplinary analysis has been perturbed by scaling the coupling input by a user-defined factor. These factors, K_1 and K_2 , are user-defined hyperparameters that intervene in the coupling functions of the Sellar1 and Sellar2 disciplines, respectively. These scaling factors affect the design in the following ways:

$$\text{Sellar 1: } \phi_1(x_0, x_1, y_2) = \sqrt{x_{0,1}^2 + x_1 + x_{0,2} - 0.2K_1y_2} \quad (4.4.1)$$

$$\text{Sellar 2: } \phi_2(x_0, y_1) = K_2|y_1| + x_{0,1} + x_{0,2} \quad (4.4.2)$$

By varying the value of the couple (K_1, K_2) , the fidelity of the resulting MDA changes. The couple $(K_1, K_2) = (1, 1)$ is considered as the high-fidelity MDA.

The cost is simulated by delaying the discipline computation depending on how far the pair (K_1, K_2) is from the high-fidelity pair $(1, 1)$. Typically, in the following, a quadratic model is built for each value K_i $i \in \{1, 2\}$ within the interval $[0, 2]$ for both disciplines. This quadratic model, given by Equation (4.4.3), is constructed such that the maximum cost is reached for $K_i = 1$ $i \in \{1, 2\}$, and such that a discipline with a value of K_i that reaches its extreme values, i.e. $K_i \in \{0, 2\}$, has a cost of zero.

$$\text{cost}(K_i) = -\nu K_i^2 + 2\nu K_i \quad i \in \{1, 2\}. \quad (4.4.3)$$

The constant ν is user-defined in order to simulate sensible cost discrepancy between models without being too demanding when converging all MDAs for computing the true error. In the following, ν is arbitrarily set to be equal to 7 for both disciplines Sellar1 and Sellar2.

Both disciplines, Sellar1 and Sellar2, are treated equally for the undergoing study. The previous methodology can be utilized to discriminate both disciplines in terms of the availability of their respective low-fidelity models. It would have been possible to consider more asymmetrical cases, where one discipline has a greater number of low-fidelity models than the other, or where a different cost model than (4.4.3) is utilized.

Error of the fidelity models on the initial design

As the high-fidelity, noted $\Psi^{HF} = \Psi_{1,1}$, is defined for the couple $(K_1, K_2) = (1, 1)$, low-fidelity MDAs, noted Ψ_{K_1, K_2} , are generated using evenly spaced values in the interval $[0, 2]$ with a gap of 0.1 between values. Therefore 21 models are available for both disciplines Sellar1 and Sellar2. Using all the possible combinations for the MDA fidelity models, there is a total of $21^2 = 441$ models to discriminate. Figure 4.13 shows the estimated error using the coupled adjoint criterion, as-well as the analytical Taylor approximation and the true error obtained by converging all MDAs fidelity at the initial design point.

$$\epsilon^{\text{taylor}} = -\nabla_{k_1, k_2} f(x_0, x_1, x_1, \Psi_{1,1}(x_0, x_1, x_1)) \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}. \quad (4.4.4)$$

All models are considered to be solved using the non-linear Jacobi algorithm and are ranked in ascending true error. Figure 4.14 offers a new outlook of the previous results, tracing both analytical Taylor error and adjoint criterion with respect to the true error for each fidelity model.

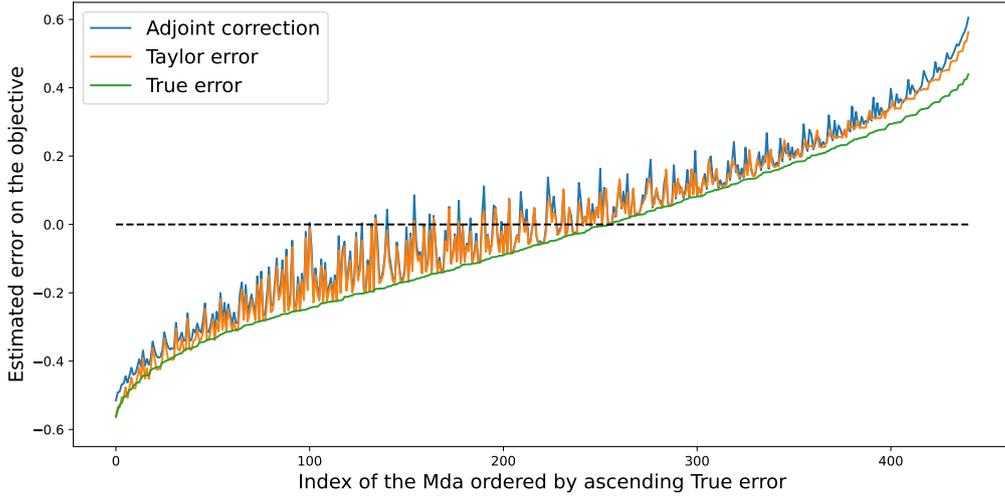


Figure 4.13: MDAs, $(K_1, K_2) \in [0, 2]^2$, ranked with the coupled adjoint criterion

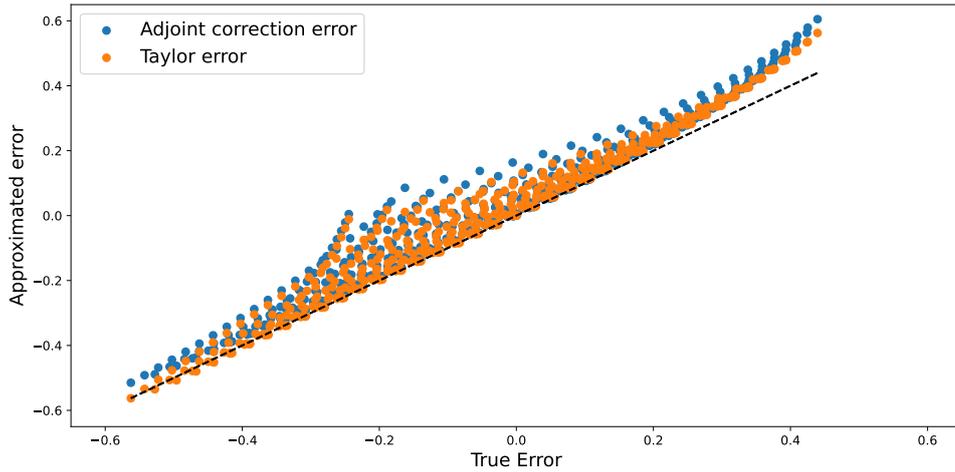


Figure 4.14: Distance from true error

Both figures show that the computed adjoint correction criterion is capturing the true error evolution trend, similarly as the first order Taylor approximation, as expected. The maximum discrepancy between the first order approximation and the true error is of the

order of 0.2, which is, considering the large amount of models to rank and the total range of 1.2, regarded as an acceptable bias. One can observe that the highest discrepancy between the predicted error and the real one appends when K_1 and/or K_2 take the more extreme values, typically below 0.3 or more than 1.8. In particular, the worst prediction is reached for $(K_1, K_2) = (0, 2)$, which, surprisingly, is far from being one of the worst models in terms of couplings perturbation, further confirming that the fidelity of combination of coupled models is not straightforward.

To illustrate that the quality of the prediction depends on how strong the perturbation is, Figure 4.15 shows the same graph but only considering the models issued from $(K_1, K_2) \in [0.5, 1.5]^2$. In this example, where extremes disciplinary perturbations are avoided, the error prediction never exceeds 0.06 and is far more reliable. Again, the worst prediction is reached when the worst fidelity models are combined, namely for $(K_1, K_2) = (0.5, 1.5)$.

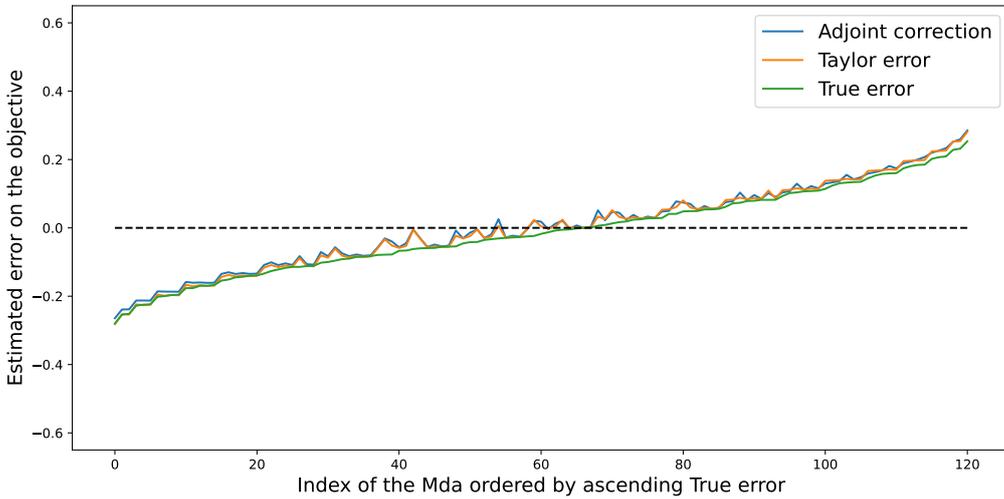


Figure 4.15: MDAs, $(K_1, K_2) \in [0.5, 1.5]^2$, ranked with the coupled adjoint criterion

Considering the computation time, results are gathered in Table 4.1, that shows the large amount of time saved for the estimation. The high-fidelity convergence takes approximately one minute for 9 iterations. The adjoint vector computation, which takes approximately 14 seconds, and the execution for each disciplinary fidelity models is a bit more expensive but still relatively small considering that the criterion is applied on 441 different models. If all MDA fidelity models needed to be converged, a simple rule of thumb can give us a good approximation of the time needed by making the small assumption that all MDAs take, on average, 9 iterations to converge (similarly as the high-fidelity model). Hence, as each disciplinary fidelity models appears in exactly 21 models, and that executing all these disciplinary models a unique time took 3.16 minutes, converging all the MDA fidelity models should take approximately $9 * 21 * 3.16 \approx 635$ minutes for the whole approximation, which represents more than 10 hours of computation.

	Elapsed time (min)
HF model convergence	0.94
LF models with Adjoint correction	3.36
LF models with True error	635.04

Table 4.1: Computation time for error estimation

Error at the optimum

The aforementioned errors were computed on the initial design point, one can ask ourselves whether these errors can be directly used as good approximations of the errors at the optimal design. In this respect, a full MDF optimization is performed for each possible model selection \mathcal{V} . Hence all 441 MDF models are optimized together and the resulting objective values $f_{\mathcal{V}}^*$ are compared to the high-fidelity one f_{HF}^* . For each model, Figure 4.16 shows the estimated error using the coupled adjoint criterion on the initial design point and the error on the objective obtained after a full optimization, i.e. $f_{\mathcal{V}}^* - f_{HF}^*$. As expected, the coupled adjoint criterion make numerous classifications whenever the fidelity on the optimal value function is considered. However, it catches the global trend of the true error on the optimal objective value, as the average error is also strictly increasing.

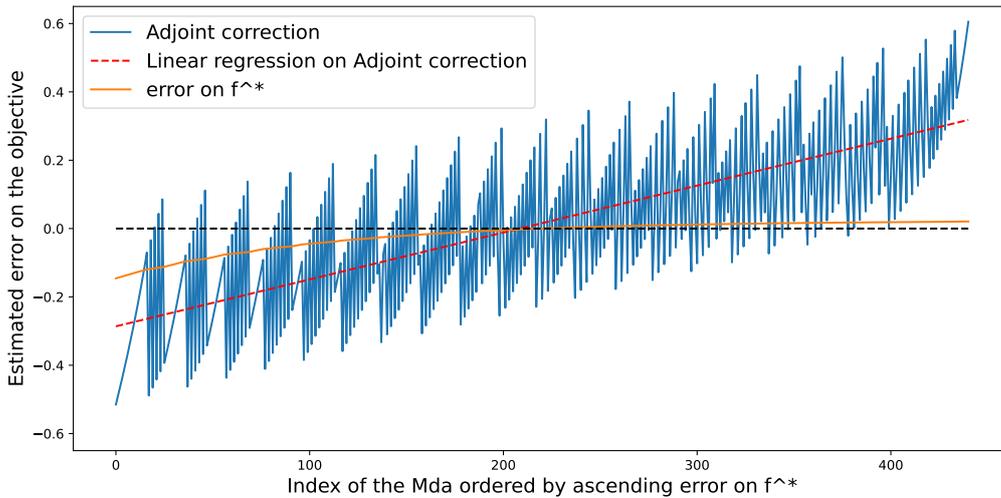


Figure 4.16: MDAs, $(K_1, K_2) \in [0, 2]^2$, ranked with the coupled adjoint criterion and error on f^*

Pareto front and model selection

Reporting both cost and error for each low-fidelity model results in the Pareto front illustrated in Figure 4.17. The obtained Pareto front, where both axis are normalized, is coherent with how the fidelity models have been constructed. The quadratic cost function used to define the cost for each fidelity is clearly observable and the high-fidelity model possesses a zero error and the highest precision. The selection algorithm was asked for the selection of 3 different Pareto front models among 10 Pareto optimal fidelity models. It should be noticed that the post-processing algorithm the lowest-fidelity model

as its neighbor is judged as having a too similar, nearly 0, cost for a slightly better precision.

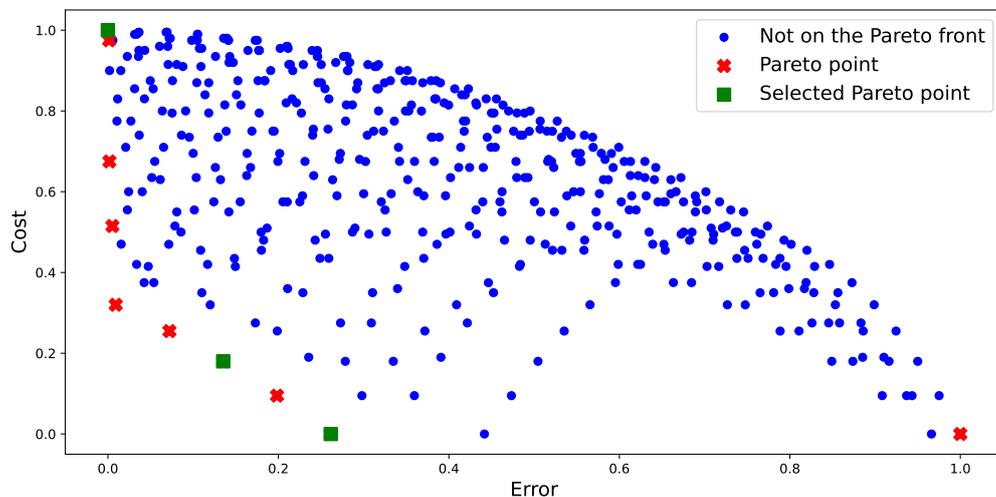


Figure 4.17: Sellar Pareto front with an approximated error using the adjoint criterion, MDAs created with $(K_1, K_2) \in [0, 2]^2$ and $\mathcal{T}_r = 0.7$

Figure 4.18 shows the true Pareto front obtained by converging all MDA fidelity models and comparing their output to the high-fidelity one. It should be noted that, similarly as with the adjoint based criterion, the lowest-fidelity level is not considered, in fact It not even on the Pareto front. On the other hand, the resulting Pareto font is more compressed, a consequence is that the post-process algorithm found out that a third fidelity model is not necessary. Hence the in-between fidelity model has been eliminated.

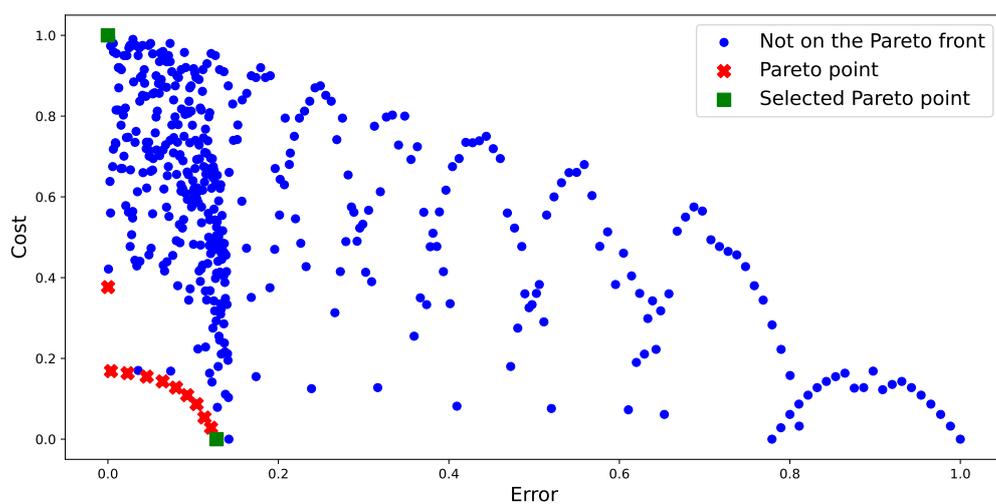


Figure 4.18: Sellar Pareto front after convergence of all models, MDAs created with $(K_1, K_2) \in [0, 2]^2$ and $\mathcal{T}_r = 0.7$

A refinement optimization

Next, the multi-fidelity refinement algorithm is applied to multiple fidelity model selections. The objective of this numerical experiment is to measure the gain, in terms of total discipline execution time, of using each considered model selection within the multi-fidelity refinement framework.

Table 4.2 shows the computed time ratio between the multi-fidelity approach and the single-fidelity approach for different selection strategies. The considered models are represented by their pair of perturbation parameters (K_1, K_2) , where the notation $(i, j) \rightarrow (i', j')$ means that the fidelity model (i', j') is optimized after model (i, j) and warm-started by the found optimal design of model (i, j) . Obviously, the high-fidelity model, represented by the pair $(1, 1)$, is always presents as the last computed model.

Among the considered strategies, two naive selections are used, where the fidelity models are chosen by selecting the extreme values of (K_1, K_2) as the lowest fidelity level and their average values for the intermediate fidelity model. Both selections shown in Figure 4.17 and Figure 4.18 are also considered based on the adjoint criterion and fully converged errors, respectively. To see if the addition of a third fidelity model has an impact on the multi-fidelity optimization, the adjoint based selection where the intermediate model is retrieved and the fully converged errors selection with 3 models, i.e. the selection before the post-process algorithm, are also considered.

Choice of the LF models	Selected models w.r.t. (K_1, K_2)	Time ratio (MF/HF)
Naive selection $n^{\circ}1$	$(0, 0) \rightarrow (0.5, 0.5) \rightarrow (1, 1)$	1.19
Naive selection $n^{\circ}2$	$(2, 2) \rightarrow (1.5, 1.5) \rightarrow (1, 1)$	1.25
3 fidelity adjoint criterion	$(0, 1.8) \rightarrow (0, 2) \rightarrow (1, 1)$	0.87
2 fidelity adjoint criterion	$(0, 1.8) \rightarrow (1, 1)$	0.87
3 fidelity true error	$(0, 2) \rightarrow (0.1, 1.1) \rightarrow (1, 1)$	1.24
2 fidelity true error	$(0, 2) \rightarrow (1, 1)$	0.83

Table 4.2: Computation time for the multi-fidelity refinement algorithm

We can observe that both naive selections have poor results, in particular that both multi-fidelity approaches are slower than using only the high-fidelity model. Both Pareto selections show a speedup over the multi-fidelity refinement approach, with a slight advantage for the true error selection, as expected. If the removal of the intermediate fidelity model from the adjoint-based selection has no effect on the recovery time of the multi-fidelity algorithm, this is not the case for the addition of a third model within the true error-based selection, which becomes much worse. This indicates that the post-process algorithm, which removes fidelity models judged to be detrimental to the multi-fidelity optimization, was beneficial.

4.4.3 Gradient alignment criterion

For the gradient alignment criterion, an application measures the impact of the discipline order of execution when backward effects are neglected on the SSBJ test case.

Impact of architecture and the disciplines order on the optimization

In the context of MDO, one potential approach to developing a low-fidelity model of the process is to disregard the backward effects of coupled disciplines. This is accomplished by initiating the computation of each discipline in a sequential order at each evaluation of the objective and constraints. This can be regarded as a one-pass Gauss-Seidel algorithm. The consequence of the absence of a loop is that the backward effects are neglected. This is because a particular discipline will not have information from the subsequent disciplines until the next optimization iteration. Consequently, the result of the process is highly influenced by the sequence of execution of the disciplines. It is therefore pertinent to consider which sequence is most suitable for the low-fidelity model to be used in a more generic multi-fidelity approach.

To illustrate let us consider the Sobieski Super Business Jet (**SSBJ**) [139] test case presented in Appendix A. This test case is composed of 3 strongly coupled disciplines, namely Structure, Propulsion and Aerodynamics and a fourth, weakly coupled, discipline Mission in charge of the computation of the objective, here, the range. Since the latter is always executed last, there are 6 possible sequence of the 3 strongly coupled disciplines and thus 6 possible low-fidelity models to consider. Figure 4.19 shows the XDSM diagram for one of the low-fidelity models, namely the one with the sequence Structure \rightarrow Propulsion \rightarrow Aerodynamics.

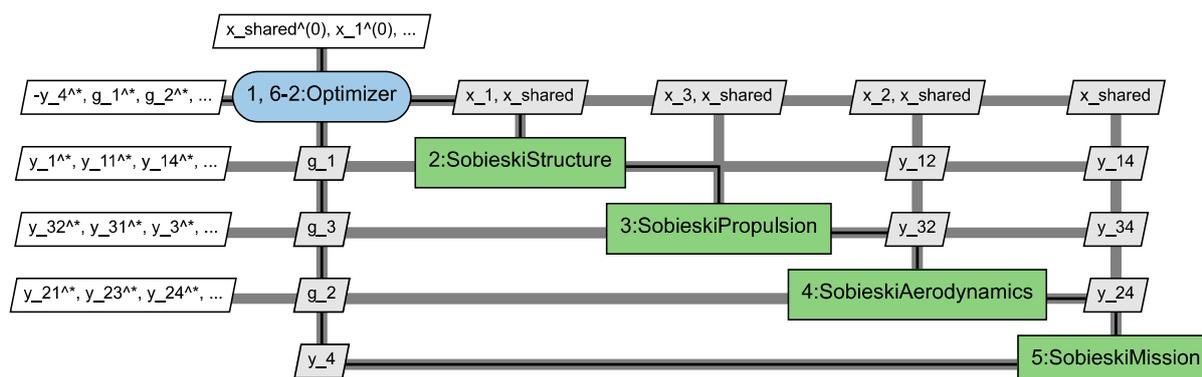


Figure 4.19: An example of XDSM diagram for SSBJ with no backward effects

The high-fidelity model, represented in Figure 4.20, is defined by the classic MDF architecture using a Gauss-Seidel MDA. It always yield a consistent solution for all designs (x_0, x) in the sense that the computation of the coupling variables, objective and constraints are not affected by the sequence of execution of the disciplines.

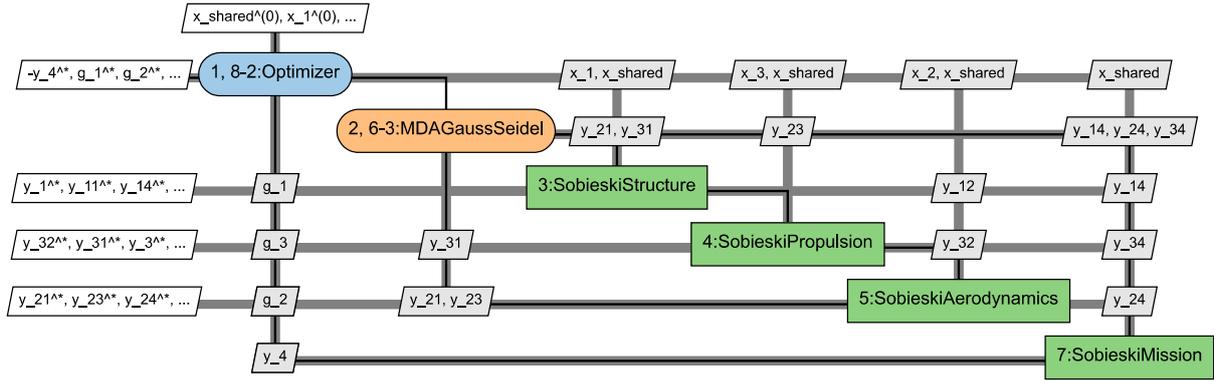


Figure 4.20: The XDSM diagram for SSBJ with a MDA based strategy using a Gauss-Seidel MDA

It is important to note that the objective is to select a single low-fidelity model for the refinement approach. Indeed, by design, every low-fidelity models have the same cost. This cost is ensured to be lower than the high-fidelity one, due to the absence of the MDA loop.

Since backward effects are neglected, and since the order of the disciplines is important for the choice of low-fidelity architecture, gradient alignment is used to approximate the error introduced by a given order of disciplines by simply applying the chain rule for each possible order. The chosen criterion takes into account the objective and the constraint g_3 , computed by the Propulsion discipline, with equal weights:

$$\epsilon_{align} = \frac{1}{2}[\epsilon_{align}^f + \epsilon_{align}^{g_3}] \quad (4.4.5)$$

Both g_1 and g_2 , computed by the Structure discipline and the Aerodynamics discipline respectively, are not considered in the criterion. This is due to the fact that these two constraints does not depend on the couplings, as a result their computation and derivatives are not affected by disregarding the backward effects. This criterion is computed on a unique point, the initial design point for the SSBJ test case (cf. Table A.1), we do not find that varying the evaluation point has a particular impact on the estimated error.

As we have few low-fidelity models, it is possible to compute the gradient alignment criterion for each one of them and perform the multi-fidelity refinement approach. To do not depend on the starting point for the optimization, a design of experiment is performed on the SSBJ's design space to generate 300 different stating point using the Latin Hypercube Sampling (LHS) method [105]. Table 4.3 shows, for each low-fidelity model, the error approximated via the gradient alignment criterion and the average ratio of the number of discipline calls between the multi-fidelity approach and the original MDF method on the 300 optimizations.

These results show that the considered criterion is efficient in giving an approximation of the fidelity ranking for the considered refinement multi-fidelity approach. Each low-fidelity model is well ranked with respect to the benefits of its inclusion in the multi-fidelity algorithm in terms of discipline calls. Considering the error estimation ϵ_{align} , three groups of models can be distinguished. The first one is uniquely represented by the sequence Structure \rightarrow Aerodynamics \rightarrow Propulsion and has the lowest estimated error with an

Choice of the LF model	ϵ_{align}	Time ratio (MF/HF)
Structure \rightarrow Aerodynamics \rightarrow Propulsion	$4.94 * 10^{-5}$	0.46
Aerodynamics \rightarrow Structure \rightarrow Propulsion	$1.82 * 10^{-3}$	0.53
Aerodynamics \rightarrow Propulsion \rightarrow Structure	$2.08 * 10^{-3}$	0.82
Structure \rightarrow Propulsion \rightarrow Aerodynamics	$4.67 * 10^{-1}$	0.92
Propulsion \rightarrow Structure \rightarrow Aerodynamics	$4.67 * 10^{-1}$	0.98
Propulsion \rightarrow Aerodynamics \rightarrow Structure	$4.69 * 10^{-1}$	0.95

Table 4.3: Computation time for the multi-fidelity refinement algorithm

order of 10^{-5} . It also provides the best benefits for the multi-fidelity approach, the total number of discipline calls being cut-off by more than a half. The second group gather the two last sequences, both starting with the Aerodynamics discipline, with an estimated error or the order of 10^{-3} . The criterion made no mistakes, as the sequence Aerodynamics \rightarrow Structure \rightarrow Propulsion is well considered as a better sequence of execution than the other one, but the difference between the two error does not represent the gap between the potential gain for the multi-fidelity algorithm. Indeed, even if ϵ_{align} has the same order for both sequences, the total number of discipline calls saved varies greatly, passing from 47% calls saved to only 18%. Nonetheless they are both better than the low-fidelity models within group 3. This last group is composed of the remaining models with an estimated error or the order of 10^{-1} . These tree sequences exhibit less than 10% discipline calls saved, and their relative ordering is not well-approximated by the gradient alignment criterion.

To summarize, the presented criterion allows to distinguished three groups of models in terms of benefits for the multi-fidelity approach, all groups are easily ranked from the worst to the best. Discrimination within a same group, however, are less reliable, ranking of the models is subject to errors and interpretation of the benefits of the inclusion of a model compared to another is unclear. This is coherent with the reason behind the existence of this criterion and the expected behavior: having a rough approximations with minor errors for closely related models, but the global ranking for all the considered models is kept unchanged.

4.4.4 A multi-fidelity application using bi-level architectures

This section contains a multi-fidelity optimization on the previously introduced SSBJ test case [139], utilizing the BL-BCD-MDF architecture, presented in Chapter 3, as a high-fidelity model of the BL-IRT architecture.

The objective of this application is to illustrate the capabilities and the flexibility of the methodologies, and their implementation within GEMSEO, to quickly build complex MDO processes.

Definition of the process

Considering the SSBJ test case [139], the goal of this application is to build a multi-fidelity refinement approach using the newly introduced BL-BCD-MDF architecture introduced in Chapter 3 as a high-fidelity model and the BL-IRT architecture as a low-fidelity model. The main argument for considering the BL-IRT architecture as a low-fidelity model of the

BL-BCD-MDF is its lower restitution time for the lower-level optimization problem, which is obtained at the cost of the discrepancy error on the system functions as highlighted in Chapter 3.

Table 4.4 shows the differences between the high and low-fidelity models. As shown, the system level of the BL-BCD-MDF is solved using NLOPT’s BOBYQA [121] algorithm. This is a DFO algorithm based on quadratic approximations of the objective, which we believe works better with the BL-BCD-MDF architecture than with the BL-IRT because of the discrepancy error reduction. Since BOBYQA does not handle constraints natively, they are added to the system objective function under a penalized form with a fixed external penalty factor. Because of the discrepancy error, NLOPT’s COBYLA [121] algorithm is considered for the system optimization problem, since the linear approximations of the functions used are less sensitive to noise. All block optimizations are solved using NLOPT’s SLSQP algorithm. An MDF-based strategy using a Gauss-Seidel MDA is implemented within the block optimization of the BL-BCD-MDF architecture.

The only difference between the BL-BCD-MDF architecture strategy used in the multi-fidelity framework and that used in the mono-fidelity optimization is the *init_step* parameter, which controls the range for the initial DOE that builds the approximations. The reason for this change is that the high-fidelity model is assumed to start closer to the optimum in the multi-fidelity framework, since the low-fidelity optimization has already been performed. For this reason, the *init_step* is lowered to $1e-2$ to avoid searching in uninteresting regions, otherwise there would be a high chance that the initial DEO would degrade the solution found by the low-fidelity optimization and thus we would lose the interest of the multi-fidelity approach.

	High-fidelity model	Low-fidelity model
Architecture name	BL-BCD-MDF	BL-IRT
System optimizer	BOBYQA	COBYLA
Disciplinary block optimizer	SLSQP	SLSQP
MDA within block optimizations	YES (Gauss-Seidel)	NO
MDA before block optimizations	YES (Jacobi)	YES (Jacobi)
MDA after block optimizations	NO	YES (Jacobi)
<i>Init_step</i>	0.25 (MONO), $1e^{-2}$ (MF)	0.25

Table 4.4: A summary of the differences between the high-fidelity model and the low-fidelity model.

A comparison between the mono-fidelity optimization and the multi-fidelity approach

First, we run the high-fidelity model as a standalone, which gives us a mono-fidelity optimization for comparison with the multi-fidelity framework. The evolution of the objection with respect to the system iterations is illustrated with Figure 4.21.

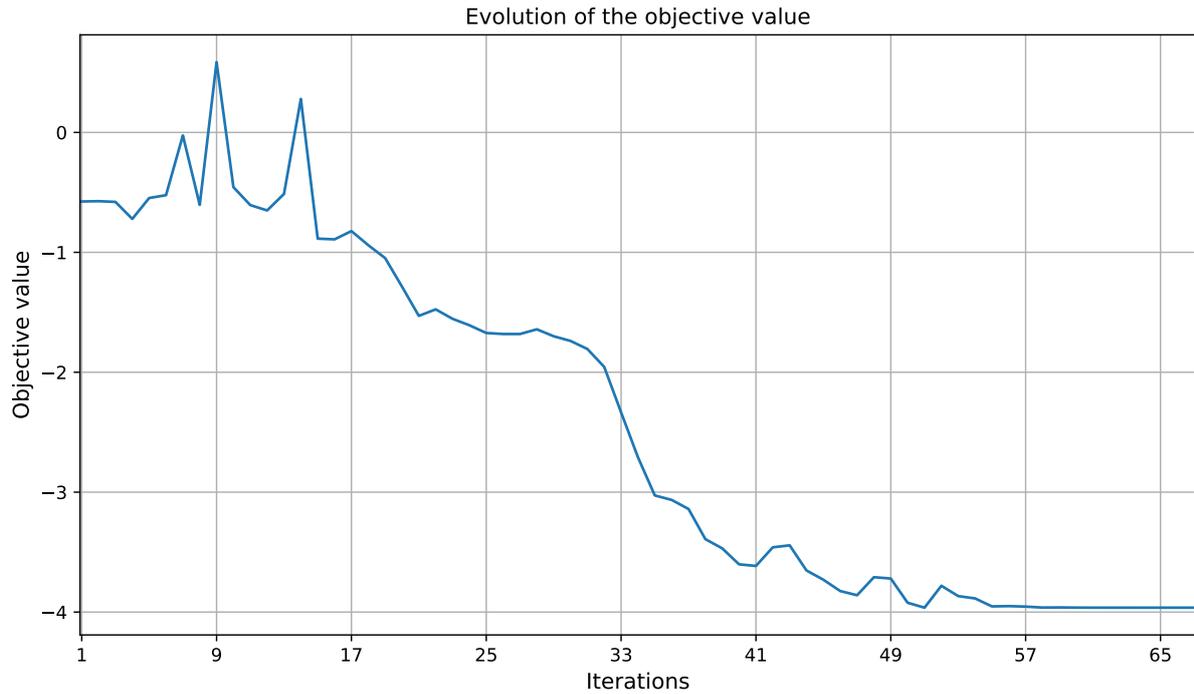


Figure 4.21: The reference mono-fidelity optimization on SSBJ using the BL-BCD-MDF architecture

For the multi-fidelity optimization, we again consider the multi-fidelity refinement algorithm. Figure 4.22 shows the evolution of the objective function for the IRT-BL optimization, while Figure 4.23 illustrates the system objective of the high-fidelity model warm-started by the low-fidelity's found solution.

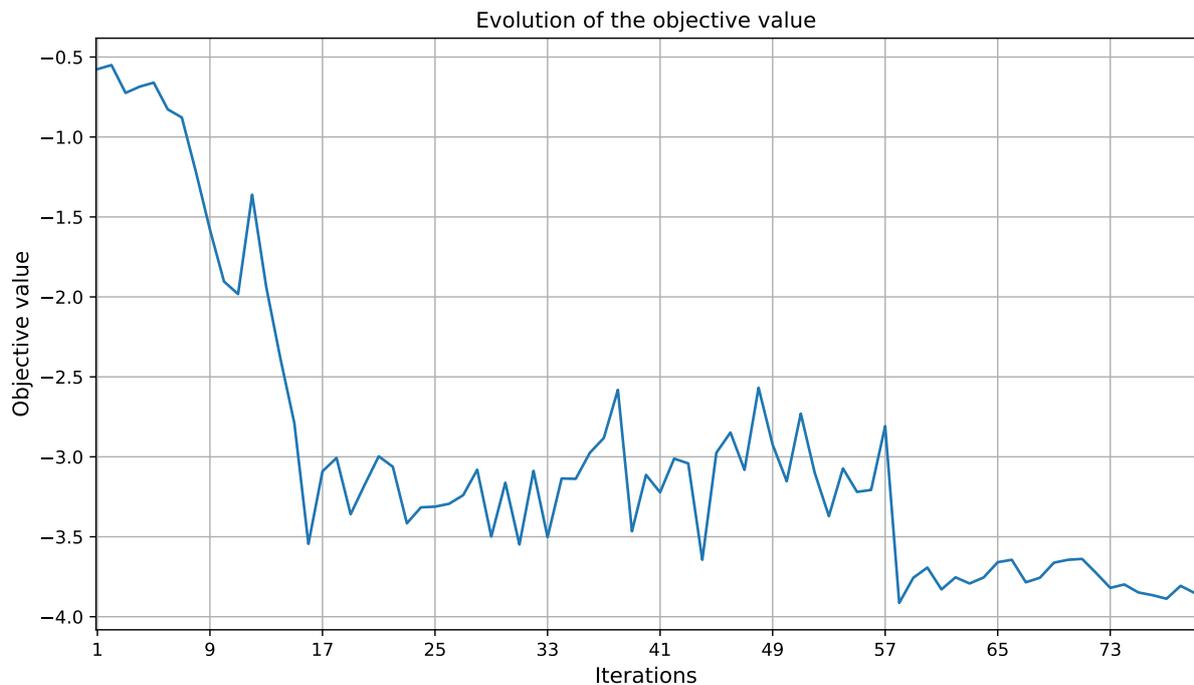


Figure 4.22: The low-fidelity optimization on SSBJ using the BL-IRT architecture within the multi-fidelity framework

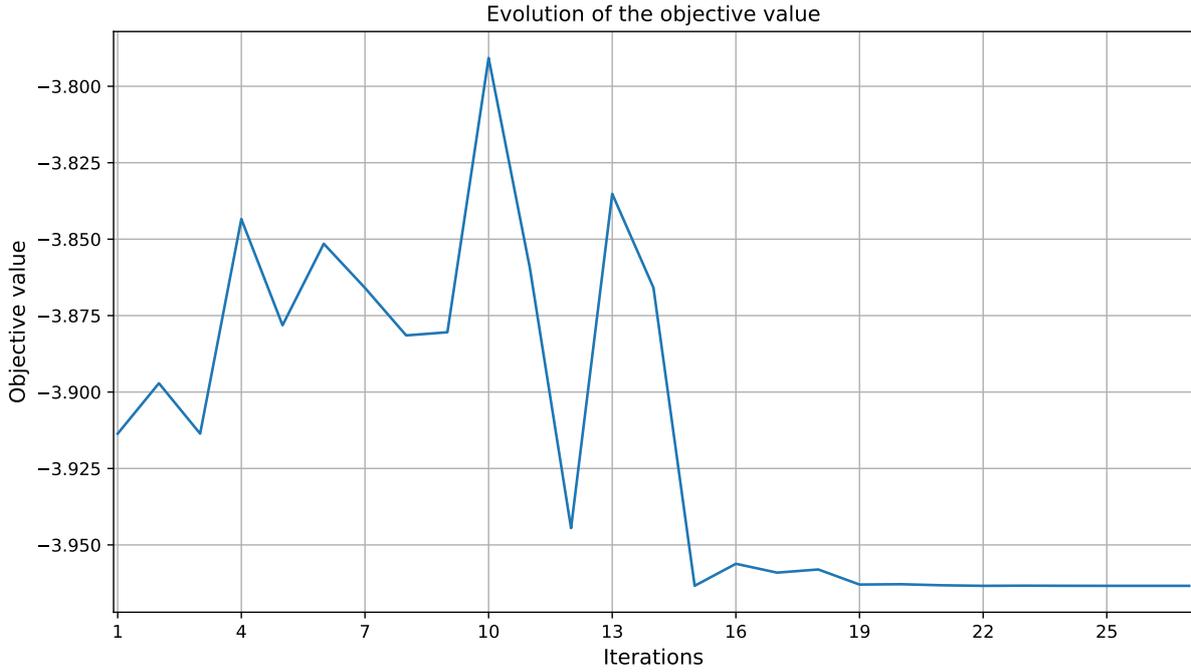


Figure 4.23: The high-fidelity optimization on SSBJ using the BL-BCD-MDF architecture within the multi-fidelity framework

Both the mono-fidelity and multi-fidelity approaches reached the correct solution for $f^* = 3963$. The mono-fidelity optimization, which only runs the high-fidelity model, reached the correct solution with a total of 22235 discipline calls, while the multi-fidelity algorithm only requires a total of 11542 discipline calls, split between the high-fidelity and low-fidelity models, to reach the same accuracy. This represents a reduction of approximately 48% in discipline calls saved by using the multi-fidelity framework. It should be noted that the number of discipline calls does not take into account the parallelization of the block optimization within the BL-IRT architecture, and therefore serves as a lower bound on the time savings.

Chapter 5

Conclusions and perspectives

The present manuscript contains several contributions related to methodologies for high-dimensional MDO. These contributions are related to the two main thrusts of this thesis: first, the bi-level MDO architectures and their convergence properties, and second, an efficient fidelity model selection for the application of multi-fidelity algorithms in MDO. This concluding chapter first summarizes the various contributions in each of these two areas, and then discusses the prospects for future research and improvements.

5.1 Contributions

Following the chapter structure, the various contributions in this manuscript can be divided into two subsets.

A new bi-level MDO architecture, namely the BL-BCD-MDF architecture, inspired by previous works from *Sobieski*, with the BLISS architecture [139], ONERA [2, 84, 131] and IRT Saint Exupéry [55], is proposed in the context of distributed architectures for high-dimensional MDO. First, the present manuscript presents a new mathematical formalization of the IRT bi-level decomposition and a mathematical framework for studying it. This bi-level architecture assumes that the number of shared variables is limited and a gradient-free optimization algorithm can be used at the system level, as opposed to the local variables, more numerous, that must be managed with the gradients within the so-called lower optimization problem. In this framework, it is shown that the correct behavior of the architecture, in terms of regularity of the system function and hence convergence, depends first and foremost on the correct solution of the subproblem, a guarantee that the aforementioned architecture cannot provide. In order to enforce such behavior by solving the lower level as precisely as possible, a new solution algorithm is proposed, the Block Coordinate Descent (BCD) algorithm, which, combined with the bi-level framework, gives rise to the aforementioned BL-BCD-MDF architecture. The BCD algorithm respects the specifications given in the preamble, i.e. it is based on a block decomposition close to the disciplinary decomposition, where gradient information can be used, allows the resolution of a high-dimensional lower level, and does not require the coupled adjoint. Several properties of the said algorithm are shown, among which the main result of this chapter is the establishment of a proof of the local convergence of the BCD algorithm under certain assumptions considered to be industrially justifiable, namely the Unique Block Minimizer (UBM) assumption and the First Order Separability

(FOS) hypothesis. This makes the entire bi-level architecture locally convergent for all derivative-free algorithms that only require functions to be continuously differentiable to converge. The same proof philosophy could be used to show the convergence of other bi-level architectures, under the hypothesis that a solution algorithm is provided that solves the lower-level optimization problem with sufficient accuracy. The verification of these assumptions is not an easy task, the Unique Block Minimizer (UBM) assumption could be tested, to some extent by multi-starting the block optimizations, and the First Order Separability (FOS) hypothesis would typically necessitate the coupled adjoint on the MDF solution and thus can only be known for applications where the solution is already known. Considering this, several variants, three in number, are proposed to allow flexibility in the assumptions and in the construction of the architecture. These variants are presented here to allow a trade-off between the convergence properties of the chosen architecture and the computational burden of extending these assumptions. The resulting architecture, the BL-BCD-MDF architecture, and one of its variants, the BL-BCD-WK architecture, are compared with the IRT-BL architecture and a gradient-free MDF architecture on several test cases. First results on the classical SSBJ test case show that the newly defined architectures indeed increase the regularity of the functions by reducing the noise induced by the distributed optimization of the blocks within the solution of the lower-level optimization problem. One of the consequences is a much better robustness with respect to the starting point of the BCD-based architectures compared to the BL-IRT, as demonstrated by the benchmark. The same benchmark also emphasizes that these architectures are very sensitive to hyperparameters such as the tolerances on the BCD loop or on the block convergence; further research on this topic should be pursued, in particular towards more adaptive approaches. Finally, a 2-block modified scalable Sellar test case was introduced and considered for scalability comparison. As expected, the bi-level architectures show much better scalability as the number of local variables, coupling variables, and constraints increases than the gradient-free MDF architecture.

With respect to multi-fidelity MDO methods, the choice has been made to study the classification of fidelity models, whose number of possibilities follows a combinatorial explosion, in order to select the most appropriate fidelity models for a multi-fidelity refinement approach. The main contribution of this chapter is the definition, the mathematical formalization, the hypothesis study as well as the validation on several applications, of two sensitivity criteria, an adjoint based one and a gradient based one, to quickly estimate the fidelity of multiple fidelity MDO models. Since both criteria use local information to estimate fidelity, they are both subject to misclassification compared to more computationally intensive methods that rely heavily on statistics. However, these two criteria are both much cheaper to compute and are still expected to outperform methods based on heuristics. In this respect, they represent an intermediate step between generally unreliable heuristics based on intuition and more precise but also too costly estimation methods considering the large number of models to be ranked. The hypothesis is that the misclassifications occur for closely related models, and that the choice of one model over the others should not affect a generic multi-fidelity approach too much. Furthermore, a methodology to construct a (*cost, error*) Pareto front and to down-select the most interesting fidelity models is presented, as well as a post-processing algorithm to eliminate irrelevant models, i.e. models that offer a cost-precision ratio that is not considered sufficiently interesting compared to the preceding and subsequent fidelity models in the hierarchy. The above hypotheses

and expected behaviors are observed by the applications shown in the manuscript. All considered applications show improved performance by using the multi-fidelity refinement approach with the fidelity selection described in the manuscript. The ranking given by the criteria is indeed noisy, as a consequence the "best" fidelity selection, in the sense that it provides the best speedup for the considered multi-fidelity optimization, may be missed. However, the resulting classification always yields a satisfactory fidelity model selection, where the selected fidelity models are not far from the real best fidelity selection in terms of performance under multi-fidelity optimization, and the worst fidelity models are avoided. It is confirmed that these two criteria represent an interesting trade-off between computational time and gain from the multi-fidelity refinement approach. First, we introduced a multi-fidelity Sellar application where the fidelity of each discipline can be controlled by a user-defined parameter. We were able to effectively classify 441 different fidelities of MDF architectures for the Sellar problem with the adjoint criterion, yielding a small speed-up of the multi-fidelity over the mono-fidelity for a small computational cost. Second, by using the gradient-based criterion, we were able to reduce by more than half the total number of discipline calls required to compute the optimal design of the SSBJ. This was done by warm-starting a high-fidelity MDF architecture with the solution of a well-chosen sequence of discipline computations, where backward effects are neglected.

These two contributions led to the development of a dedicated package within the open source (GEMSEO) [53] python library. The bi-level package contains implementations of both the BL-BCD-MDF and BL-BCD-WK architectures. An effort has been made to make the implementations more automated than other in-house implementations, in the sense that the bi-level and disciplinary decompositions, as well as the assignment of design variables/constraints, are fully automated. This package also includes the block-modified Sellar problem described in Section 3.6.2, which allows a two-block decomposition of the disciplinary level of generic bi-level architectures for testing purposes, as well as its scalable version for the scalability study discussed above. The multi-fidelity criteria package regroups the two sensitivity criteria, a Pareto front modification that allows the fidelity selection presented in the manuscript, as well as the multi-fidelity Sellar problem. Within the framework of this thesis, there was the opportunity to update the previous multi-fidelity package originally developed by *Romain Olivanti's* [108] in Python2 and compatible with GEMS V3, to the latest version of GEMSEO. The extension and porting of this code is now fully compatible with python3 and the current version of GEMSEO, including the generic multi-fidelity refinement approach used in this manuscript, as well as all the tools and methods mentioned in his thesis [108], such as trust-region mechanisms, Hessian approximations, or asynchronous validation. As it is, it is possible to benefit from all the latest features of GEMSEO in terms of workflow generation and data exchange, as well as the implementation of all the work described in this manuscript. Therefore, a great flexibility is allowed on the approaches considered, possibly combining all these methodologies to treat different applications.

5.2 Perspectives

Even if the newly defined bi-level architecture with the Block Coordinate Descent algorithm, the BL-BCD-MDF architecture, represents a step towards a more sophisticated distributed strategy in which no separability is imposed on functions, the assumptions made in this

manuscript for the local convergence result are, however, not given in many applications and hard to verify *a priori*, i.e. without any prior knowledge of the targeted application and the desired disciplinary division. These assumptions can be tested to some extent, e.g. by multi-starting the block optimizations for the UBM assumption, or, if coupled derivatives are available and several high-fidelity optimizations can be preformed, one can test the FOS hypothesis for several parameterizations of the lower-level optimization problem. We believe that a smaller set of assumptions for the local convergence result may be found in the future, especially since several experiments on applications that do not verify these hypotheses still find the expected local optima. In other words, these hypotheses are shown to be sufficient, but not necessarily necessary. Another limitation of the BCD algorithm is that a fixed point of the BCD operator is not necessarily a local optimum for the lower optimization problem. More importantly, the presented architectures do not handle infeasible blocks, i.e. whenever a block optimization is unable to find a feasible point, the process continues, but all guarantees of local convergence and feasibility are lost. Switching to a least-squares minimization of the constraint violation could be considered as a lesser evil, the block optimization thus returns the least infeasible point. Alternatively, relaxation and/or penalty methods can be considered to prevent the whole process from failing. It has also been observed that the described architectures are strongly influenced by the tolerances of both the block optimizer and the BCD loop, which affect both the convergence to the optima and the time required to solve the suboptimization problem. One solution may be to incorporate strategies based on similar architectures, such as those used by *Tosseram et al.* [146, 148] in their own BCD loop, where the tolerances are tightened as the optimization proceeds, allowing for faster initial system iterations where accuracy may not be essential, depending on the optimization algorithm. Finally, experiments should be conducted on more complex real-world problems within the targeted applications, including high-dimensional/high-fidelity aerospace applications [78], as well as a comparison with other classical distributed architectures such as BLISS98 [139], or the Augmented Lagrangian coordination method proposed by *Tosseram and al.* [146, 148]. The question arises of the limitations of the approach of this thesis with respect to the limitations:

- The proof of convergence is given with hypotheses that are difficult to test on real cases.
- Numerical experiments are performed on the SSBJ case, which is a standard benchmark but may have different properties from the real cases in question, so the robustness gains may not be found in real applications. However, our experience is that it is surprisingly representative of real multilevel applications, which makes benchmarking algorithms on this test case a good compromise given the relatively low cost of the disciplines involved.

Further tests have been carried out at IRT on high-fidelity aerospace cases, and gains in convergence robustness have been observed with the BL-BCD architecture, particularly with regard to the starting point, compared with the BL-IRT architecture. This confirms the overall relevance of the approach.

The contributions on multi-fidelity criteria for the down-selection of fidelity models are currently promising, but must still be considered as preliminary results. Further validation tests need to be conducted by varying the application cases. For example, it is not clear whether these criteria are applicable and/or relevant to classify more complex fidelity

models, such as a change in architecture as seen in [156], or whether the resulting selection is relevant to other local multi-fidelity optimization methods than the considered multi-fidelity refinement approach [108], such as those based on trust region mechanisms [5, 61]. Furthermore, it may be interesting to consider more design points within the available computational resources for the computation of the two criteria, as was considered during the SSBJ application within Section 4.4.3. It is expected to reduce the ranking error when the fidelity of the considered models varies over the entire definition space. Of course, by considering more design points and averaging the error, the cost of computing the criteria increases linearly, leading to a trade-off between sufficient accuracy and an affordable computational budget. The question of how many points to consider, their impact on the classification, and the resulting multi-fidelity optimization is completely open.

Finally, in continuity with the work started in Section 4.4.4, it is envisaged to pursue strategies that combine all that has been seen in this manuscript. Indeed, since both contributions aim at facilitating the resolution of high-fidelity/high-dimensional MDO problems with numerous disciplines and coupling variables, it is expected that improvements can emerge from a multi-fidelity approach in which MDO architectures are considered as different fidelity models. In particular, distributed architectures that exhibit more coordination and convergence problems than their monolithic counterparts, such as the bi-level ones presented in this thesis, can be considered as highly adaptable low-fidelity models. Especially since distributed architectures, by nature, exhibit many design choices at different levels:

- The choice of the fidelity for each discipline.
- Choices related to organizational aspects, such as the choice of the coordination mechanisms or on the order of execution of the different models.
- The choice of the, potentially numerous, hyperparameters settings, such as the different tolerances.

It would allow many low-fidelity architectures to be defined, and then further classified and down-selected by adapting the methodology presented in Chapter 4 to account for this wider range of fidelity sources. Finally, an already known robust monolithic architecture, such as MDF, can be considered as a high-fidelity model to ensure convergence to the correct optimal design at the end of the multi-fidelity process.

Considering the development within GEMSEO, these implementations allow faster and easier experiments. Since GEMSEO is already used in industry, and the included algorithms are well integrated and validated, they are now available for testing on more representative industrial problems.

Bibliography

- [1] Mark A Abramson and Charles Audet. Convergence of mesh adaptive direct search to second-order stationary points. *SIAM Journal on Optimization*, 17(2):606–619, 2006. [50](#), [51](#)
- [2] A Bi-Level High Fidelity Aero. Methodology-a focus on the structural sizing process. [9](#), [137](#)
- [3] Edward Albano and William P Rodden. A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows. *AIAA journal*, 7(2):279–285, 1969. [63](#)
- [4] Natalia M Alexandrov, Robert Michael Lewis, Clyde R Gumbert, Lawrence L Green, and Perry A Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, 38(6):1093–1101, 2001. [105](#)
- [5] Dennis J. E. Lewis R. M. Torczon V. Alexandrov N. M. A trust-region framework for managing the use of approximation models in optimization. 1998. [101](#), [104](#), [105](#), [141](#)
- [6] Douglas Allaire and Karen Willcox. A mathematical and computational framework for multifidelity design and analysis with computer models. *International Journal for Uncertainty Quantification*, 4(1), 2014. [97](#), [104](#), [107](#), [108](#)
- [7] Mohamed Amine Bouhlel, Nathalie Bartoli, Rommel G Regis, Abdelkader Otsmane, and Joseph Morlier. Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Engineering Optimization*, 50(12):2038–2053, 2018. [108](#)
- [8] Holt Ashley. On making things the best-aeronautical uses of optimization. *Journal of Aircraft*, 19(1):5–28, 1982. [7](#), [15](#)
- [9] Mathieu Balesdent, Nicolas Bérend, Philippe Dépincé, and Abdelhamid Chriette. A survey of multidisciplinary design optimization methods in launch vehicle design. *Structural and Multidisciplinary optimization*, 45:619–642, 2012. [7](#), [8](#), [15](#), [18](#), [19](#), [42](#), [47](#)
- [10] Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta mathematicae*, 3(1):133–181, 1922. [57](#)
- [11] Arnaud Barthet. *Amélioration de la prévision des coefficients aérodynamiques autour de configurations portantes par méthode adjointe*. PhD thesis, 2007. Thèse de

- doctorat dirigée par Braza, Marianna et Airiau, Christophe Mécanique des fluides Toulouse, INPT 2007. [119](#)
- [12] Arnaud Barthet, Christophe Airiau, Marianna Braza, and Loïc Tourrette. Adjoint-based error correction applied to far-field drag breakdown on structured grid. In *24th AIAA Applied Aerodynamics Conference*, page 3315, 2006. [119](#)
- [13] Amir Beck and Luba Tretushvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013. [69](#)
- [14] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015. [102](#)
- [15] Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015. [45](#), [66](#)
- [16] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997. [10](#), [27](#), [40](#), [45](#), [66](#), [67](#)
- [17] James C Bezdek and Richard J Hathaway. Some notes on alternating optimization. In *AFSS international conference on fuzzy systems*, pages 288–300. Springer, 2002. [10](#), [27](#), [40](#), [45](#), [66](#)
- [18] James C Bezdek, Richard J Hathaway, DB Fogel, and CJ Robinson. Two new convergence results for alternating optimization. *Computational Intelligence: The Experts Speak*, pages 149–164, 2003. [10](#), [27](#), [40](#), [45](#), [66](#)
- [19] James C Bezdek, Richard J Hathaway, Michael J Sabin, and William T Tucker. Convergence theory for fuzzy c-means: counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(5):873–877, 1987. [69](#)
- [20] Nicolas P Bons and Joaquim RRA Martins. Aerostructural design exploration of a wing in transonic flow. *Aerospace*, 7(8):118, 2020. [110](#)
- [21] Mohamed Amine Bouhleb, Nathalie Bartoli, Abdelkader Otsmane, and Joseph Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53:935–952, 2016. [9](#)
- [22] Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations research*, 21(1):37–44, 1973. [51](#)
- [23] Robert Braun, Peter Gage, Ilan Kroo, and Ian Sobieski. Implementation and performance issues in collaborative optimization. In *6th symposium on multidisciplinary analysis and optimization*, page 4017, 1996. [43](#)
- [24] Robert David Braun. *Collaborative optimization: an architecture for large-scale distributed design*. Stanford University, 1996. [39](#), [43](#)
- [25] Jasper H Bussemaker, Nathalie Bartoli, Thierry Lefebvre, Pier Davide Ciampa, and Björn Nagel. Effectiveness of surrogate-based optimization algorithms for system architecture optimization. In *AIAA Aviation 2021 forum*, page 3095, 2021. [104](#)

- [26] W. Candler and R.D. Norton. *Multi-level Programming*. Discussion Papers, Development Research Center, International Bank for Reconstruction and Development. World Bank, 1977. [52](#)
- [27] Marco Carini, Christophe Blondeau, Nicolo Fabbiane, Michael Meheut, Mohammad Abu-Zurayk, Johan M Feldwisch, Caslav Ilic, and Andrei Merle. Towards industrial aero-structural aircraft optimization via coupled-adjoint derivatives. In *AIAA Aviation 2021 Forum*, page 3074, 2021. [50](#)
- [28] Remy Charayron, Thierry Lefebvre, Nathalie Bartoli, and Joseph Morlier. Pareto optimal fidelity level selection for multi-fidelity bayesian optimization applied to drone design. In *12 th EASN International conference on " Innovation in Aviation & Space for opening New Horizons"*, 2022. [10](#), [98](#), [112](#)
- [29] Daniel Erik Christensen. *Multifidelity methods for multidisciplinary design under uncertainty*. PhD thesis, Massachusetts Institute of Technology, 2012. [97](#), [108](#)
- [30] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153:235–256, 2007. [55](#), [56](#)
- [31] Jean-Pierre Costa, Luc Pronzato, and Eric Thierry. A comparison between kriging and radial basis function networks for nonlinear prediction. In *NSIP*, pages 726–730, 1999. [101](#), [103](#)
- [32] Evin J. Cramer, J. E. Dennis, Jr., Paul D. Frank, Robert Michael Lewis, and Gregory R. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization*, 4(4):754–776, November 1994. [7](#), [18](#), [19](#), [20](#), [21](#)
- [33] Albert de Wit and Fred van Keulen. Numerical comparison of multi-level optimization techniques. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1895, 2007. [46](#)
- [34] Scott Delbecq, Marc Budinger, and Aurélien Reysset. Benchmarking of monolithic mdo formulations and derivative computation techniques using openmdao. *Structural and Multidisciplinary Optimization*, 62(2):645–666, 2020. [45](#), [104](#)
- [35] Jean Demange, A Mark Savill, and Timoleon Kipouros. A multifidelity multiobjective optimization framework for high-lift airfoils. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 3367, 2016. [105](#)
- [36] Victor DeMiguel and Walter Murray. A local convergence analysis of bilevel decomposition algorithms. *Optimization and Engineering*, 7:99–133, 2006. [45](#)
- [37] Stephan Dempe. An implicit function approach to bilevel programming problems. In *Multilevel Optimization: Algorithms and Applications*, pages 273–294. Springer, 1998. [55](#)
- [38] Stephan Dempe. *Bilevel optimization: theory, algorithms and applications*, volume 3. TU Bergakademie Freiberg, Fakultät für Mathematik und Informatik Freiberg . . . , 2018. [56](#)
- [39] Stephan Dempe and Joydeep Dutta. Is bilevel programming a special case of a mathematical program with complementarity constraints? *Mathematical programming*, 131:37–48, 2012. [54](#), [55](#), [57](#)

- [40] Stephan Dempe, Vyacheslav Kalashnikov, Gerardo A Pérez-Valdés, and Nataliya Kalashnykova. Bilevel programming problems. *Energy Systems. Springer, Berlin*, 10:978–3, 2015. [53](#), [56](#), [57](#)
- [41] Stephan Dempe and Alain B Zemkoho. The generalized mangasarian-fromowitz constraint qualification and optimality conditions for bilevel programs. *Journal of optimization theory and applications*, 148:46–68, 2011. [53](#)
- [42] Stephen Dempe, Nguyen Dinh, and Joydeep Dutta. Optimality conditions for a simple convex bilevel programming problem. *Variational Analysis and Generalized Differentiation in Optimization and Control: In Honor of Boris S. Mordukhovich*, pages 149–161, 2010. [55](#)
- [43] JE Dennis and Virginia Torczon. Managing approximation models in optimization. *Multidisciplinary design optimization: State-of-the-art*, 5:330–347, 1997. [105](#)
- [44] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91:201–213, 2002. [90](#)
- [45] Ali Elham and Michel JL van Tooren. Multi-fidelity wing aerosturctural optimization using a trust region filter-sqp algorithm. *Structural and Multidisciplinary Optimization*, 55(5):1773–1786, 2017. [9](#), [105](#)
- [46] M Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*, 2016. [9](#), [97](#), [100](#), [102](#), [108](#)
- [47] Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical programming*, 10(1):287–311, 1976. [8](#), [34](#), [56](#), [61](#)
- [48] Joshua E Fontana, Pat Piperni, Zhi Yang, and Dimitri J Mavriplis. Aerostructural wing optimization using a structural surrogate in a coupled adjoint formulation. *AIAA Journal*, pages 1–18, 2024. [9](#), [107](#)
- [49] Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1-3):50–79, 2009. [101](#), [103](#), [107](#)
- [50] Alexander I.J Forrester, András Sóbester, and Andy J Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, December 2007. Publisher: Royal Society. [101](#), [103](#), [107](#)
- [51] Roy Frostig, Matthew James Johnson, and Chris Leary. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 4(9), 2018. [8](#)
- [52] François Gallard. *Aircraft shape optimization for mission performance*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2014. [118](#)
- [53] Francois Gallard, Charlie Vanaret, Damien Guenot, Vincent Gachelin, Rémi Lafage, Benoit Pauwels, Pierre-Jean Barjhoux, and Anne Gazaix. GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, January 2018. American Institute of Aeronautics and Astronautics. [11](#), [124](#), [139](#)

- [54] Nicolas Garland, Le Riche Rodolphe, Richet Yann, and Nicolas Durrande. Multi-fidelity for mdo using gaussian processes. 2020. [97](#), [108](#)
- [55] Anne Gazaix, François Gallard, Vincent Ambert, Damien Guénot, Maxime Hamadi, Stéphane Grihon, Patrick Sarouille, Thierry Y Druot, Joël Brézillon, Vincent Gachelin, et al. Industrial application of an advanced bi-level mdo formulation to aircraft engine pylon optimization. In *AIAA Aviation 2019 Forum*, page 3109, 2019. [9](#), [12](#), [40](#), [49](#), [91](#), [108](#), [137](#)
- [56] Anne Gazaix, Francois Gallard, Vincent Gachelin, Thierry Druot, Stéphane Grihon, Vincent Ambert, Damien Guénot, Rémi Lafage, Charlie Vanaret, Benoit Pauwels, Nathalie Bartoli, Thierry Lefebvre, Patrick Sarouille, Nicolas Desfachelles, Joel Brézillon, Maxime Hamadi, and Selime Gurol. Towards the Industrialization of New MDO Methodologies and Tools for Aircraft Design. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, Colorado, June 2017. American Institute of Aeronautics and Astronautics. [39](#), [41](#), [47](#), [48](#), [62](#), [63](#), [64](#)
- [57] Arthur M Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972. [39](#)
- [58] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005. [88](#)
- [59] Stefan Goertz, Caslav Ilic, Jonas Jepsen, Martin Leitner, Matthias Schulze, Andreas Schuster, Julian Scherer, Richard Becker, Sascha Zur, and Michael Petsch. Multi-level mdo of a long-range transport aircraft using a distributed analysis framework. In *18th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 4326, 2017. [8](#), [31](#)
- [60] Robert B Gramacy and Herbert KH Lee. Adaptive design and analysis of super-computer experiments. *Technometrics*, 51(2):130–145, 2009. [104](#)
- [61] Serge Gratton, Mélodie Mouffe, Annick Sartenaer, Philippe L. Toint, and Dimitri Tomanos. Numerical experience with a recursive trust-region method for multilevel nonlinear bound-constrained optimization. *Optimization Methods and Software*, 25(3):359–386, 2010. [105](#), [141](#)
- [62] Serge Gratton and Luis Vicente. A merit function approach for direct search. *SIAM Journal on Optimization*, 24(4):1980–1998, 2014. [50](#), [51](#)
- [63] Anne Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997. [23](#), [24](#), [26](#)
- [64] Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear gauss–seidel method under convex constraints. *Operations research letters*, 26(3):127–136, 2000. [10](#), [27](#), [40](#), [66](#), [67](#), [79](#)
- [65] Wolfgang Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013. [102](#)
- [66] Raphael T Haftka. Optimization of flexible wing structures subject to strength and induced drag constraints. *Aiaa Journal*, 15(8):1101–1106, 1977. [7](#), [50](#)

- [67] Raphael T Haftka, Jaroslaw Sobieszczanski-Sobieski, and Sharon L Padula. On options for interdisciplinary analysis and design optimization. *Structural optimization*, 4:65–74, 1992. [18](#)
- [68] S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964. [103](#)
- [69] John Michael Hammersley, David Christopher Handscomb, JM Hammersley, and DC Handscomb. General principles of the monte carlo method. *Monte Carlo Methods*, pages 50–75, 1964. [101](#)
- [70] Zhong-Hua Han, Stefan Görtz, and Ralf Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and technology*, 25(1):177–189, 2013. [102](#)
- [71] Laurent Hascoet and Valérie Pascual. The tapenade automatic differentiation tool: principles, model, and specification. *ACM Transactions on Mathematical Software (TOMS)*, 39(3):1–43, 2013. [8](#)
- [72] Hisao Ishibuchi, Hiroyuki Masuda, and Yusuke Nojima. Pareto fronts of many-objective degenerate test problems. *IEEE Transactions on Evolutionary Computation*, 20(5):807–813, 2015. [121](#)
- [73] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3:233–260, 1988. [50](#)
- [74] Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21:345–383, 2001. [103](#), [104](#)
- [75] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455, 1998. [101](#), [102](#), [103](#)
- [76] Marc C Kennedy and Anthony O’Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000. [101](#)
- [77] Gaetan Kenway and Joaquim RRA Martins. Aerostructural shape optimization of wind turbine blades considering site-specific winds. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 6025, 2008. [7](#), [15](#)
- [78] Gaetan KW Kenway and Joaquim RRA Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, 2014. [140](#)
- [79] Hongman Kim, Scott Ragon, Grant Soremekun, Brett Malone, and Jaroslaw Sobieszczanski-Sobieski. Flexible approximation model approach for bi-level integrated system synthesis. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4545, 2004. [47](#)
- [80] Srinivas Kodiyalam and Jaroslaw Sobieszczanski-Sobieski. Bilevel integrated system synthesis with response surfaces. *AIAA journal*, 38(8):1479–1485, 2000. [47](#)

- [81] Norbert Kroll, Ralf Heinrich, W Krueger, and Björn Nagel. Fluid-structure coupling for aerodynamic analysis and design: a dlr perspective. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, page 561, 2008. 50
- [82] Ilan Kroo, Steve Altus, Robert Braun, Peter Gage, and Ian Sobieski. Multidisciplinary optimization methods for aircraft preliminary design. In *5th symposium on multidisciplinary analysis and optimization*, page 4325, 1994. 7, 15
- [83] Andrew B Lambe and Joaquim RRA Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46:273–284, 2012. 18
- [84] Thierry Lefebvre, Peter Schmollgruber, Christophe Blondeau, and Gérald Carrier. Aircraft conceptual design in a multi-level, multi-fidelity, multi-disciplinary optimization process. In *Proceedings of the 28th International Congress of The Aeronautical Sciences, Brisbane, Australia*, pages 23–28, 2012. 9, 137
- [85] Leifur Leifsson and Slawomir Koziel. Aerodynamic shape optimization by variable-fidelity computational fluid dynamics models: A review of recent progress. *Journal of Computational Science*, 10:45–54, 2015. 102
- [86] Xiang Li, Weiji Li, and Chang’an Liu. Geometric analysis of collaborative optimization. *Structural and Multidisciplinary Optimization*, 35:301–313, 2008. 43
- [87] Jacques Louis Lions and Pierre Lelong. Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. (*No Title*), 1968. 8
- [88] Jie Lu, Jialin Han, Yaoguang Hu, and Guangquan Zhang. Multilevel decision-making: A survey. *Information Sciences*, 346:463–487, 2016. 56
- [89] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984. 69
- [90] Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996. 52, 53
- [91] Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993. 67, 69
- [92] Zhoujie Lyu, Gaetan KW Kenway, and Joaquim RRA Martins. Aerodynamic shape optimization investigations of the common research model wing benchmark. *AIAA journal*, 53(4):968–985, 2015. 110, 124
- [93] Parviz M. Zadeh and Vassili Toropov. Multi-fidelity multidisciplinary design optimization based on collaborative optimization framework. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 5504, 2002. 10, 108, 110
- [94] Richard MacNeal. *The NASTRAN theoretical manual*, volume 221. Scientific and Technical Information Office, National Aeronautics and Space . . . , 1970. 7
- [95] Charles A Mader, Joaquim RRA Martins, Juan J Alonso, and Edwin Van Der Weide. Adjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA journal*, 46(4):863–873, 2008. 62

- [96] Andrew March and Karen Willcox. Multifidelity approaches for parallel multidisciplinary optimization. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 5688, 2012. [10](#), [104](#), [108](#), [110](#)
- [97] Andrew March and Karen Willcox. Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA journal*, 50(5):1079–1089, 2012. [106](#)
- [98] Andrew March, Karen Willcox, and Qiqi Wang. Gradient-based multifidelity optimisation for aircraft design using bayesian model calibration. *The Aeronautical Journal*, 115(1174):729–738, 2011. [101](#), [103](#), [106](#)
- [99] Andrew Irving March. *Multifidelity methods for multidisciplinary system design*. PhD thesis, Massachusetts Institute of Technology, 2012. [104](#)
- [100] DH802390 Martin. The essence of invexity. *Journal of optimization Theory and Applications*, 47:65–76, 1985. [33](#)
- [101] Joaquim Martins and Andrew Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 09 2013. [7](#), [8](#), [18](#), [19](#), [20](#), [21](#), [39](#), [41](#), [42](#), [47](#), [49](#), [51](#)
- [102] Joaquim RRA Martins. Aerodynamic design optimization: Challenges and perspectives. *Computers & Fluids*, 239:105391, 2022. [15](#)
- [103] Joaquim RRA Martins, Juan J Alonso, and James J Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6:33–62, 2005. [8](#), [15](#), [18](#), [31](#)
- [104] Joaquim RRA Martins and John T Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA journal*, 51(11):2582–2599, 2013. [8](#)
- [105] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000. [87](#), [132](#)
- [106] Ali Mehmani, Souma Chowdhury, and Achille Messac. Managing variable fidelity models in population-based optimization using adaptive model switching. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 2436, 2014. [108](#)
- [107] Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964. [103](#)
- [108] Romain Olivanti. *Multi-fidelity aerodynamic shape optimization considering flexible adjoint at multiple operating conditions*. PhD thesis, Toulouse, ISAE, 2021. [10](#), [11](#), [12](#), [22](#), [99](#), [100](#), [101](#), [105](#), [110](#), [120](#), [123](#), [124](#), [139](#), [141](#)
- [109] Romain Olivanti, François Gallard, Joël Brézillon, and Nicolas Gourdain. Comparison of generic multi-fidelity approaches for bound-constrained nonlinear optimization applied to adjoint-based cfd applications. In *AIAA Aviation 2019 Forum*, page 3102, 2019. [105](#), [106](#), [123](#), [124](#)

- [110] James M Ortega and Robert J Plemmons. Extensions of the ostrowski-reich theorem for sor iterations. *Linear algebra and its applications*, 28:177–191, 1979. [10](#), [23](#), [26](#), [27](#), [28](#), [40](#), [66](#), [67](#), [72](#)
- [111] James M Ortega and Werner C Rheinboldt. Monotone iterations for nonlinear equations with application to gauss-seidel methods. *SIAM Journal on Numerical Analysis*, 4(2):171–190, 1967. [10](#), [23](#), [26](#), [40](#), [45](#), [66](#), [67](#), [72](#)
- [112] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000. [17](#), [23](#)
- [113] James M Ortega and Maxine L Rockoff. Nonlinear difference equations and gauss-seidel type iterative methods. *SIAM Journal on Numerical Analysis*, 3(3):497–513, 1966. [10](#), [23](#), [26](#), [40](#), [45](#), [66](#), [67](#), [72](#), [113](#)
- [114] Jiří V Outrata. A note on the usage of nondifferentiable exact penalties in some special optimization problems. *Kybernetika*, 24(4):251–258, 1988. [55](#)
- [115] Ricardo M Paiva, André RD Carvalho, Curran Crawford, and Afzal Suleman. Comparison of surrogate models in a multidisciplinary optimization framework for wing design. *AIAA journal*, 48(5):995–1006, 2010. [9](#), [97](#), [107](#)
- [116] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *arXiv:1806.10761 [cs, math, stat]*, June 2018. arXiv: 1806.10761. [9](#), [97](#), [100](#), [101](#), [102](#), [103](#), [112](#), [124](#)
- [117] P Piperni, A DeBlois, and R Henderson. Development of a multilevel multidisciplinary-optimization capability for an industrial environment. *AIAA journal*, 51(10):2335–2352, 2013. [102](#)
- [118] Olivier Pironneau. On optimum profiles in stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, 1973. [50](#)
- [119] Michael JD Powell. On search directions for minimization algorithms. *Mathematical programming*, 4(1):193–201, 1973. [67](#), [88](#), [159](#)
- [120] Michael JD Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994. [64](#), [88](#), [159](#)
- [121] Michael JD Powell. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007. [88](#), [134](#), [159](#)
- [122] Lisa Pretsch, Ilya Arsenyev, Catharina Czech, and Fabian Duddeck. Interdisciplinary design optimization of compressor blades combining low-and high-fidelity models. *Structural and Multidisciplinary Optimization*, 66(4):70, 2023. [10](#), [108](#), [109](#), [110](#)
- [123] Pieter-Jan Proesmans and Roelof Vos. Comparison of future aviation fuels to minimize the climate impact of commercial aircraft. In *AIAA Aviation 2022 Forum*, page 3288, 2022. [15](#)
- [124] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005. [104](#)

- [125] Kaushik Radhakrishnan, KT Deck, P Proesmans, Florian Linke, Feijia Yin, Volker Grewe, Roelof Vos, Benjamin Lührs, Malte Niklaß, and IC Dedoussi. Minimizing the climate impact of the next generation aircraft using novel climate functions for aircraft design. In *33rd Congress of the International Council of the Aeronautical Sciences, ICAS 2022*, page ICAS2022_0602. International Council of the Aeronautical Science (ICAS), 2022. 15
- [126] Dev Gorur Rajnarayan. *Trading risk and performance for engineering design optimization using multifidelity analyses*. Stanford University, 2009. 104
- [127] Isabelle Ramière and Thomas Helfer. Iterative residual-based vector methods to accelerate fixed point iterations. *Computers & Mathematics with Applications*, 70(9):2210–2226, 2015. 22, 40
- [128] James Reuther, Juan Alonso, Joaquim RRA Martins, and Stephen Smith. A coupled aero-structural optimization method for complete aircraft configurations. In *37th Aerospace sciences meeting and exhibit*, page 187, 1999. 50
- [129] TD Robinson, MS Eldred, KE Willcox, and R Haimes. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *Aiaa Journal*, 46(11):2814–2822, 2008. 105
- [130] Brian D Roth and Ilan M Kroo. Enhanced collaborative optimization: a decomposition-based method for multidisciplinary design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 43253, pages 927–936, 2008. 43
- [131] Itham Salah El Din, Antoine Dumont, and Christophe Blondeau. Transonic wing-body civil transport aircraft aero-structural design optimization using a bi-level high fidelity approach—a focus on the aerodynamic process. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 144, 2013. 9, 137
- [132] Lucien A Schmit. Structural design by systematic synthesis. In *Proceedings of the Second National Conference on Electronic Computation, ASCE, Sept., 1960*, 1960. 7, 50
- [133] Gerd Schuhmacher, Fernað Daoud, Ogmundur Petersson, Markus Wagner, R Straße, G Schuhmacher, and O Petersson. Multidisciplinary airframe design optimization. In *28th international congress of the aeronautical sciences*, volume 1, pages 44–56, 2012. 7
- [134] Dieter Schwamborn, Thomas Gerhold, and Ralf Heinrich. The dlr tau-code: recent applications in research and industry. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods . . . , 2006. 7
- [135] R Sellar, S Batill, and J Renaud. Response surface based, concurrent subspace optimization for multidisciplinary system design. In *34th aerospace sciences meeting and exhibit*, page 714, 1996. 9, 10, 11, 41, 46, 86, 92, 97, 100, 106, 107, 125, 156, 157

- [136] Timothy W Simpson, Timothy M Mauery, John J Korte, and Farrokh Mistree. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA journal*, 39(12):2233–2241, 2001. [9](#), [97](#), [107](#)
- [137] Ian P Sobieski and Ilan M Kroo. Collaborative optimization using response surface estimation. *AIAA journal*, 38(10):1931–1938, 2000. [9](#), [43](#), [97](#), [106](#)
- [138] Jaroslaw Sobieszczanski-Sobieski. Optimization by decomposition: a step from hierarchic to non-hierarchic systems. In *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, Hampton, VA, NASA CP, volume 3031, pages 51–78, 1988. [46](#)
- [139] Jaroslaw Sobieszczanski-Sobieski, Jeremy Agte, and Jr. Sandusky, Robert. Bi-level integrated system synthesis (BLISS). In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Multidisciplinary Analysis Optimization Conferences. American Institute of Aeronautics and Astronautics, September 1998. [8](#), [9](#), [11](#), [18](#), [39](#), [47](#), [86](#), [100](#), [131](#), [133](#), [137](#), [140](#), [154](#)
- [140] Jaroslaw Sobieszczanski-Sobieski, Jeremy S Agte, and Robert R Sandusky Jr. Bilevel integrated system synthesis. *AIAA journal*, 38(1):164–172, 2000. [9](#), [39](#), [43](#), [97](#), [106](#)
- [141] Jaroslaw Sobieszczanski-Sobieski and Raphael T Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization*, 14:1–23, 1997. [7](#)
- [142] Heinrich von Stackelberg. Marktform und gleichgewicht. (*No Title*), 1934. [51](#)
- [143] Georg Still. Lectures on parametric optimization: An introduction. *Optimization Online*, page 2, 2018. [61](#)
- [144] Knut Sydsæter and Peter J Hammond. *Essential mathematics for economic analysis*. Pearson Education, 2008. [35](#)
- [145] Nathan P Tedford and Joaquim RRA Martins. Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11:159–183, 2010. [46](#), [104](#)
- [146] S. Tosserams, L. F. P. Etman, and J. E. Rooda. An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Structural and Multidisciplinary Optimization*, 34(3):211–227, July 2007. [8](#), [44](#), [45](#), [57](#), [69](#), [140](#)
- [147] Simon Tosserams, LF Pascal Etman, and JE Rooda. A classification of methods for distributed system optimization based on formulation structure. *Structural and Multidisciplinary Optimization*, 39:503–517, 2009. [8](#), [18](#), [41](#), [42](#), [47](#), [51](#)
- [148] Simon Tosserams, LFP Etman, and JE Rooda. Augmented lagrangian coordination for distributed optimal design in mdo. *International journal for numerical methods in engineering*, 73(13):1885–1910, 2008. [8](#), [44](#), [45](#), [140](#)
- [149] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, March 2009. [67](#)
- [150] Charlie Vanaret, François Gallard, and Joaquim RRA Martins. On the consequences of the” no free lunch” theorem for optimization on the choice of an appropriate mdo

- architecture. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 3148, 2017. [18](#), [39](#), [42](#), [45](#)
- [151] Richard S Varga. Iterative analysis. *New Jersey*, 322, 1962. [24](#)
- [152] David Venditti and David Darmofal. Grid adaptation for functional outputs of 2-d compressible flow simulations. In *Fluids 2000 Conference and Exhibit*, page 2244, 2000. [119](#)
- [153] Luis Vicente, Gilles Savard, and Joaquim Júdice. Descent approaches for quadratic bilevel programming. *Journal of Optimization theory and applications*, 81(2):379–399, 1994. [52](#), [57](#)
- [154] Luis N Vicente and Paul H Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, 5(3):291–306, 1994. [56](#)
- [155] Xiaobang Wang, Mao Li, Yuanzhi Liu, Wei Sun, Xueguan Song, and Jie Zhang. Surrogate based multidisciplinary design optimization of lithium-ion battery thermal management system in electric vehicles. *Structural and Multidisciplinary Optimization*, 56:1555–1570, 2017. [108](#)
- [156] Xiaobang Wang, Yuanzhi Liu, Wei Sun, Xueguan Song, and Jie Zhang. Multidisciplinary and multifidelity design optimization of electric vehicle battery thermal management system. *Journal of Mechanical Design*, 140(9):094501, 2018. [10](#), [97](#), [108](#), [109](#), [110](#), [141](#)
- [157] Neil Wu, Gaetan Kenway, Charles A Mader, John Jasa, and Joaquim RRA Martins. pyoptspase: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020. [88](#)
- [158] Neil Wu, Charles A Mader, and Joaquim RRA Martins. A gradient-based sequential multifidelity approach to multidisciplinary design optimization. *Structural and Multidisciplinary Optimization*, 65(4):131, 2022. [9](#), [10](#), [98](#), [99](#), [109](#), [110](#), [112](#), [114](#), [121](#), [124](#)
- [159] Yangyang Xu and Wotao Yin. A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update. *Journal of Scientific Computing*, 72(2):700–734, August 2017. [69](#)
- [160] Lili Yang and Deyu Wang. Modified collaborative optimization for feasibility problem of final solution. *Structural and Multidisciplinary Optimization*, 56(5):1109–1123, 2017. [43](#)
- [161] Raul Yondo, Esther Andrés, and Eusebio Valero. A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Progress in aerospace sciences*, 96:23–61, 2018. [97](#), [107](#)
- [162] Alain B Zemkoho. *Bilevel programming: reformulations, regularity, and stationarity*. PhD thesis, Dissertation, Freiberg, TU Bergakademie Freiberg, 2012, 2012. [56](#)

Appendix A

The Sobieski Super Business Jet (SSBJ) test case

The Sobieski Super Business Jet (SSBJ) test case, illustrated by Figure A.1, was first introduced in [139] to present the Bi-Level Integrated System Synthesis (BLISS) architecture (Figure 3.1). It is a well-known conceptual design code for the MDO community, conceived to test the different architectures at an affordable cost. As it has been originally designed for the BLISS architecture, it is well suited to test more general bi-level architecture such as the ones considered in this paper.

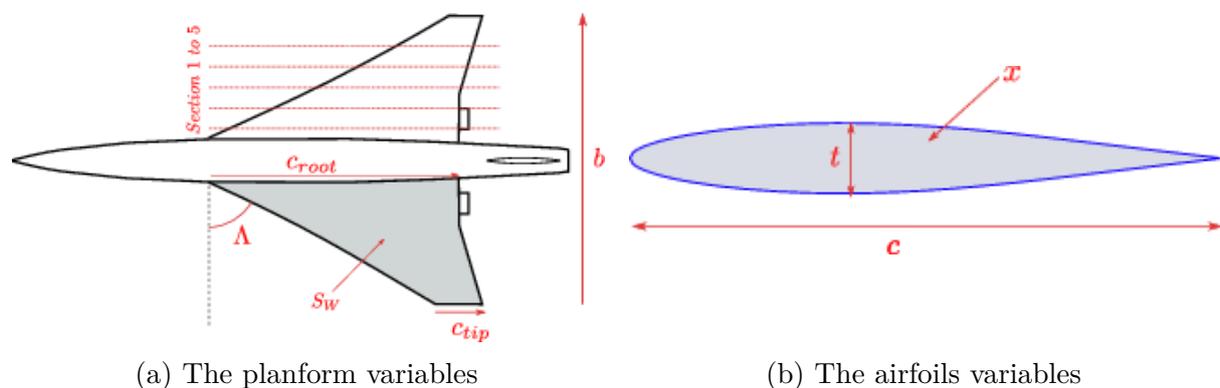


Figure A.1: Graphical representation of the SSBJ test case

The SSBJ model is composed of three strongly coupled disciplines: Structures (discipline 1), Aerodynamics (discipline 2), and Propulsion (discipline 3). This MDO problem aims to maximize the range of the business jet which is computed by a fourth, non-coupled, discipline (Mission). Considering the dimensions of the problem, there is 6 shared variables, 4 local variables, 10 coupling variables and 12 inequality constraints. The MDF XDSM diagram on the SSBJ test case is given in Figure A.2.

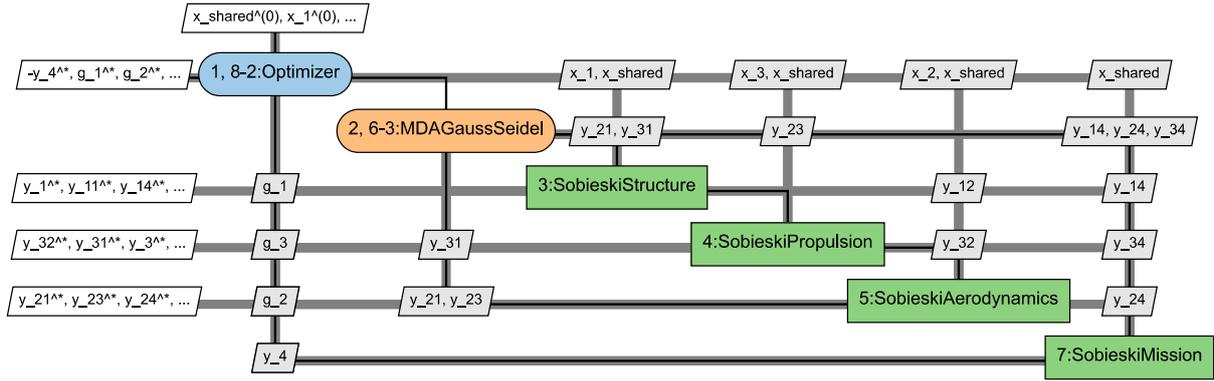


Figure A.2: MDF XDSM diagram on SSBJ test case

The description of all design variables, with their initial and optimal values for the MDF architecture are given in Table A.1. At the optimal design point, there is 2 active constraints, g_2 (computed by Aerodynamics), which represents the constraint about the pressure gradient, and $g_3[2]$ (computed by Propulsion), the throttle constraint.

Variable	Block	Initial	Optimum
Range (nm)	None	$f^{(0)} = 535.79$	$f^* = 3963.88$
λ	$x_{1,1}$	0.25	0.38757
x	$x_{1,2}$	1	0.75
C_f	x_2	1	0.75
Th	x_3	0.5	0.15624
t/c	$x_{0,1}$	0.05	0.06
$h(ft)$	$x_{0,2}$	45000	60000
M	$x_{0,3}$	1.6	1.4
AR	$x_{0,4}$	5.5	2.5
$\Lambda(deg)$	$x_{0,5}$	55	70
$S_w(ft^2)$	$x_{0,6}$	1000	1500

Table A.1: MDF design variables for SSBJ test case

Appendix B

The Sellar problem and analytical expressions for the two-block variant

Here we present the equations for the original Sellar optimization problem [135] from which several variants, resulting from slight modifications of the problem, are created for the sake of the numerical experiments in the manuscript.

One of these variants is a two-block modification of the original Sellar problem to allow block decomposition of the lower optimization problem. Considering points around the MDF solution to avoid discontinuity and non-differentiability, the analytical expression of the coupling variables at equilibrium is given. In addition, the first and second derivatives of all functions and coupling expressions are given.

B.1 The original Sellar problem

The original Sellar optimization problem [135] is a simple MDO problem with two strongly coupled disciplines. The original Sellar problem is given in the following form:

$$\begin{aligned} & \text{minimize } f(x_{0,1}, x_{0,2}, x_1, y_1, y_2) = x_1^2 + x_{0,2} + y_1^2 + e^{-y_2} \\ & \text{with respect to } x_{0,1}, x_{0,2}, x_1, y_1, y_2 \\ & \text{subject to :} \\ & -10 \leq x_{0,1} \leq 10 \\ & 0 \leq x_{0,2} \leq 10 \\ & 0 \leq x_1 \leq 10 \\ & c_1(y_1) = 3.16 - y_1^2 \leq 0 \\ & c_2(y_2) = y_2 - 24 \leq 0 \\ & y_1 - y_1(x_{0,1}, x_{0,2}, x_1, y_2) = 0 \\ & y_2 - y_2(x_{0,1}, x_{0,2}, y_1) = 0 \end{aligned} \tag{B.1.1}$$

$$\text{Discipline 1: } y_1(x_{0,1}, x_{0,2}, x_1, y_2) = \sqrt{x_{0,1}^2 + x_1 + x_{0,2} - 0.2y_2} \quad (\text{B.1.2})$$

$$\text{Discipline 2: } y_2(x_{0,1}, x_{0,2}, y_1) = |y_1| + x_{0,1} + x_{0,2} \quad (\text{B.1.3})$$

B.2 Analytical expressions for the two-block Sellar

The manuscript contains a two-blocks modification of the already existing Sellar optimization problem [135]. This modified problem, described by (3.6.2) within Section 3.6.2, is mainly used to benchmark the bi-level architectures presented in Chapter 3 by allowing the block decomposition of the lower optimization problem.

Here, the computation of the coupling variables at equilibrium, i.e. after the execution of an MDA, is reunited for all possible design vectors that are sufficiently closed to the MDF solution to avoid discontinuities and complex values. First and second derivatives for the couplings, the objective, and the constraints are also provided.

B.2.1 Couplings expressions

As we are interested in studying the problem in a vicinity of the unique MDF solution of the original sellar problem, cases where y_1 and y_2 take negative and/or complex values are not considered.

Considering a design vector $(x_{0,1}, x_{0,2}, x_1, x_2)$ near the unique MDF solution of Equation (3.6.2), the design is feasible if and only if:

$$\begin{cases} y_1^2 = x_{0,1}^2 + x_1 + x_{0,2} - 0.2y_2 \\ y_2 = y_1 + x_{0,1} + x_{0,2} - x_2 \\ (y_1, y_2) \in (\mathbb{R}^+)^2 \end{cases} \iff \begin{cases} y_1^2 + 0, 2y_1 + C = 0 \\ y_2 = y_1 + x_{0,1} + x_{0,2} - x_2 \\ (y_1, y_2) \in (\mathbb{R}^+)^2 \end{cases} \quad (\text{B.2.1})$$

Let $\Delta = 0,04 - 4C$ with $C = -x_1 - 0.2x_2 + x_{0,1}(0.2 - x_{0,1}) - 0.8x_{0,2}$

The direct formula which explicitly compute the coupling variable at equilibrium, with relates to the design variables, is given by:

$$\begin{cases} y_1(x_{0,1}, x_{0,2}, x_1, x_2) = \frac{-0.2 + \sqrt{\Delta}}{2} \\ y_2(x_{0,1}, x_{0,2}, x_1, x_2) = \frac{-0.2 + \sqrt{\Delta}}{2} + x_{0,1} + x_{0,2} - x_2 \end{cases} \quad (\text{B.2.2})$$

B.2.2 Couplings's first and second derivatives

$$\nabla y_1 = \frac{1}{4\sqrt{\Delta}}(\nabla\Delta) = \frac{1}{4\sqrt{\Delta}}(-4\nabla C) = \frac{-1}{\sqrt{\Delta}}\nabla C \quad (\text{B.2.3})$$

$$\nabla y_2 = \frac{-1}{\sqrt{\Delta}}\nabla C + \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \nabla y_1 + \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix} \quad (\text{B.2.4})$$

$$\nabla^2 y_1 = \nabla^2 y_2 = \frac{-1}{\sqrt{\Delta}} \left[\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \frac{2}{\Delta} \nabla C \nabla C^T \right] \quad (\text{B.2.5})$$

B.2.3 Objective and constraints's gradient

$$\nabla f = \begin{bmatrix} 2x_1 \\ 2x_2 \\ 0 \\ 1 \end{bmatrix} + 2y_1 \nabla y_1 - e^{-y_2} \nabla y_2 \quad (\text{B.2.6})$$

$$\nabla c_1 = -2y_1 \nabla y_1 \quad (\text{B.2.7})$$

$$\nabla c_2 = \nabla y_2 \quad (\text{B.2.8})$$

B.2.4 Objective and constraint's Hessian

$$\nabla^2 f = \begin{bmatrix} 2 * I_2 & 0_{2,2} \\ 0_{2,2} & 0_{2,2} \end{bmatrix} + 2[(\nabla y_1 \nabla y_1^T) + y_1 \nabla^2 y_1] + e^{-y_2} [\nabla y_2 \nabla y_2^T - \nabla^2 y_2] \quad (\text{B.2.9})$$

$$\nabla^2 c_1 = -2[(\nabla y_1 \nabla y_1^T) + y_1 \nabla^2 y_1] \quad (\text{B.2.10})$$

$$\nabla^2 c_2 = \nabla^2 y_2 \quad (\text{B.2.11})$$

Appendix C

Notes on COBYLA

This section briefly describes the main principle of Powell's COBYLA algorithm [119, 120, 121]. This is the derivative free optimization algorithm that is considered in this manuscript to handle the system level of the presented bi-level architectures.

C.1 Algorithm

Let the following nonlinear optimization problem :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & c_i(x) \leq 0 \quad i = 1, 2, \dots, m \end{aligned} \tag{C.1.1}$$

Given $n + 1$ vertices $\{x^{(j)} : j = 0, 1, \dots, n\}$ of a non-degenerate simplex in \mathbb{R}^n , there are unique linear functions \hat{f} and $\{\hat{c}_i : i = 0, 1, \dots, m\}$ that interpolate f and $\{c_i : i = 0, 1, \dots, m\}$. Considering the following linear programming problem :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \hat{f}(x) \\ \text{subject to} \quad & \hat{c}_i(x) \leq 0 \quad i = 1, 2, \dots, m \end{aligned} \tag{C.1.2}$$

COBYLA defines a merit function $\hat{\Phi}(x) = \hat{f}(x) + \mu[\max\{0\} \cup \{\hat{c}_i(x) : i = 0, 1, \dots, m\}]$ and order all the vertices such that $\hat{\Phi}(x^{(0)}) < \hat{\Phi}(x^{(1)}) < \dots < \hat{\Phi}(x^{(n)})$.

At each iterate, COBYLA finds the best point $x^{(*)}$ such that $\|x^{(*)} - x^{(0)}\| \leq \rho$. With ρ being the trust region radius.

Either $x^{(*)}$ is considered good enough and replace $x^{(n)}$ as a vertex, or we decrease the trust region radius ρ and/or increase the penalization parameter μ .

C.2 Convergence properties

No global convergence proof is given for COBYLA, however the 4 assumptions ensure termination of the algorithm:

1. μ the penalization parameter is bounded.

2. ρ the trust region radius decrease a finite number of time.
3. As μ and ρ remain constant, $x^{(0)}$ (the best point yet) must changed in a finite number of iteration.
4. As μ and ρ remain constant, the number of replacement of $x^{(0)}$ is finite.

As the algorithm ensure that (2) and (3) are ensured through the updating rules, it is however complex to show that (1) and (4) hold in the scope of COBYLA and/or may need too much restrictions to be considered.

Titre : Architectures bi-niveau et algorithmes multi-fidélité pour l'optimisation multidisciplinaire en haute dimension.

Mots clés : Optimisation, MDO, Bi-niveau, Multifidélité, Adjoint

Résumé : L'optimisation multidisciplinaire est un domaine de l'ingénierie impliquant l'intégration de méthodes numériques, d'outils d'ingénierie et de modèles mathématiques afin de résoudre des problèmes d'optimisation impliquant plusieurs champs disciplinaires. Les approches MDO visent à améliorer la qualité de la solution, la robustesse et si possible réduire les coûts de calculs du processus global en prenant en compte les interdépendances et l'aspect organisationnel des différents modèles impliqués. Si les méthodes MDO ont montré leur intérêt dans plusieurs domaines d'ingénierie, tout d'abord en aéronautique puis dans d'autres domaines tels que l'automobile ou le dimensionnement d'éoliennes, plusieurs verrous restent à lever pour profiter pleinement des capacités MDO en haute dimension. Parmi ces verrous, on retrouve la formalisation mathématique d'architectures MDO, c'est-à-dire une reformulation du problème d'optimisation et une suite de calcul associée. En particulier pour les architectures distribuées, dans lesquels plusieurs sous-problèmes d'optimisation sont définis, qui sont réputées pour avoir de moins bonnes propriétés de convergence. Un autre verrou important est l'intégration efficace de modèles de substitution, moins coûteux, pour réduire les temps de calcul globaux.

La première partie de la thèse porte sur l'élaboration d'une nouvelle architecture bi-niveau, dans lequel deux problèmes d'optimisation sont imbriqués. Dans celle-ci, les variables partagées, entrées pour toutes les disciplines et peu nombreuses, sont optimisées avec un algorithme sans gradient, tandis que les variables locales, entrées d'une unique discipline et plus nombreuses, sont laissées à un sous-problème d'optimisation imbriqué dans le premier. L'originalité de cette architecture étant de résoudre le sous-problème par blocs, en utilisant un algorithme de minimisations alternées, le block-coordinate-descent (ou BCD). Sous certaines hypothèses de bonne définition et séparabilité des blocs, un résultat de convergence locale de l'algorithme est obtenu, améliorant les propriétés de régularité des fonctions et la convergence. Plusieurs applications sur des cas test classiques de MDO illustrent ces propriétés, ainsi que la meilleure scalabilité en fonction du nombre de couplages, contraintes et variables locales des approches bi-niveau par rapport aux architectures monolithiques quand les dérivées couplées de l'ensemble du système ne sont pas disponibles, tel que dans certains contextes industriels.

La seconde partie du manuscrit se porte sur l'application des méthodes multi-fidélités dans des processus MDO. Ces méthodes exploitent à la fois un modèle haute-fidélité, de grande précision mais au coût élevé, et un ou plusieurs modèles approchés, de basses fidélités, moins coûteux mais à la précision amoindrie. Pour le bon fonctionnement de ces méthodes il est souvent nécessaire de choisir des niveaux de basse-fidélité adéquats, c'est-à-dire des modèles possédant un rapport coût/précision intéressant. Ceci est d'autant plus critique dans le cadre de la MDO étant donné les nombreux moyens de générer ces modèles de basse-fidélité dont l'impact de leur introduction dans l'optimisation n'est pas connu a priori. Les contributions de ce chapitre portent sur la classification d'un grand nombre de modèles de basse-fidélité, notamment par l'introduction de deux critères d'estimation d'erreur ne nécessitant pas de statistiques coûteuses. Le parti pris étant que, étant donné le très grand nombre de modèles à classer, une approximation grossière est suffisante pour espérer un gain en temps de la multi-fidélité. De petits cas test de MDO sont considérés pour illustrer ces critères en exploitant les spécificités de la MDO pour rapidement générer des modèles de basse-fidélité. Les modèles de fidélité sont ensuite choisis selon les critères proposés et une optimisation multi-fidélité, basée sur une approche de raffinement, démontre l'impact de ces choix sur son efficacité.

Titre: Bi-level architectures and multi-fidelity algorithms for multidisciplinary optimization in high dimensions.

Key words: Optimization, MDO, Bi-level, Multi-fidelity, Adjoint

Abstract: Multidisciplinary Optimization is a field of engineering that integrates numerical methods, engineering tools, and mathematical models to solve optimization problems involving multiple disciplines. MDO approaches aim to improve the quality and robustness of the solution and, if possible, reduce the computational cost of the overall process by taking into account the interdependencies and organizational aspects of the different models involved. While MDO methods have been proven in a number of engineering fields, starting with aeronautics and then moving to other areas such as automotive and wind turbine design, there are still a number of hurdles to overcome before we can take full advantage of the high-dimensional MDO capabilities. These include the mathematical formalization of MDO architectures, i.e. a reformulation of the optimization problem and an associated computational sequence. This is especially true for distributed architectures, where multiple optimization sub-problems are defined, which are known to have worse convergence properties. Another important issue is the efficient integration of less costly substitution models to reduce the overall computation time.

The first part of the thesis deals with the development of a new bi-level architecture in which two optimization problems are nested. In this architecture, the shared variables, inputs for all disciplines and few in number, are optimized with a gradient-free algorithm, while the local variables, inputs for a single discipline and more numerous, are left to an optimization sub-problem nested within the first. The originality of this architecture lies in the fact that the sub-problem is solved in blocks, using an alternating minimizations algorithm, the block-coordinate-descent (BCD) algorithm. Under certain assumptions of good block definition and separability, a local convergence result of the algorithm is obtained, which improves the regularity properties of the functions and the convergence. Several applications to classical MDO test cases illustrate these properties, as well as the better scalability as a function of the number of couplings, constraints, and local variables of bi-level approaches compared to monolithic architectures when coupled derivatives of the whole system are not available, as in some industrial contexts.

The second part of the manuscript focuses on the use of multi-fidelity methods in MDO processes. These methods make use of both a high-fidelity model, with high accuracy but high cost, and one or more low-fidelity approximate models, with lower cost but lower accuracy. For these methods to work properly, it is often necessary to choose appropriate low-fidelity levels, i.e., models with an attractive cost/accuracy ratio. This is all the more critical in the context of MDO, since there are many ways to generate these low-fidelity models, the impact of which on optimization is not known a priori. The contributions in this chapter focus on the classification of a large number of low-fidelity models, in particular through the introduction of two error estimation criteria that do not require costly statistics. The assumption is that given the very large number of models to be classified, a rough approximation is sufficient to hope for a gain in time from the multifidelity. Small MDO test cases are considered to illustrate these criteria, exploiting the specificities of MDO to quickly generate low-fidelity models. Fidelity models are then chosen according to the proposed criteria, and a multi-fidelity optimization based on a refinement approach demonstrates the impact of these choices on its efficiency.