

Cranfield University

Jean DEMANGE

Multifidelity Multiobjective Trust-Region-Based Optimisation For High-Lift Devices

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

PhD thesis

Academic years 2014–2018

Supervised by: **Professor Mark Savill**
Doctor Timoleon Kipouros

Submitted February 2018

Cranfield University

Jean DEMANGE

Multifidelity Multiobjective Trust-Region-Based Optimisation For High-Lift Devices

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

PhD thesis

Academic years 2014–2018

Supervised by: **Professor Mark Savill**
Doctor Timoleon Kipouros

Submitted February 2018

© Cranfield University 2018. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

To my grandparents. . .

This is one corner. . . of one country, in one continent, on one planet that's a corner of a galaxy that's a corner of a universe that is forever growing and shrinking and creating and destroying and never remaining the same for a single millisecond. And there is so much, so much to see.

– The doctor.

Abstract

This thesis addresses the maximum lift prediction early in the design stage. To do so, the wing high-lift element positions is optimised to estimate the maximum achievable lift. Aircraft design is currently done in at least two main stages: a first high-level design work is undertaken using fast but not very reliable tools which is then improved in a subsequent loop using more accurate tools but requiring more computational time. The present work uses synergies between those models to quickly estimate the maximum lift capabilities of an aerofoil with the use of a derivative-free method along with a trust region framework to perform the multiobjective optimisation while ensuring the convergence towards the most accurate tool optima.

The method is first applied on a single-objective formulation and shows significant time saving when the aerodynamic problem is simple. A decrease in benefits is observed when applied to the high-lift devices optimisation because of the increased differences between the low- and the high-fidelity models and a decrease in the tools robustness. A co-Kriging model instead of the additive correction is shown to be beneficial for the accuracy without reducing time saving.

Two trust region definitions are compared and shown not to be equivalent: the Euclidean trust region is more accurate but usually more expensive whereas the step-based one uses a more approximated solution of the subproblem which decreases the cost. It is also shown, in accordance with literature, that the multifidelity method increases the variability in convergence because it exacerbates the suboptimiser stochastic characteristics.

The main contribution to knowledge is the extension to multiobjective problems. Because of the low correlation between the low- and high-fidelity tools, the method with an additive correction is shown to be dominated by the high-fidelity-only optimiser, albeit the multifidelity is more diversified. The use of a co-Kriging model shows a significant improvement of the Pareto front optimality and extent. Single-fidelity Surrogate Based Optimisation however may provide similar benefits.

A novel visualisation approach to compare two models of different fidelity is introduced and a qualitative analysis of the low-fidelity accuracy effect on the multifidelity convergence is done: the most sensitive variables should be correctly captured by the lower fidelity model whereas the less sensitive ones will be corrected by the model correction.

Acknowledgement

First and foremost, I would like to express my gratitude to my Professor, Prof. Savill: thank you for providing me with the opportunity of researching in your group, thank you for your understanding and support throughout the 3 years, and thank you for your reassuring and concise guidance. I ought to greatly thank Dr. Timos Kipouros who provided the essential technical hints allowing this thesis to hatch.

I vigorously acknowledge Airbus Ltd. that has made possible this work and provided all the required resources. In particular, I wish to thank Murray Cross and Philippe Garang for their support and management; I would like to express my gratitude to Joël Brezillon who despite the distance provided the key elements for a successful work on aerodynamic optimisation. Gratitude as well to Stefano Tursi for extending my knowledge in Python. I acknowledge the help from my former Airbus colleagues and I want to thank the people who challenged my work which contributed at making it sounder, particularly Dr. Sanjiv Sharma.

The project was co-funded by Innovate UK and its support is gratefully acknowledged.

A PhD comes from inspiration and as a token of gratitude, I want to thank all my friends and colleagues encountered before the start of my PhD who inspired me at pursuing the dream: thank you Dr. Guiseppe Trapani, Dr. Christian Agostinelli, Dr. Giovana Ferraro and Dr. Mahbubul Alam. In particular I also wish to thank Dr. Feng Zhu who provided such inspiration but also guidance and help during my PhD; thank you as well for our long discussions and my time in China!

Despite being in a different country, those years would not have been possible without my entire family. I would like to offer my special thanks to my parents who provided the right environment and support. My siblings as well are thanked: Paul, and his family, Gilles, Luc and Marie. I am also grateful to my elders: my grandfathers who inspired me as well to pursue my work, and my grandmothers.

I want to thank also all my research group friends and colleagues: thank you Joe Loxham for my paper proof reading, thank you Kamal Haider for enduring my special way of teaching, thank you Dr. Davide Di Pasquale for our work together, thank you Dr. Jesus Nieto for your unique smile and enthusiasm and last but not least, thank Gillian Hargreaves for your patience concerning my late replies.

Last but not least, I express my sincere gratitude towards Dr. Gabriele Luigi Mura

Acknowledgement

who relentlessly challenged my work and provided guidance and an attentive thesis proofreading, but also towards Spyros Kontogiannis and Alexandros Georgiades for sharing the Bristolian beers and music and towards Simone Simeone and Dr. Andrea Castrichini for our regular and unmissable coffee breaks.

Contents

Abstract	i
Acknowledgement	iii
List of Figures	xi
List of Tables	xvii
Nomenclature	xix
1 Introduction	1
1.1 Context	1
1.2 Aim and Objectives	2
1.3 Contribution to Knowledge	3
1.4 Overview	3
2 High-Lift Design	7
2.1 Industrial Design Process	7
2.2 Design Objectives for High-Lift Devices	8
2.2.1 Take-Off	9
2.2.2 Landing	9
2.3 Existing Systems	12
2.3.1 Leading Edge Devices	12
2.3.2 Trailing Edge Devices	13
2.3.3 Kinematics as Constraints	14
2.3.4 High-Lift Devices Parametrisation	15
2.4 High-Lift Devices Description Through Physics	18
2.5 From Design to Optimisation	20
2.5.1 Optimisation Design Process	20
2.5.2 Aerodynamic Specificities	21
3 Optimisation Applied to Computational Fluid Dynamics	23
3.1 Computational Fluid Dynamics	23
3.1.1 Navier-Stokes Equations	24
3.1.2 Reynolds Averaged Navier Stokes	25

3.1.3	Euler	30
3.1.4	Potential Flow and Panel Method	30
3.1.5	Boundary Layer Equations	31
3.1.6	Coupling Inviscid and BLE	32
3.1.7	ESDU Datasheets	35
3.2	Optimisation	35
3.2.1	Generalities	35
3.2.2	MultiObjective	36
3.2.3	Local Methods	37
3.2.4	Metaheuristic Methods	40
3.2.5	MACROS	43
3.3	Surrogate Modelling	46
3.3.1	Sampling Methods	46
3.3.2	Parametric Models	47
3.3.3	Radial Basis Functions	48
3.3.4	Kriging	49
3.3.5	Co-Kriging	51
3.3.6	Curse of Dimensionality	54
3.4	Implementation	56
4	Multifidelity Literature Review	57
4.1	Introduction	57
4.2	Fidelity Types	58
4.2.1	Physics	58
4.2.2	Numerical Accuracy	60
4.2.3	Surrogate	60
4.2.4	Experiments	61
4.3	Fusing Fidelities	62
4.3.1	Hierarchical	62
4.3.2	Correction	62
4.3.3	Space Mapping	64
4.3.4	Surrogate	64
4.4	Model Management Strategy	65
4.4.1	Filtering	65
4.4.2	Surrogate Based Optimisation	66

4.4.3	Trust Region	67
4.5	Multifidelity Multiobjective	68
4.6	Multifidelity Applied to High-Lift Devices Optimisation	70
4.7	Method Choice	71
5	Multifidelity Optimisation Methodology: Application to Single-Objective Optimisation	73
5.1	Method	73
5.1.1	Surrogate Model	75
5.1.2	Subproblem Optimiser	76
5.1.3	Ratio of Improvement Computation	77
5.1.4	Trust Region Management	77
5.1.5	New Point Selection	79
5.2	Implementation	79
5.3	Mathematical Test Functions	80
5.3.1	One-Dimensional Example	80
5.3.2	Two-Dimensional Example	80
5.4	Single Element Aerofoil Test Case	84
5.4.1	Problem Formulation	84
5.4.2	Results	86
5.5	Multielement Aerofoil Academic Test Case	88
5.5.1	Geometry, Parameterisation and Objective	90
5.5.2	High-Fidelity CFD Solver	91
5.5.3	Low-Fidelity CFD Solver	92
5.5.4	Very Low-Fidelity CFD Solver	93
5.5.5	Suboptimisation	95
5.5.6	Results	95
5.6	Multielement Aerofoil Industrial Test Case	97
5.6.1	Geometry, Parameterisation and Objective	97
5.6.2	High-Fidelity CFD Solver	101
5.6.3	Low-Fidelity CFD Solver	102
5.6.4	Suboptimisation	102
5.6.5	Initial 2-Variables Optimisation	103
5.6.6	Results: Small Boundaries	106
5.6.7	Results: Extended Boundaries	108

5.7	Conclusion	109
6	Multifidelity Multiobjective Optimisation Methodology	113
6.1	Method Extension	113
6.1.1	Main Process	113
6.1.2	Ratio of Improvement Computation	114
6.1.3	Trust Region Management	117
6.1.4	New Point Selection	118
6.1.5	Hypervolume Indicator	119
6.1.6	Implementation	120
6.2	Academic Test Case	121
6.2.1	Suboptimisation	122
6.2.2	Convergence and Final Pareto Front	123
6.2.3	Low-Fidelity Effect	130
6.2.4	Three Configurations Analysis	130
6.3	Industrial Test Case	134
6.3.1	Suboptimisation	134
6.3.2	Convergence and Final Pareto Front	134
6.4	Problem Identification	137
6.5	Discussion	142
7	Improvements with a co-Kriging Model	145
7.1	Method Extension	145
7.2	Application on Surrogate Optimisation	146
7.3	Results on the Academic Test Case	149
7.3.1	Convergence and Final Pareto Fronts	149
7.3.2	Three Configurations Analysis	155
7.4	Results on the Industrial Test Case	155
7.4.1	Single-Objective Optimisation	155
7.4.2	Multiobjective Optimisation	159
7.5	Discussion	168
8	Influence of the Low-Fidelity Tool Accuracy on Convergence	171
8.1	Trend Agreement Visualisation	171
8.1.1	Matrix Contour Plot	172
8.1.2	Multidimensional Trend Visualisation	173

8.2	Sensitivity Removal on the Low-Fidelity Model	176
8.2.1	Modification on the Low-Fidelity Model	178
8.2.2	Visualisation	178
8.2.3	Multifidelity Optimisation Results	179
8.3	Noise Addition on the Low-Fidelity Model	182
8.3.1	Modification on the Low-Fidelity Model	182
8.3.2	Multifidelity Optimisation Results	183
8.4	Discussion	183
9	Three-Dimensional Transonic Wing Optimisation	187
9.1	Geometry, Parametrisation and Objective	187
9.2	Results	190
9.2.1	Single Objective Framework	191
9.2.2	Multiobjective Framework	193
9.3	Conclusion	198
10	Conclusion & Future Work	201
10.1	Conclusion	201
10.2	Future work	205
	Bibliography	207
A	Publications	225
A.1	Published Conference Papers	225
A.2	Journal Papers in Preparation	225
A.3	Presentations & Posters	225
B	Code Implementation	227
B.1	MultiObjective Particle Swarm Optimiser JS	227
B.2	Surrogate Models	228
B.3	Single-Objective MultiFidelity	229
B.4	MultiObjective MultiFidelity	231
C	Extra Figures	235
C.1	Multifidelity Single-Objective Optimisation	235
C.2	Multifidelity Multiobjective Optimisation	236

Contents

C.3	Influence of the low-fidelity tool accuracy on convergence	238
C.3.1	Lack of Sensitivity	238
C.3.2	Reverse Sensitivity	240
C.4	Three-Dimensional Transonic Wing Optimisation	242
D	Maximum Lift Estimation from Clean Configuration	245
D.1	Generating High-Lift Aerofoil From a Clean Configuration	245
D.2	Kinematics and Multifidelity Optimisation	246

List of Figures

1.1	Thesis contribution to knowledge.	4
2.1	Take-off procedure as in FAR	10
2.2	Landing procedure as in FAR	11
2.3	Leading edge high-lift devices systems	13
2.4	Trailing edge high-lift devices systems (Dam 2002).	14
2.5	High-lift devices parametrisations	16
2.6	Flow features around a high-lift airfoil	18
2.7	Effect of gap and lap on maximum lift on a two-dimensional aerofoil	20
3.1	Navier-Stokes Equations assumptions for numerical simulations	26
3.2	MACROS GTOpt anchors search	46
3.3	High dimensionality problems optimisation using meta-models	56
4.1	The 3-parts process to use multifidelity for optimisation.	59
4.2	Comparison of multifidelity additive, multiplicative and comprehensive corrections	63
5.1	Single-objective multifidelity optimisation framework.	75
5.2	Illustration of the Euclidean and of the step trust regions in the design and objective spaces	78
5.3	Single- and multifidelity optimisations on the 1D function	81
5.4	Convergence history of the two-dimensional high-fidelity-only and multifidelity optimisations	82
5.5	Two-dimensional optimisations path for the high-fidelity-only and the multifidelity optimisation	83
5.6	NACA 23012 FFD parametrisation	85
5.7	Convergence history for one initial trust region (TR) compared to f_{high} -only optimisation	87
5.8	NACA multifidelity optimisation convergence histories for different initial trust region sizes compared with the high-fidelity-only run	89
5.9	NACA trust region size convergence for the Euclidean (left) and step (right) trust regions.	89

List of Figures

5.10	Two-dimensional GARTEUR aerofoil geometry at take-off with the flap and slat parametrisation	90
5.11	Polar comparison between experimental data, Fluent, MSES and ESDU simulations for the Garteur test case.	93
5.12	Pressure coefficients distribution from CFD on the GARTEUR aerofoil in take-off configuration for two angles of attack and comparison with experimental data.	94
5.13	Convergence comparison between high-fidelity-only and multifidelity optimisations	98
5.14	Pressure coefficient plots and shapes associated for the multifidelity optimisations.	99
5.15	High-lift two-dimensional profile used for the industrial optimisation in datum configuration	100
5.16	Baseline configuration aerodynamic polars for the Airbus test case from TAU and HiLi.	102
5.17	Multifidelity convergence on the 2-design variables optimisation with and without restart strategy	104
5.18	Two-dimensional design space showing the path of the multifidelity optimisation.	105
5.19	Normalised high-fidelity approximation (left) and low-fidelity sampled values (right).	105
5.20	Convergence of high- and multifidelity initial, and extended (EB), boundaries optimisations with respect to the number of high-fidelity function calls (left) and in terms of elapsed time (right).	107
5.21	Single-objective industrial-case optimisation comparison with parallel-coordinates plot	108
6.1	Multiobjective multifidelity optimisation framework main process. . . .	115
6.2	Multiobjective multifidelity optimisation framework suboptimisation part. . . .	116
6.3	Multifidelity suboptimisation results with ESDU model and step trust region, at iteration 8.	118
6.4	Hypervolume definition from a normalised Pareto Front and a reference point	121
6.5	Iteration statuses distribution during the ESDU and MSES multifidelity optimisations	123

6.6	Pareto front after the same number of high-fidelity calls for the f_{high} -only and both multifidelity optimisations (ESDU and MSES)	125
6.7	From top left to bottom right: Pareto front after 10, 50, 100, 250, 500 and 1000 CFD calls for both the high- and multifidelity optimisations.	126
6.8	Hypervolume indicator convergence for the Garteau multiobjective optimisation with respect to the number of high-fidelity function calls (left) and to the wall-clock time (right).	127
6.9	Wall-clock time distributions of the multifidelity optimisations between CFD and suboptimisation	128
6.10	Parallel coordinates plot for the Pareto fronts with the f_{high} -only (blue), f_{low} -only (yellow), multifidelity ESDU (red) and multifidelity MSES (green) optimisations including all design variables and the two high-fidelity objectives	129
6.11	On top: normalised parallel-coordinates plot of the selected configurations representing the low-drag, compromise and high-lift configurations for the high- and both multifidelity optimisations; on bottom: selected configurations geometries.	132
6.12	Lift and drag coefficients breakdown for both high- and multifidelity optimisations, for the selected configuration (high-lift, compromise — <i>compr.</i> — and low-drag)	133
6.13	Pareto front after approximatively 250 CFD calls for the industrial test case.	135
6.14	Multifidelity iteration statuses for the three optimisations.	137
6.15	Hypervolume indicator evolution during convergence of all optimisers	139
6.16	Pareto front and parallel coordinates for all optimisers	141
6.17	Comparison between corrected and high-fidelity objectives.	143
7.1	Hypervolume convergence for the optimisation on the CFD surrogate with the co-Kriging correction.	147
7.2	Pareto front after the optimisation on the CFD surrogate with the co-Kriging correction.	147
7.3	Comparison between corrected and high-fidelity objectives with co-Kriging correction.	148
7.4	Convergence indicators for the academic test case using a co-Kriging correction	150

List of Figures

7.5	Iteration statuses for all multifidelity optimisation convergence (academic test case with co-Kriging corrected model)	151
7.6	Time repartition for the multifidelity optimisation when MSES is used as low-fidelity tool (top) and trust region size along with the number of corrected function calls.	153
7.7	Pareto fronts after approximatively 700 CFD calls for multiobjective optimisations using co-Kriging as correction.	154
7.8	Comparison of high- c_l , compromise and low- c_d points for all methods for the multiobjective optimisation using a co-Kriging correction with a parallel coordinates plot on top and with the final geometries on bottom.	156
7.9	Convergence of high- and multifidelity EB boundaries optimisations with respect to the number of high-fidelity function calls (left) and in terms of elapsed time (right).	158
7.10	Single-objective industrial-case optimisation comparison with parallel-coordinates plot	159
7.11	Hypervolume indicator convergence for the multiobjective multifidelity optimisation against the number of high-fidelity calls on the right and against the elapsed time on the left.	161
7.12	Time distribution for the multifidelity optimisation (left and middle) on the industrial case and comparison between CPU time and number of CFD calls for multifidelity (MF) and high-fidelity-only (right).	163
7.13	Suboptimisation wall-clock time comparison between academic and industrial cases.	163
7.14	Multiobjective optimisation Pareto front comparison	165
7.15	Parallel coordinates plot (top) and geometries (bottom) for the anchors and compromise points for each method.	167
7.16	Normalised maximum lift coefficients comparison between multiobjective high- c_l (blue) and single-objective with small bounds (green).	168
8.1	Three dimensional function with cuts organised in a contour matrix plot, coloured by high-fidelity function value.	172
8.2	Contour matrix plot of the high-fidelity function (TAU) at the datum point.	174
8.3	Multifidelity trends visualisation in high-dimension space and analogy with one-dimension function.	176

8.4	Multifidelity matrix contour plot for the maximum c_l predicted by TAU (high fidelity) and by HiLi (low-fidelity).	177
8.5	Normalised convergence with respect to the number of high-fidelity function calls (left) and to the wall-clock time (right) for the sensitivity modifications on the low-fidelity model.	180
8.6	Parallel-coordinates plot showing the normalised optimum for each optimisation.	181
8.7	Normalised convergence with respect to the number of high-fidelity function calls (left) and to the wall-clock time (right) for the noise modifications on the low-fidelity model.	185
8.8	Optimum accuracy and convergence speed evolution with respect to the sensitivity and the noise level.	185
9.1	XRF1 coarse mesh top-view	189
9.2	Convergence comparison between multifidelity (single-objective formulation) and high-fidelity-only XRF1 optimisations	192
9.3	Suboptimisation convergence during multifidelity (single-objective formulation) XRF1 optimisation	192
9.4	Parallel coordinates plot for single- and multifidelity XRF1 optimisation convergences	194
9.5	Convergence comparison between multifidelity (multiobjective formulation) and high-fidelity-only XRF1 optimisations	196
9.6	Parallel coordinates plot for single- and multifidelity optima from XRF1 optimisations.	197
9.7	Spanwise lift distribution for the XRF1 optimum configurations.	199
B.1	MOPSO JS simplified UML diagram.	227
B.2	Surrogate models simplified UML diagram.	228
B.3	Single-objective multifidelity code simplified UML diagram.	229
B.4	Single-objective multifidelity code detailed UML diagram.	229
B.5	Single-objective multifidelity code detailed UML diagram (end).	230
B.6	MultiObjective multifidelity code simplified UML diagram.	231
B.7	MultiObjective multifidelity code detailed UML diagram (end).	232
B.8	MultiObjective multifidelity code detailed UML diagram.	233
C.1	Fluent (high-fidelity) and wind tunnel validation.	235

List of Figures

C.2	Grid convergence for the Garteur Fluent CFD.	235
C.3	Time repartition for the multifidelity optimisation when ESDU is used as low-fidelity tool (top) and trust region size along with number of high-fidelity function calls.	236
C.4	Time repartition for the multifidelity optimisation when MSES is used as low-fidelity tool without low-fidelity sample (top) and trust region size along with number of high-fidelity function calls.	237
C.5	Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with no sensitivity on the slat (modified low-fidelity)	238
C.6	Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with no sensitivity on the flap (modified low-fidelity)	239
C.7	Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with reverse sensitivity on the slat (modified low-fidelity)	240
C.8	Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with reverse sensitivity on the flap (modified low-fidelity)	241
C.9	Coarse and fine meshes of the XRF1 wing.	242
C.10	Spanwise wave drag distribution for XRF1 the optimum configurations.	243
D.1	High-level framework for load and maximum lift estimation from a clean aerofoil.	247
D.2	Clean aerofoil and parametrised slat and flap.	248
D.3	Example of two maximum deflection configurations leading to two single hinge positions.	248
D.4	A single drop hinge geometrical construction from stowed (blue) and deployed (red) configuration.	249
D.5	Proposed aerodynamics multifidelity multiobjective optimisation framework for single-hinge including kinematics.	250
D.6	Multiobjective kinematic optimisation initial results	250

List of Tables

2.1	Summary of requirements for take-off and landing (Reckzeh 2003) . . .	11
3.1	Hypervolume indicator values and standard deviation (Std) for different MOPSO parameters and comparison with MACROS.	45
5.1	NACA 23012 optimisation parameters from (Jarrett and Ghisu 2012). .	84
5.2	Multifidelity parameters for the NACA test case.	85
5.3	Comparison between high-fidelity-only and multifidelity optimisations for the NACA test case	87
5.4	High-lift optimisation aerodynamic parameters.	91
5.5	Multifidelity parameters for the Garteur single-objective test case. . . .	95
5.6	Normalized optimisation boundaries values for the industrial test case.	100
5.7	High-lift aerodynamic parameters for the industrial test case.	101
5.8	Multifidelity parameters for the industrial single-objective test case. . .	103
5.9	Optimisation starting and boundaries values for the initial boundaries case, normalised by variable range.	106
5.10	Subplex running times	106
5.11	Optimisation normalised starting and boundaries values for the extended boundaries case.	109
6.1	Multifidelity parameters for the Garteur multiobjective test case. . . .	122
6.2	Surrogate information constructed by Macros to replace CFD	138
7.1	Savings provided by multifidelity to achieve an hypervolume indicator of 0.55 for the number of CFD calls and 0.60 for the time.	151
7.2	Trust region parameters for the single-objective extended bounds multifidelity optimisation.	157
7.3	Trust region parameters for the multiobjective multifidelity optimisation.	160
7.4	Savings provided by each methods compared to high-fidelity-only to achieve an hypervolume indicator of 0.64 and 0.66 for the number for the CPU time (industrial case).	162
9.1	Aerodynamic parameters for the XRF1 wing.	188
9.2	Initial and boundaries values for the XRF1 test case.	189
9.3	Datum integrated aerodynamic coefficients for both meshes.	190

List of Tables

9.4	Multifidelity parameters for the initial XRF1 optimisation.	191
9.5	Multiobjective multifidelity framework parameters for the XRF1 single-objective optimisation.	194
9.6	Optimum configurations lift and drag for XRF1 optimisation.	197

Nomenclature

Superscripts

(i)	i^{th} coordinate
$\bar{\cdot}$	Time averaged quantities
$\cdot'$	Fluctuating quantities

Subscripts

$\hat{\cdot}$	Predicted output
0	Initial
approx	Approximation
corr	Corrected
d	Relating to the difference model
high	High-fidelity
k	Relative to current iteration
l	Relating to the low-fidelity model
low	Low-fidelity
nf	No flap
ns	No slat
rf	Reverse flap
rs	Reverse slat
s	Suboptimiser optimum

Variables

Δ	Vector of differences for co-Kriging
Δ	Trust region size
δ_{bl}	Boundary layer thickness
δ_{ij}	Kronecker delta
η	Noise function (from RBF interpolation)
ε	MACROS GTOpt step size
ε_t	Random turbulence for MOPSO

Nomenclature

ϵ_c	Gaussian RBF correlation factor or shape parameter
θ_f	Flap deflection
γ_{climb}	Climb rate
γ_e	Trust region expansion coefficient
γ_s	Trust region shrink coefficient
$\boldsymbol{\lambda}$	Vector of RBF coefficients
$\hat{\mu}$	Maximum likelihood mean estimate
μ	Training data mean
μ_r	Single-objective convergence runs mean
μ_t	Turbulent viscosity
Φ	RBF Gram matrix
Ψ	Correlation matrix for Kriging and coKriging interpolation
ϕ	RBF basis function
ϕ_f	Flow potential
$\phi_{i,j}$	Gram matrix elements at i, j
ρ	Ratio of improvement
ρ_f	Fluid density
$\hat{\sigma}$	Maximum likelihood variance estimate
θ_s	Slat deflection
s_{ij}	i, j viscous stress tensor components
σ_r	Single-objective convergence runs standard deviation
τ_g	Tyre/tarmac friction coefficient
$\boldsymbol{\xi}$	Kriging and co-Kriging width hyperparameters vector
ζ	Noise level
ζ_z	Transformation function for space mapping with parameters $\mathbf{z}$
$\mathbf{1}$	Vector of ones
$\mathbf{C}$	co-Kriging full correlation matrix
$\mathbf{c}$	Noise RBF control point
c_1	Personal factor for MOPSO
c_2	Social factor for MOPSO
C_D	Aircraft drag coefficient
$C_{L_{\text{appr}}}$	Aircraft approach lift
$C_{L_{V_2}}$	Lift coefficient at V_2
$C_{L_{\text{max}}}$	Wing maximum lift
cor	Correlation function

cov	Covariance function
c_r	Contribution rate
$\mathbf{D}$	training samples matrix
$\mathbf{d}$	Training point coordinates vector
$\mathcal{D}$	Design space
d	Euclidean distance
$\mathbf{Y}$	Vector of sampled function values for RBF
E	Fluid energy
$E[I(x)]$	Expected Improvement function
$\mathbf{e}_k$	Surrogate model of the error between low- and high-fidelity functions at iteration k
$\mathbf{f}$	Multioutput objective function
$\mathbf{F}_0$	Initial list of fitness values for MOPSO
$f_{\text{corr}}^{\text{SM}}$	Space mapping corrected function
g_f	Flap gap
l_f	Flap lap
$\tilde{f}_{\text{low},\zeta}$	Low-fidelity function modified with ζ noise level
$\tilde{f}$	Modified low-fidelity function
G	Gaussian process
$\mathbf{h}$	Hinge point
I_h	Hypervolume indicator
I_{hs}	Surrogate Pareto front hypervolume indicator
k	Kinetic energy
L	Likelihood for Kriging and co-Kriging
l	Objective space dimension
L/D	Lift over drag coefficient
L_d	Deployed leading edge
L_s	Stowed leading edge
m_p	Number of Pareto point
n	Design space dimension
$\tilde{n}_{cp}$	Number of control point for the low-fidelity noise function
n_f	Number of objective required for <i>good</i> iteration
n_p	Number of Pareto front to compare
n_p	Number of points required for <i>good</i> iteration
$\mathbf{P}$	Swarm of particles for MOPSO

Nomenclature

$\mathbf{p}$	Point in objective space
$\mathcal{P}$	Pareto front set
p	Number of sampling points
$\mathbf{P}_c$	Compromise point
PF	Pareto Front
PF'	Reduced Pareto Front
PF_s	Surrogate Pareto Front
p_h	Number of high-fidelity sampling points
p_l	Number of low-fidelity sampling points
$\mathbf{P}_n$	Nadir point
p_r	Fluid pressure
$\mathbf{P}_u$	Utopia point
$\tilde{\mathbf{p}}$	Normalised point by utopia and nadir points in objective space
$\mathbf{q}$	Kriging and co-Kriging exponent hyperparameters vector
q	Heat flux
$\mathbf{r}$	Correlation vector between sampling points and new point
r_1	Reference point coordinate 1
r_2	Reference point coordinate 2
r_e	Relaxation parameter for MOPSO
Re	Reynolds number
$\mathbf{r}_p$	Hypervolume indicator reference point
$\mathbf{r}_p$	Reference point
$\hat{s}^2$	Mean squared error (MSE) prediction
g_s	Slat gap
l_s	Slat lap
$\mathcal{T}$	Trust region
t	Time
T_d	Deployed trailing edge
T_r	Reverse thrust
TR_{lim}	Trust region size limit
TR_{lim}	Trust region size limit
T_s	Stowed trailing edge
T/W	Thrust-over-weight ratio
$\mathcal{U}$	Uniform distribution (between 0 and 1)
u_j	j^{th} fluid velocity component

$\mathbf{v}$	Velocity of a particle for MOPSO
$\mathbf{V}_0$	Initial list of velocities for MOPSO
V_2	2 nd segment speed
V_A	Approach speed
V_{LOF}	Lift-off speed
$V_{S_{1g}}$	1g-stall speed
$V_{S_{min}}$	Dynamic stall speed
w	Inertia for MOPSO
W/S	Wing loading
$\mathbf{x}$	Design vector
$\mathbf{X}_0$	Initial list of positions for MOPSO
$\mathbf{y}$	Function output vector
$\hat{y}$	Maximum likelihood prediction
y^+	Dimensionless wall distance
$\mathbf{z}$	Space mapping parameters

Acronyms

AASO	Adaptive Approximation in Single Optimisation
AMMO	Approximation and Model Management Optimiser
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BLE	Boundary Layer Equations
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
DoE	Design of Experiments
EA	Evolutionary Algorithm
EASA	European Aviation Safety Agency
EB	Extended Bounds
EFT	Enhanced Fidelity Transonic wing
EI	Expected Improvement
ESDU	Engineering Sciences Data Unit
ES	Evolutionary Strategy
FAR	Federal Aviation Regulations
FFD	Free Form Deformation
HPC	High Performance Computer

Nomenclature

LHS	Latin Hypercube Sampling
MOGA	MultiObjective Genetic Algorithm
MOPSO	MultiObjective Particle Swarm Optimiser
NSGA-II	Nondominated Sorting Genetic Algorithm-II
NS	Navier-Stokes
NURBS	Non-Uniform Rational B-Spline
PSO	Particle Swarm Optimiser
QNTR	Quasi-Newton Trust Region
RANS	Reynolds Averaged Navier-Stokes
SBO	Surrogate Based Optimisation
SBPLX	Subplex
SB	Small Bounds
TR	Trust Region
UML	Unified Modelling Language

Introduction

1.1 Context

Over the last 40 years, the aircraft industry has improved to meet the requirements from airlines, passengers and regulation authorities: airlines and passengers push for low operational cost, including aerodynamic efficiency, maintenance and reliability while authorities impose regulations and set challenging targets on emission and noise levels (Argüelles and al. 2001; Green 2005; R. Wahls 2010). Moreover, aircraft manufacturers are also subjected to the market and they need methods to quickly develop new concepts to efficiently respond to market demands. Naturally, all actors are equally concerned by safety which is also formally addressed in regulations such as the Federal Aviation Regulations (FAR) and European Aviation Safety Agency (EASA) regulations. Aircraft manufacturers are also planning disruptive technologies that contain a higher degree of risk and cannot rely solely on past aircraft knowledge. Therefore, in order to reliably and quickly produce feasible new aircraft, a complex physic understanding relying on simulations and fast design methods need to be made available to the designers in the near future. This is achieved with simulation models and the use of optimisation methods to enable quick design space exploration and efficient optimal design.

Furthermore, to reduce design time and cost, designers cannot rely solely on physical experiments such as wind tunnel and flight test. So, during the past decades and thanks to the advent of computational power, simulation has been widely used at various degrees of fidelity: first computational codes implemented simplified equations using empirical corrections while nowadays more complete resolution of the flow governing equations are possible and used in day-to-day design. Those simulation models do not use the same design maturity and therefore are not using the same geometrical representation making their use relevant at various design maturity stages. To allow for more accurate optimisation earlier in the design stage, optimisation methods should rely on multiple tools to provide design hints: the concept stage should benefit from complex simulations to decrease choice uncertainty while detailed design can benefit from the lower fidelity tool to decrease optimisation com-

putational time. This thesis uses the concept of multifidelity simulations to take into consideration such aerodynamic tools pool for the aircraft aerodynamic design.

The Airbus-led Enhanced Fidelity Transonic Wing (EFT) project aims at addressing complex physic phenomena simulation by focusing on the:

1. wing maximum lift prediction,
2. transonic drag rise prediction,
3. wing aeroelasticity prediction for aircraft drag estimation,
4. and the use of more theoretical methods for the development of disruptive concepts.

As part of the EFT project, this work focuses on the aircraft maximum lift prediction. Maximum lift is achieved by deploying high-lift devices up- and downstream of the wing and is required at slow speed ie for safe take-offs and landings. Early in the design stage, before any detailed geometry is available, the maximum lift estimation is performed by creating devices from the cruise shape, and finding the optimum positions providing the maximum lift, lift-to-drag ratio and minimum drag. This is achieved by performing optimisation, with multiple aerodynamic tools, hence with the use of multifidelity optimisation. Conflicting objectives are treated with a specific multiobjective formulation providing a set of solutions instead of a unique compromise solution.

1.2 Aim and Objectives

The aim of this thesis is the development and assessment of an optimisation method that takes advantage of the multiple levels of fidelity available during the high-lift wing design process. To achieve it, the following objectives are drawn and fulfilled in this thesis:

1. Implement a multifidelity optimiser taking into account the high-lift aerodynamics specificities;
2. Perform a multifidelity optimisation on a high-lift aerofoil at maximum lift conditions;
3. Extend the multifidelity optimiser definition to a multiobjective formulation to take into account conflicting requirements;

4. Assess the effect of the low-fidelity model on the multifidelity optimisation convergence and optimum;
5. Apply the multifidelity optimisation to three-dimensional geometries.

1.3 Contribution to Knowledge

The methodology starts from a trust-region based multifidelity method found in the literature and is then extended to a multiobjective context. In addition, the multifidelity multiobjective optimisation is applied to multielement aerofoil position optimisation which is a complex aerodynamic function. The test cases and aerodynamic tools are representative of an industrial context. Because a picture is worth a thousand words, figure 1.1 presents the thesis contribution to knowledge and can be summarised as the extension of the trust-region based optimisation algorithm, that has been proven to work well on multielement aerofoil (Alexandrov, Nielsen, et al. 2000), to a multiobjective formulation. The figure shows existing publications dealing with multifidelity or multiobjective optimisation applied to multielement aerofoil. No work is found that deals with those three topics at the same time: an in-depth literature review supporting this claim is presented in chapter 4.

For the assessment of low-fidelity accuracy on the final solutions, visualisation techniques are also introduced for the first time in this work.

1.4 Overview

The thesis structure is as follow:

Chapter 1 described the context of the work and defined the thesis aim and objectives.

Chapter 2 is dedicated to the high-lift devices presentation. First the industrial design process is described to identify design maturity of interest and the optimisation objectives are highlighted. In a second part, the mechanical systems available are introduced to identify non-aerodynamic constraints. Finally, the aerodynamic flow physics involved around a multielement aerofoil is explained and used to draw the model-to-optimize specificities.

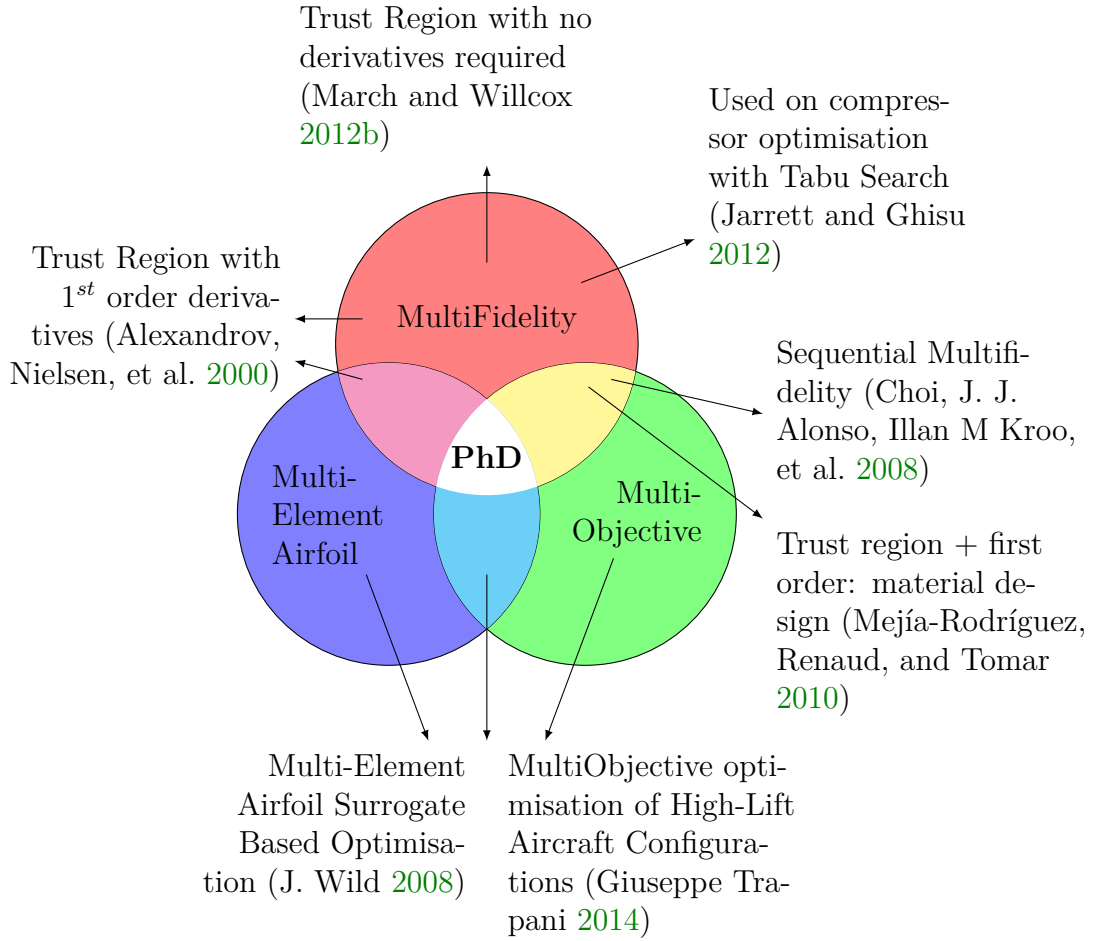


Figure 1.1: Thesis contribution to knowledge.

Chapter 3 links the high-lift aerodynamics to the optimisation world. The chapter first treats the models for computational aerodynamics, oriented for high-lift aerofoils. Secondly, the optimisation definitions and state-of-the art work are presented and finally complemented with surrogate modelling, used to alleviate the high cost associated with the numerical optimisation of the Navier-Stokes equations.

Chapter 4 is the literature review on multifidelity multiobjective optimisation applied to multielement aerofoils. At first the chapter defines the various aerodynamic models that can be encountered in the literature. Then methods to fuse the various fidelity models are presented and the distinct model managements introduced. The last two sections are dedicated to the applications found in the

literature on multiobjective optimisation and optimisation applied on high-lift aerofoils.

Chapter 5 presents the multifidelity single-objective implementation developed by the author and validate it by performing optimisation first on mathematical functions, then on a two-dimensional single-element aerofoil. Finally, the method is applied on two different geometries using two different sets of models.

Chapter 6 extends the methodology to multiobjective problems and applies the method to the two multielement aerofoil optimisation cases defined previously. An extension of the method is then introduced to overcome non-optimal solution encountered.

Chapter 8 quantifies the effect of the low-fidelity accuracy on the multifidelity single-objective optimisation by artificially introducing noise and sensitivity change in the low-fidelity function. A visualisation technique is also introduced to analyse the discrepancies between low- and high-fidelity functions.

Chapter 9 applies the method to a more computational expensive process: the optimisation of a three-dimensional wing defined by the sponsor. The flight domain is changed: the optimisation is done at transonic conditions.

Chapter 10 concludes by summarising the findings. Future considerations are introduced, especially the use of kinematics inside the multifidelity optimisation loop to represent more accurately the constraints introduced by the real systems.

High-Lift Design

Aircraft manufacturers design their products to achieve the best quality for the lowest cost. Of course, airlines operating the aircraft are also leaning towards efficient aircraft that will lead to the lowest Direct Operating Cost (DOC) which is the cost associated with everything needed to fly the aircraft. A large part of this cost can be accounted to fuel ie aircraft efficiency, and maintenance cost (ICAO 2017).

Reduction in fuel consumption is achieved by reducing the cruise drag. To achieve this reduction, aircraft manufacturers have reduced the wing surface. In return, to allow landing and take-off on a reasonable runway length, the aircraft speed must be reduced resulting in a change in wing shape to increase the lift in this critical operation part, without affecting the cruise shape. This is where the high-lift devices come into play.

This chapter introduces the high-lift devices design in section 2.1, first from an industrial point of view, focusing on the *as is* situation. The aerodynamic quantities of interest are studied in section 2.2 and then the complexity involved in designing high-lift devices is highlighted, first in terms of kinematic considerations in section 2.3, second on the intricacies brought by maximum lift prediction in section 2.4. The thesis subject is finally formalised for an optimisation context in section 2.5.

2.1 Industrial Design Process

An aircraft can take up to 8 years to be developed, from requirements capture to first delivery. As seen previously, manufacturers must take into account many factors which puts constraints on the design process. To circumvent these difficulties, the process is normally separated in 3 main stages: 1. conceptual design, 2. preliminary design, and 3. detailed design. Using this approach allows the requirements definition and the design features selection in an order that minimises risk. High-lift devices design follow the same approach:

Pre-development The high-lift devices are iteratively designed based on the preliminary cruise wing. This is mainly done with semi-empirical methods (lifting line and lifting surface corrected by empirical databases). When satisfying config-

urations are found, they are tested on simple wind tunnel models (2D effects) followed by more finer models (including 3D effects and nacelles influence). The selected configuration is then analysed in more details with the wind tunnel to demonstrate the aerodynamic performance and that the safety requirements are fulfilled.

Development This phase corresponds to the adaptation of the configuration to the final cruise shape taking into account structures and kinematics. The position settings and fine geometric details are optimised to improve the maximum lift coefficient $C_{L_{\max}}$ and the lift-to-drag ratio L/D . The aerodynamic performances are estimated using Reynolds Averaged Navier-Stokes (RANS) Computational Fluid Dynamics (CFD) and intensive wind tunnel tests; they are then scaled to flight conditions using empirical corrections.

Pre-flight This occurs after the design freeze to create a final geometry that can be thoroughly tested in wind tunnels. This produces high-confidence database that will be checked in flight tests.

Hence, at each stage, a different tool is used to assess the performance of the studied configuration. Those tools require different geometry representations, from very coarse which defines only the planform, to very fine including the gaps between elements. The higher the certainty on the geometry, the higher the tools accuracy. This statement is directly linked to the aim of this work: as we progress in the design more and more tools to estimate the aerodynamic performances are available. It seems therefore natural to use this information in a way that can benefit the design process steps.

Now that the availability of multiple tools for performance evaluation have been justified, it is necessary to shed light on the objectives selected for the high-lift devices performance optimisation.

2.2 Design Objectives for High-Lift Devices

Any aircraft used in United States of America must meet requirements defined by the Federal Aviation Regulations (FAR) which ensure that the aircraft is safe to operate. Similar authorities exist in other part of the world such as the European Aviation Safety Agency (EASA) in Europe, but their guidelines are usually aligned on the FAR. To be a commercial success, aircraft performance, operating cost, and safety

standards must be carefully monitored and optimised. This section aims at defining precisely the constraints and requirements for take-off and landing in terms of safety but also in terms of operating cost.

2.2.1 Take-Off

The regulations (Federal Aviation Regulations 2002) for take-off are summarised in figure 2.1 and require (Rudolph 1996):

1. the lift-off speed V_{LOF} to be greater or equal to 1.1 times the minimum *unstuck* speed V_{MU} which is a function of the maximum lift $C_{L_{max}}$ and also a function of the maximum rotation allowed by the tail;
2. the speed V_2 , achieved after a 35 ft obstacle, to be greater or equal to 1.1 times the minimum control speed (with one engine inoperative) V_{mc} and also greater or equal than 1.2 times the dynamic stall speed $V_{S_{min}} \approx 95\%V_{S_{1g}}$ (the 1g-stall speed); equivalently, the lift needed at V_2 , noted $C_{L_{V_2}}$ must be lower than $C_{L_{max}}/1.14^2$;
3. the climb rate γ_{climb} with one engine inoperative at V_2 , function of the thrust-over-weight ratio and of the lift-over-drag, to be greater or equal to 2.3% for a 2 engines aircraft (3.0% for a 4 engines aircraft).

On top of those requirements,

4. the take-off distance (as shown in figure 2.1) should be minimised in order to access the highest number of airports and in order to reduce the noise; the acceleration on the ground is a function of the: thrust-over-weight ratio, wing loading, maximum lift coefficient, drag and ground friction coefficient (Reckzeh 2003), and the airborne distance being a function of the climb index.

To sum up, in terms of aerodynamic high-lift devices, the take-off requires a high $C_{L_{max}}$ and high L/D while the drag C_D and the wing loading W/S must remain as low as possible (Reckzeh 2003).

2.2.2 Landing

Similarly, FAR requirements are summarised in figure 2.2 and require:

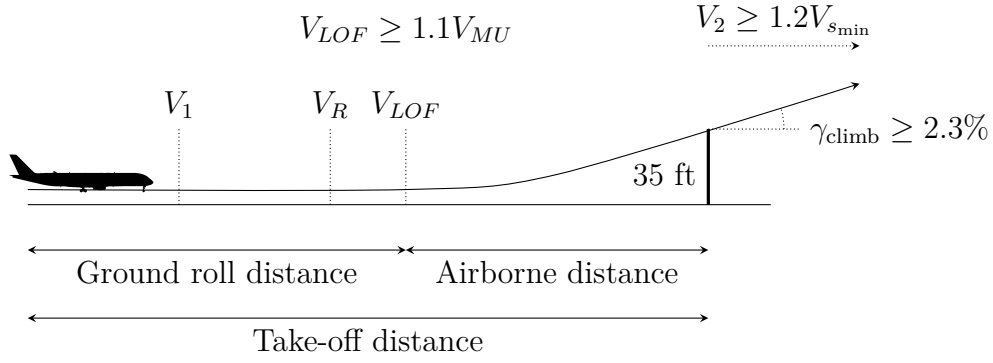


Figure 2.1: Take-off procedure as in FAR, from (Flaig and Hilbilg 1991).

1. the approach speed V_A , to be greater or equal to 1.3 times the dynamic stall speed $V_{S_{min}}$. As previously, in terms of lift, it requires the approach lift $C_{L_{appr}}$ to be greater or equal to $C_{L_{max}}/1.54$;
2. the pilot to have a good visibility while landing which limits the maximum pitch attitude of the aircraft, also limited during the flare and touch down to avoid tail strikes.
3. the climb index after a go-around (at a speed $1.23V_{S_{1g}}$ and landing gears down) to be no less than 3.2% with all engines operative, which imposes a higher lift-over-drag than for landing.

On top of that,

4. The approach speed should be minimised in order to decrease the probabilities of an accident, to reduce the braking distance after touchdown and to limit the tire wear (Rudolph 1996). The achievable speed is driven by the maximum lift coefficient $C_{L_{max}}$ and by the wing loading W/S .
5. the landing distance should be minimised by minimising the landing speed and the braking power. The approach angle has to be maximised (Reckzeh 2003), without affecting the maximum lift (required to achieve low speed); hence the drag has to be maximised.

The requirements presented above are summarised in table 2.1. Despite being very dependent on the aircraft configuration, the aerodynamic design of high lift devices aims at finding a trade-off between maximum usable lift coefficient and lift-to-drag

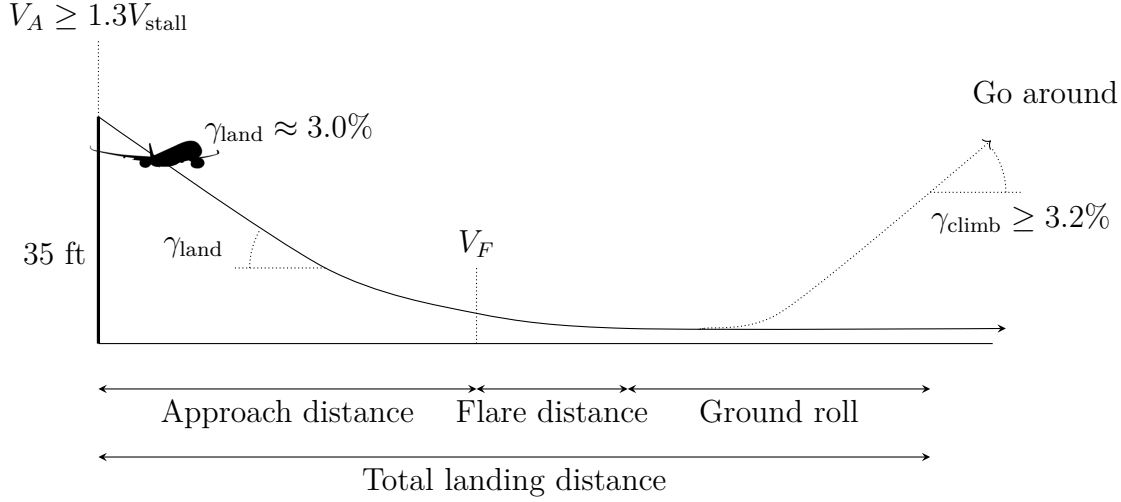


Figure 2.2: Landing procedure as in FAR, from (Flaig and Hilbilg 1991).

Table 2.1: Summary of requirements for take-off and landing (Reckzeh 2003)

#	Function of	Optimisation
T.O.	1. $C_{L_{\max}}$, α at $C_{L_{TO}}$	$\max [C_{L_{\max}}]$ and $\min [\alpha_{C_{L_{TO}}}]$
	2. $C_{L_{\max}}$, L/D	$\max [C_{L_{\max}}, L/D]$
	3. T/W , L/D	$\max [L/D, \max T]$
	4. T/W , W/S , $C_{L_{\max}}$, C_D , τ_g , γ_{climb}	$\max [T, C_{L_{\max}}, \tau_g]$ and $\min [W/S, C_D]$
Landing	1. $C_{L_{\max}}$	$\max [C_{L_{\max}}]$
	2. α at $C_{L_{\max}}$, tail	$\min [\alpha_{\max}]$
	3. T/W , L/D	$\max [L/D, \max T]$
	4. W/S , $C_{L_{\max}}$	$\max [C_{L_{\max}}]$ and $\min [W/S]$
	5. Reverse thrust T_r , C_D , τ_g	$\max [T_r, C_D, \tau_g]$

ratio performances. According to Flaig and Hilbilg (1991), the aircraft maximum lift coefficient, and hence the high-lift system complexity, is defined by the landing case.

The points developed in this section are solely focused on the objective of high-lift devices. However to achieve those requirements, a physical system needs to be present to actuate the devices in order to position them to produce the respective performances. This aspect is considered in the following section.

2.3 Existing Systems

While the high-lift devices are used to increase the maximum lift of an aeroplane during take off and landing, they have an important impact on its overall performances and external noise (Meredith 1991). Nowadays, in addition to maximizing the lift, the design is focused on the decrease of the complexity and the complete systems weight. High-lift devices complexity appears at many design levels: because of the complex flow involved, the design and tests are time consuming and expensive. Also, the vast amount of parts involved increases the weight, and the production and maintenance cost (Dam 2002).

High-lift devices are usually made of two elements: the *slat* which once folded is the leading edge of the main wing and the *flap* which once folded is the trailing edge of the main wing. Those two parts can actually be composed of more than one aerofoil. The actuation of those parts are of paramount importance to allow a large variety of placement for a minimum decrease in the overall aircraft performance. An exhaustive literature review of the systems can be found in Rudolph (1996) and C. Niu and M. Niu (1999), the highlights are presented below.

2.3.1 Leading Edge Devices

Leading edge devices (figure 2.3) can be separated in three types from the simplest to the most complex (J. Wild and Brezillon 2005):

Krueger is a simple flat plate deployed from the leading edge lower side. It usually has only two positions (cruise and deployed) which makes it less effective for take off than more complex systems. However, because of its simplicity, it is also lighter and less expensive to maintain. Finally, when retracted, only the lower surface presents small gaps and therefore makes it particularly interesting for laminar flow wing (Dam 2002).

Droop nose has a more complex shape and actuation system that can only be deployed by rotation, leaving no gap between itself and the main wing. Also called *sealed* slats, their actuation system allows a deployment in different positions for take off and landing.

Slat has in addition of the rotation, the possibility to be translated to leave a gap between its trailing edge and the leading edge of the main element downstream.

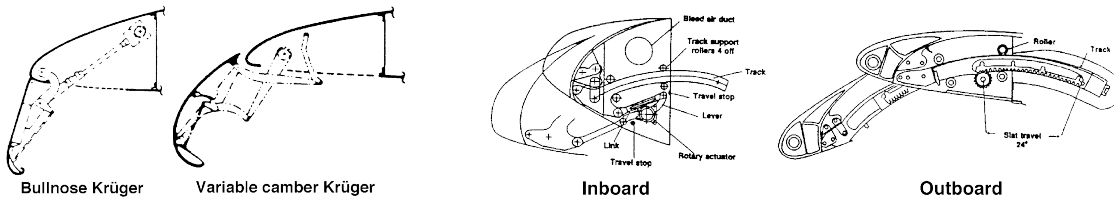


Figure 2.3: First two on the left: Inboard bullnose Kruger and outboard variable camber Kruger on B747. Second two on the right: Slat actuation and support mechanisms on A330/340 (Dam 2002).

Earlier designs used two positions, but nowadays three positions slats are more common. For take off, the preferred configuration is slotted to maximize lift and reduce drag but for landing the configuration is aeroplane dependent.

2.3.2 Trailing Edge Devices

Trailing edge devices, namely flaps, can be classified based on the number of elements and their actuation system. The majority of aircraft uses the *Fowler* movement: the aft translation movement of the flaps is resulting in an increase of the planform area (Dam 2002). Older aircraft were using triple slotted flaps to achieve the required lift but with the advance in total mass and aerodynamics, single and double slotted flaps are preferred nowadays. In addition to the maximum lift, other parameters have to be considered for the choice of the flaps configuration: the actuation system has an important effect on the aerodynamic efficiency, from simplest to more complex:

Fixed hinge systems are very simple but present very restricted Fowler movement which can be extended by using bigger flap fairing (resulting in more cruise drag, figure 2.4a).

Linkage systems allow more freedom in the flap position to the detriment of a more complex system. However the fairing can be reduced compared to fixed hinge (figure 2.4b).

Track systems are the most permissible in terms of flap movement freedom. However those systems are heavier than the two previous ones (figure 2.4c).

Hybrid systems combine the two previous systems to offer a compromise. This is for

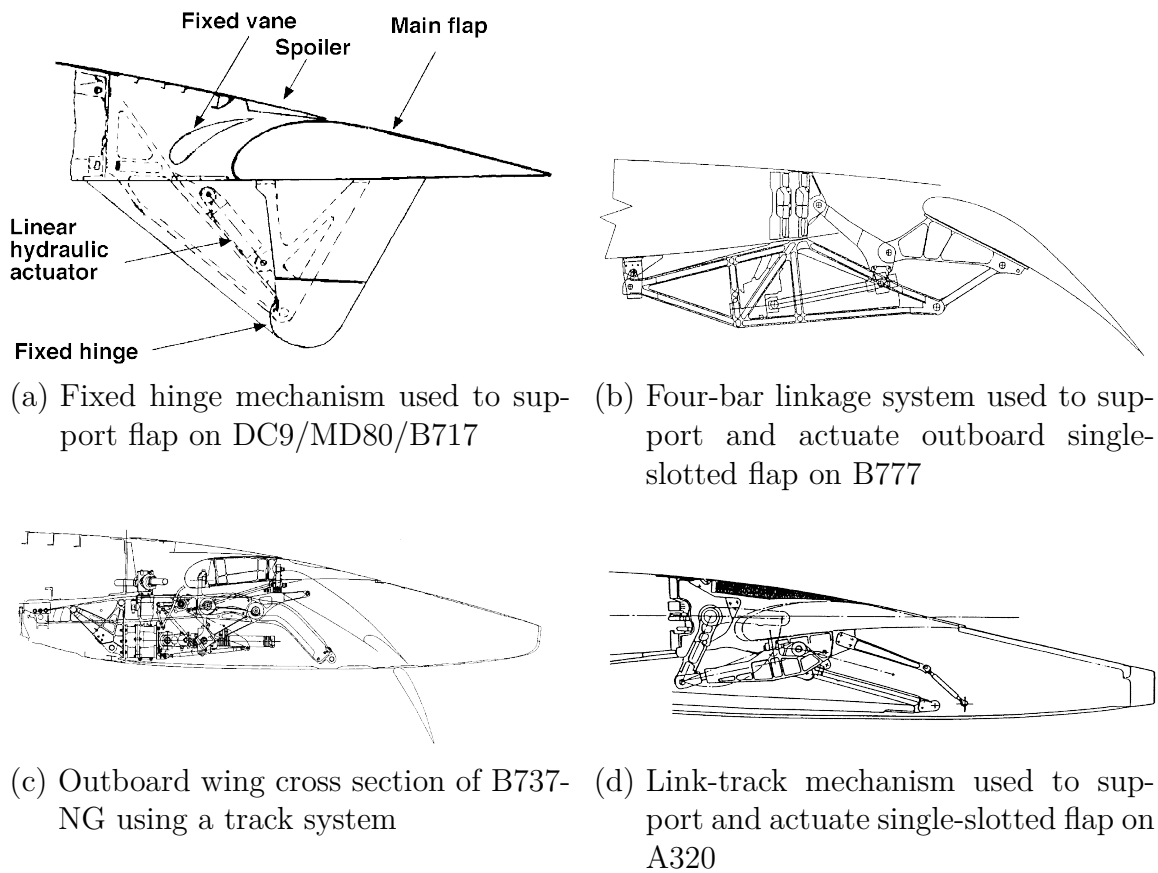


Figure 2.4: Trailing edge high-lift devices systems (Dam 2002).

example the solution retained for the Airbus A320 aircraft (Flaig and Hilbilg 1991) (figure 2.4d).

2.3.3 Kinematics as Constraints

The requirements presented in the previous section are based on the fact that high-lift devices will deploy. The reliability of such devices is therefore of paramount importance. The more pieces needed for the kinematics, the more likely a failure will occur. Hence, to prevent any fatal accident the maintenance must be increased. A compromise appears: for best aerodynamic efficiency, a complex kinematics may be needed to achieve the positions for take off and landing, but this complexity will be paid in maintenance, cruise drag and weight increase.

Finally part of the governmental objectives for the next decades is the reduction of aircraft noise during landing and take-off. This noise is two-fold: the direct aero-

dynamic noise from the wing and high-lift devices, originating from pressure levels change caused by the turbulent boundary layer, separation and vortices (Brooks, Pope, and Marcolini 1989), and the noise from the engines depending on the aircraft aerodynamic efficiency in take off and landing configurations amongst other (Rudolph 1996).

Therefore to be complete, a high-lift devices study should include all those aspects. The future work section (Chapter 10.2) extends the work for such considerations, but the rest of the thesis is focused on the aerodynamic side of high-lift devices design.

2.3.4 High-Lift Devices Parametrisation

The purpose of the kinematics is to move parts of the aerofoil in the vicinity of the wing. This transformation causes a change in the geometry that needs to be reflected in the model. The aerodynamic performance estimation requires the use of accurate geometry representation. Depending on the equations solved this required accuracy varies and ranges from simple planform view to highly accurate 3D models. Generally, the geometry is created in a Computational-Aided Design (CAD) software but in order to be included in an optimisation process, the geometry shape needs to be changed according to parameters in what is called *parametrisation*. The parametrisation reduces the number of variables, compared to the use of all geometry points, by including some physical knowledge in the relation between each point. Thus parametrisation allows to reduce the number of variables while representing a maximum number of feasible geometrical representations. For high-lift design, three types can be distinguished: outline, position and shape parameters (J. Wild 2008).

Outline Parameters Outline parameters are all the parameters that define the high-lift devices in the planform view (view from the top of the wing), see figure 2.5a. This includes for example, the device spanwise and chord extents but also the wing planform. This is rather a high-level definition that is preferably used in early design phases since it does not include thickness or any shape-related characteristics.

Position Parameters Position parameters relate to the relative position of deployed flaps and slats with respect to the main wing. This type can be divided in two main parametrisations: *Cartesian coordinates* (figure 2.5b) and *gap-overlap definition* (figure 2.5c).

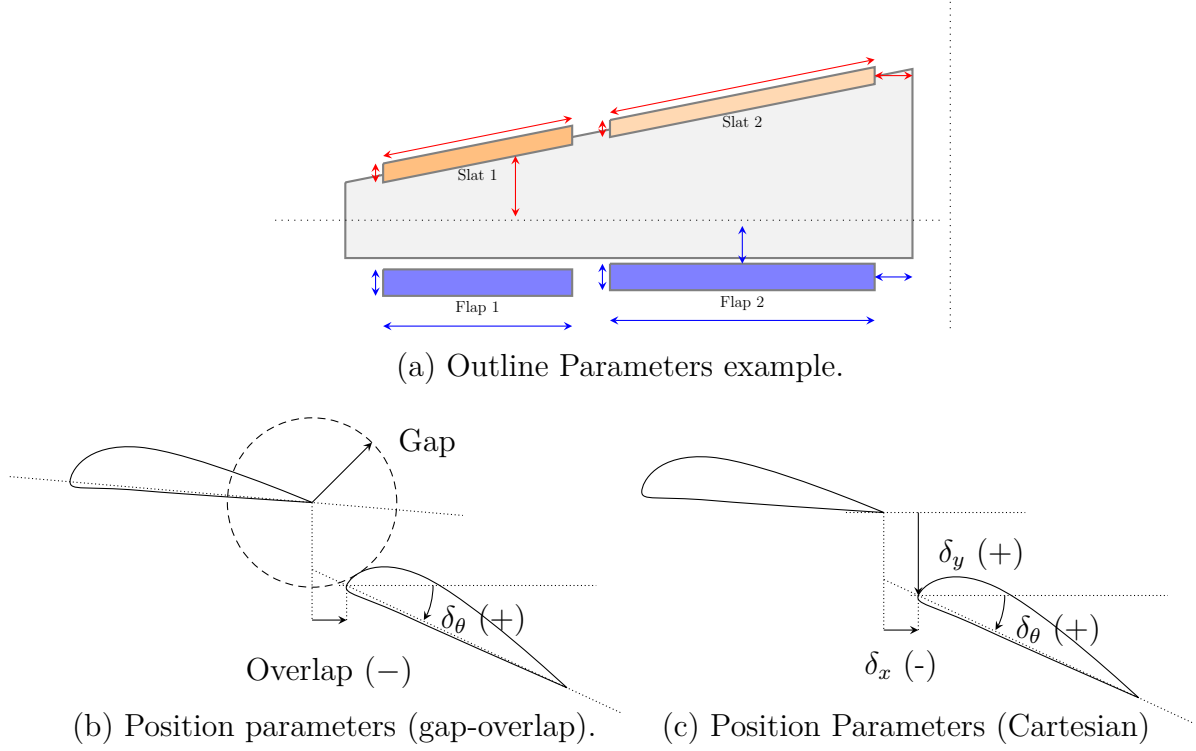


Figure 2.5: High-lift devices parametrisations

On the one hand, Cartesian coordinates represent the distance between the element trailing edge and the downstream element leading edge. By convention, the displacement is positive when the leading edge is under and fore the trailing edge (Giuseppe Trapani 2014). The third parameter is the element deflection angle relative to the main element clean chord, counted positive for clockwise rotation. This parametrisation is more convenient for mathematical modelling, such as in computer codes since the position is straightforwardly defined. A derivative used in optimisation is the use of relative position with respect to the initial position.

On the other hand, the gap-overlap parametrisation relates to the flow physics. Again three parameters are used: the *gap*, the *overlap* and as previously the *deflection angle*. The gap is the radius of a circle centred on the trailing edge of the preceding element and tangent to the following element. It is always positive. The overlap is the distance, along the stowed chord, between the upstream element trailing edge and the downstream leading edge. It is positive when elements are overlapping and negative otherwise. The deflection angle is defined as for the Cartesian parametrisation. The implementation in a computer code is no longer straightforward but requires some

transformation to be implemented; it is however preferred by experts due to the link between gap-lap values and underlying physics.

Shape Parameters Shape parameters define the actual shape of the devices. Although, the main wing shape is usually frozen during the early phase of the design, only the hidden part of the geometry during cruise flight is susceptible to be modified at the high-lift devices design stage.

The choice of the parametrisation has a direct influence on the result of the optimisation since it will restrict the available design space. The parametrisation also has an influence on the optimiser efficiency since for a similar design space coverage, a different number of defining parameters are needed.

Partial Differential Equation (PDE) is an efficient parametrisation that requires a small number of variables to describe complex 2D or 3D geometries. It is based on the resolution of an elliptic partial differential equation, and therefore has the drawback of having esoteric shape parameters.

Free Form Deformation (FFD) originates from computer graphics to deform models (Sederberg and Parry 1986), FFD is increasingly used in optimisation due to the high degree of freedom while being independent of the geometry representation. The geometry is fitted with a lattice of control points that, when moved, will affect all the points included inside the surface or the volume formed by the adjacent control points. Although the link between FFD variables and their effect on aerodynamics cannot be quantified, this approach is used in the preliminary work due to its easy implementation and freedom on new shapes.

Polynomial and spline are more intuitive parameters. Bézier and NURBS (Non-Uniform Rational B-Spline) are widely used in the aerodynamic community due to the reduced number of parameters to define 2D and simple 3D geometries (Giuseppe Trapani 2014). Unlike the FFD, the baseline geometry needs to be fitted, introducing an error in the initial shape.

Designers usually need to understand the optimisation path, and therefore need to link the design parameters to the flow features observed. Moreover, the adjoint method used to obtain cheaply the sensitivities requires analytical parametrisation methods. The most popular method in the aerodynamic community is the PAR-SEC (Sobieczky 1999) requiring 11 aerodynamic-related parameters to define an aerofoil. However, shapes are restricted and a method is needed to approximatively fit the parameters to the initial geometry.

This thesis aims at estimating the maximum lift after the shape has been frozen, and therefore the positions parameters are used. The method is also tested on a cruise shape for which the FFD shape parametrisation is used because of the large shape freedom and the absence of a fitting procedure that degrades the initial shape accuracy.

The challenge with high-lift devices design is the flow simulation around such a complex geometry. Many interactions and difficult-to-predict phenomena are present; the next section focuses on this aspect.

2.4 High-Lift Devices Description Through Physics

Figure 2.6 shows the main features encountered by a multielement aerofoil at high-lift conditions. Smith (1975) dressed a list of effects that appears in such cases. They are summarised below:

Slat effect The velocity induced by the slat, which can be considered as a vortex, is in the opposite direction than the one of the main element (downstream). This reduces the pressure suction peak which consequently reduces the pressure recovery demand and finally delays the separation of the main element allowing higher angle of attack.

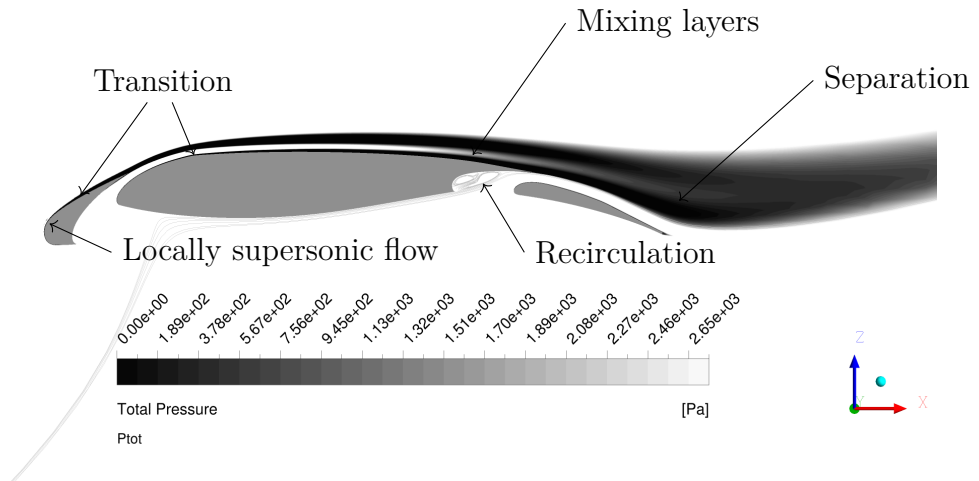


Figure 2.6: Flow features around a high-lift airfoil (GARTEUR airfoil) (J. Wild 2008)

Circulation and dumping effects The downstream element (flap usually) also alters the upstream flow on the trailing edge of the main element. This time the circulation induces an increase of the velocity at the trailing edge which also reduces the pressure recovery demand. Moreover, to satisfy the Kutta-condition, the circulation at the trailing edge must increase which provides an increase in lift. This effect is also called *flap effect* since it mainly affects the flap (Tooley et al. 2009).

Off-surface pressure recovery Usually the flow remains attached up to the elements trailing edge. However the boundary layer becomes quite large and, when leaving the trailing edge, forms a wake that develops behind the elements. For a well-designed multi-element aerofoil, the slots between elements allow the wakes to remain separate. The deceleration in the wake can lead to reverse flow which is detrimental for the lift. However, it has been shown that the off-surface recovery is more efficient than the one on a wall allowing more pressure recovery. Finally the wakes can merge with the existing downstream elements boundary layer or with the downstream wakes, known as *confluent boundary layer*. This effect is greatly affected by the *gap* parameter.

Fresh boundary layer Each element starts with a fresh boundary layer that consequently is thinner and can withstand greater adverse pressure gradients.

Transition As for any flow around an aerofoil, transition has a strong influence on lift and drag. The wake from the upstream element however can contaminate the boundary layer inducing early transition. 3D effects also have a strong impact on transition which make the analysis of 2D flow not sufficient on its own.

Recirculation The slat and the main element usually contain cove areas (to allow the retraction during cruise condition) that induces permanent recirculation in those areas (figure 2.6).

Transonic effects Despite being used at relatively low Mach number (around 0.20), the flow becomes transonic in high-acceleration regions such as slat and main element leading edges.

The gap-lap parametrisation gives a direct link between position parameters and effect on the flow, explained by Brune and McMasters (1990). Considering an inviscid

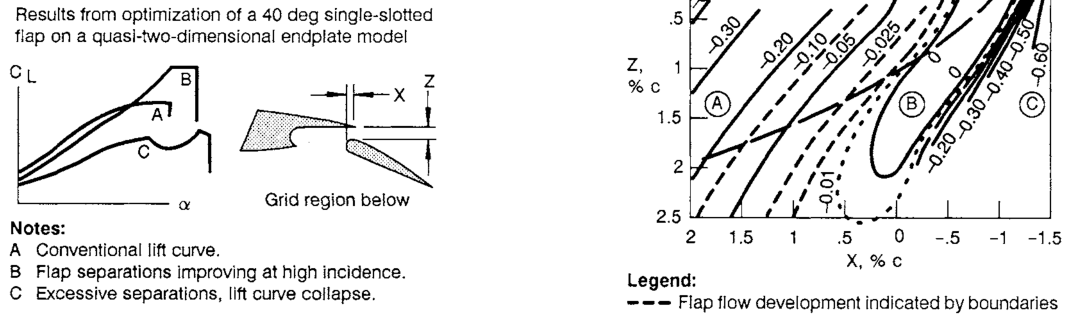


Figure 2.7: Effect of gap and lap on maximum lift on a two-dimensional aerofoil, from Brune and McMasters (1990).

flow, Foster (1972) showed that the wing lift decreases linearly as gap increases, but adding viscosity leads to an optimum position for maximum lift. Figure 2.7 shows such optimum location along with the change in lift with respect to gap and overlap. Three regions are identified, considering a small lap:

- A for small gap values, the main element wake interacts strongly with the flap boundary layer, reducing the flap suction peak, until a conventional separation;
- B for intermediate gap values, the interactions between wakes are reduced, provoking a separation at small angle of attack but reducing the flap suction peak as the main element wake strength increases which delays the flap separation: it leads to a higher maximum lift;
- C finally, for very large gap values, the interactions between wakes are too weak and the flap boundary layer cannot stay attached, provoking a large loss of lift.

2.5 From Design to Optimisation

2.5.1 Optimisation Design Process

The previous sections have shown the complex design process that is used for high-lift devices and the reasons for such intricacies. In order to apply optimisation techniques on such problems, the design process must be formalised as follow (J. Wild 2008):

1. Define the objective function(s) that describes the overall goal of the optimisation.

2. Define the parametrisation that will give enough freedom to resolve the problem in a constrained time.
3. Define the constraints to ensure a realistic solution: this should include structural, kinematic and aerodynamic requirements.
4. Define the aerodynamic solver (CFD or empirical) according to the level of fidelity required and the available time. This includes also the meshing strategy that has to take into account the parametrisation. The CFD needs also to be validated and verified from a baseline to ensure satisfactory resolution.
5. Define the objectives according to the needs. The selection is explained in section 2.2.
6. Validate and verify the final solution to confirm the superiority of the new geometry. This includes wind tunnel verification in industry but can also include CFD polar calculations.

2.5.2 Aerodynamic Specificities

The complex aerodynamic flow around high-lift devices creates multiple challenges in an optimisation context that need to be taken into account when selecting the optimisation strategy. The optimisation introduction is the subject of the next chapter; a quick summary on the specificities of solving optimisation problem on aerodynamics is shown below (Songqing Shan and G. G. Wang 2010a):

1. *non-smooth*: the underlying physical phenomenon is non-smooth due to the unsteady behaviour of separation, and due to the CFD solver iterative nature that produces noise leading to a non-smooth objective function,
2. contains *multiple local optima*: aerodynamics is highly non-linear and presents many local optima around of, and far from the true global optimum,
3. *expensive*: for Navier-Stokes based CFD, a single solution can take up to several hours to be obtained,
4. *black-box*: CFD software is usually used as a function without any knowledge (and no access) to its internal working,

5. used in *high-dimension* problems: geometry parametrisation requires many parameters which requires many input variables for the CFD process.

Hence, the optimisation strategy should respectively be (J. Wild [1998](#)):

1. robust to the noise,
2. able to leave local minima and find the global minimum,
3. efficient in terms of number of objective function calls,
4. independent of the CFD solver and should not require any internal modifications on the solver,
5. able to deal with increasing problem dimensionality despite an unavoidable performance cost.

Optimisation Applied to Computational Fluid Dynamics

The thesis is dedicated to the high-lift devices optimisation to predict the maximum lift of a particular aerofoil. The previous chapter highlighted the complex aerodynamics involved in the flow study around such a multielement aerofoil. The fluid dynamic equations describing a fluid behaviour are mostly derived from the Navier-Stokes equations, their exact solution is not known at this day: it is one of the seven most important problems in mathematics according to the Clay Mathematics Institute (Fefferman 2017). Hence it is clear that their numerical solution is a complex problem; the first section (3.1) of this chapter is therefore dedicated to the introduction to the equations and different approximations to provide a solution of various fidelity (and various computational cost).

The previous chapter also connected the multielement aerofoil flow specificities with the optimiser requirements to efficiently solve the optimisation problem. This is the subject of section 3.2. As efficient optimisers can be, the requirement of escaping local optima increases drastically the number of function calls. Hence, section 3.3 introduces surrogate modelling, allowing to use the real function information to produce a cheap-to-evaluate mathematical model.

3.1 Computational Fluid Dynamics

The estimation of the configurations aerodynamic performance is of primordial importance during the design stages. Numerical simulations must provide enough confidence at most of the angles of attack for making decisions before a sufficiently mature geometry is available to be evaluated experimentally. The flow properties were introduced in section 2.4 from chapter 2, and the main conclusion highlights the importance of viscosity in the flow around high-lift devices (Reckzeh 2003). Brune and McMasters (1990) showed that high-lift aerodynamic characteristics were the results of interactions between inviscid and viscous effects. Their prediction and coupling is therefore the object of this chapter in order to present the multiple methods developed throughout the years. First in section 3.1.1 the full Navier-Stokes (NS)

equations are introduced and challenges to numerically solve them are discussed. Simplifications to their modelling are introduced sequentially to form the Reynolds Averaged Navier-Stokes equations (section 3.1.2), Euler equations (section 3.1.3), potential flows (section 3.1.4), Boundary Layer Equations (section 3.1.5). The ESDU datasheets, that introduce further assumptions and mix equations with empirical data are also introduced in section 3.1.7. Although the high-lift devices design is a three-dimensional problem, this thesis focuses on two-dimensional methods since the viscosity complexity exists and makes the problem suitable for multifidelity optimisation without the computational cost overhead.

3.1.1 Navier-Stokes Equations

Navier-Stokes equations have been treated in many publications (A. Jameson 1989), and only a quick summary is presented here. The Navier-Stokes equations are coupled partial differential equations representing the conservation of mass, momentum and energy for a flow in a thermodynamic equilibrium. The mass (3.1), the three momentum (3.2) and the energy (3.3) equations are expressed as follows (using the summation convention):

$$\text{Mass} \quad \frac{\partial \rho_f}{\partial t} + \frac{\partial \rho_f u_j}{\partial x_j} = 0 \quad (3.1)$$

$$\text{Momentum} \quad \frac{\partial \rho_f u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho_f u_i u_j + p_r \delta_{ij} - \sigma_{ij}) = 0, \quad i = 1, \dots, 3 \quad (3.2)$$

$$\text{Energy} \quad \frac{\partial \rho_f E}{\partial t} + \frac{\partial}{\partial x_j} (\rho_f E u_j + p u_j + q_j - u_i \sigma_{ij}) = 0 \quad (3.3)$$

with σ_{ij} the viscous stress tensor defined in equation (3.4) and q_j the heat flux.

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2\mu}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad (3.4)$$

There are 3 momentum equations and with the mass and energy equations, only 5 equations are available for 6 dependent variables: pressure p_r , density ρ_f , temperature (in the energy equation, part of the total energy E) and the 3 components of the velocity u_i . The remaining variables are independent: the time t , and the Cartesian coordinates x_j . To close the system of equations, an additional equation is introduced

relating the pressure, density and temperature, called the state equation (3.5):

$$p_r = (\gamma - 1)\rho_f(E - 1/2u_iu_i), \quad (3.5)$$

with γ the ratio of specific heat.

The momentum equation relates the transfer of energy between convection (flow properties are transported by flow movement) and diffusion (properties transported by molecule movement) involving viscosity and the stress tensor. The later represents a computational challenge to be solved in a reasonable amount of time. For this reason, many assumptions are introduced in those equations to solve them with today's computers. Figure 3.1 shows the different assumptions that can be made and the methods name associated for the numerical resolution of Navier-Stokes equations. In particular, decomposing fluctuating variables into average and fluctuation reduces those difficulties and leads to the Reynolds Averaged Navier-Stokes equations (not shown in figure 3.1), presented in more details in the following section.

3.1.2 Reynolds Averaged Navier Stokes

Equations To be solved numerically, all the turbulence scales in the NS equations need to be resolved, requiring a very fine space discretisation and a small time step. For today's computers used in industry, the time needed to perform few time steps is beyond any practical reach. Large Eddy Simulations methods alleviates the CPU time required by modelling the smallest scales but still solving the largest ones. Still out of reach for most day to day aerodynamic simulations, they tend to be used mostly in research. To reduce further the CPU time required for the simulation, all scales can be modelled and not resolved from the equations. Reynolds Averaged Navier Stokes method decomposes fluctuating quantities ψ (the dependant variables: p_r , ρ_f , E , u_i) in a time averaged term $\bar{\psi}$ and its variation ψ' around its mean value, see equation (3.6):

$$\psi(\mathbf{x}, t) = \bar{\psi}(\mathbf{x}) + \psi'(\mathbf{x}, t), \quad (3.6)$$

with the RANS time averaged defined in equation (3.7):

$$\bar{\psi}(\mathbf{x}, t) = \int_{t_0}^{t_0+\Delta t} \psi(\mathbf{x}, t) dt \quad (3.7)$$

with Δt being large enough for the average to be independent of its value.

Substituting (3.6) in equation (3.2) yields to the appearance of the Reynolds stress

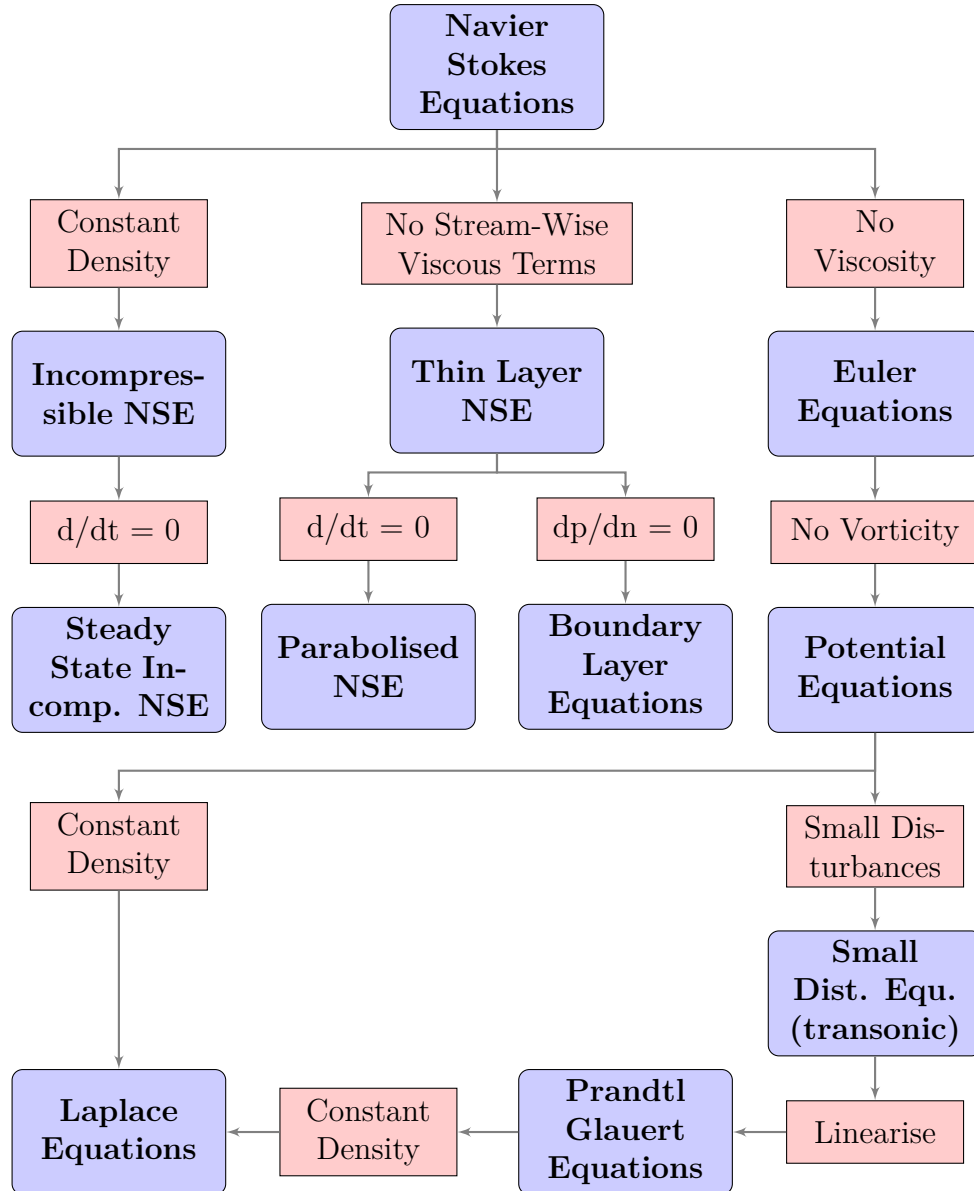


Figure 3.1: Navier-Stokes Equations assumptions for numerical simulations, adapted from A. Jameson (1989).

terms: $\rho_f \overline{u'_i u'_j}$ that cannot be determined from RANS equations, leading to more unknown variables than equations; this is known as the closure problem (Wilcox 1998) and is commonly solved by modelling the Reynolds stress term with the Boussinesq approximation that considers the Reynolds stress to be linearly dependant on the mean strain-rate tensor (plus a term to obtain the correct trace (Wilcox 1998)):

$$\rho_f \overline{u'_i u'_j} = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \left(\rho_f k + \mu_t \frac{\partial \bar{u}_k}{\partial x_k} \right) \delta_{ij}; \quad (3.8)$$

μ_t is the turbulent viscosity, and k the kinetic energy, solved by using additional transport and diffusion equations, as part of the different turbulence models presented below. The Boussinesq approximation assumes that the turbulent viscosity is isotropic, mostly true for external flows that does not involve large boundary layer separation.

Turbulence models Each CFD application has its own turbulence model, but for external aerodynamic transonic flow, one-equation Spallart-Allmaras (SA) (Spalart, Allmaras, et al. 1994) and two-equations k - ω (Chien 1982; Menter 1994; Wilcox 1988) turbulence models are the most common due to reduced number of extra equations to solve and their conservative (eg early prediction of stalling behaviour) nature beneficial for aircraft design. On the one hand, SA model ignores the kinetic energy in equation 3.8, and therefore uses a single transport equation to solve the turbulent viscosity. SA is used since it adds a single equation to the RANS equations which is simpler and cheaper (in the computational sense) to solve; it is also developed specifically for external transonic flows and can handle adverse pressure gradients with small separation although large separation are not accurately predicted. Finally, it has successfully been used in complex multielement flows by J. Wild (2008) and J. Wild and Brezillon (2005).

On the other hand, k - ω models have another unknown: the rate of dissipation thus it requires an additional transport equation, effectively increasing the computational requirements. The model handles adverse pressure gradients and improves the prediction of shock-induced separation. Despite being more suitable for separated flows (Bardina et al. 1997), experimental data is contradicting (J. Wild 2008).

The Drag Prediction Workshop (Mavriplis et al. 2009) highlighted the effect of the turbulence model parameters on the integrated aerodynamic coefficients, but the aerodynamic studies in this work are purely used as part of an optimisation process;

and as long as the flow-features are correctly captured, only the relative changes in the aerodynamic coefficients are of interest, not the absolute values. For all those reasons, the SA turbulence model is used in all RANS models throughout the thesis, always considered as a high-fidelity model.

Space discretisation Solving a physical (hence continuous) problem with a computer requires the discretisation of the physical space. As seen previously, flows involved in high-lift simulation are complex and the integrated lift and drag depend heavily on the good resolution of those phenomenon. The resolution and the cells quality impact the accuracy. In particular, in the elements wake and in the boundary layers, a fine mesh should be ensured to solve the complex viscous interactions. Two meshes types can be distinguished: *structured* for which every cells have the same number of neighbours, the connectivity being simply retrieved with indices, and *unstructured* for which vertices have arbitrary neighbours. Combining the two method is possible to create an *hybrid* mesh.

The simplest and fully automatic structured grids are *Cartesian* meshes (Melber-Wilkending et al. 2005). Despite the mesh generation easiness, this type of mesh is not suitable for viscous resolution due to a rapid increase in cells number to properly resolve boundary layer regions. Structured mesh overcome this issue with a very fine control on the element sizes. Moreover, because the cells location is easily retrieved instead of being stored as for unstructured grid, the memory requirements are low. The mesh qualities are usually better especially for the stream-wise resolution near the leading edge and the normal direction which is particularly convenient for boundary layer resolution. However, the connectivity and mesh quality comes at a price: the meshing process requires the creation of a topology, a very time consuming and manual process (Brezillon, R. Dwight, and J. Wild 2008). To finish with drawbacks, the more complex the geometry, the harder the mesh creation. The automation of such mesh strategy is therefore a complex problem that is solved in optimisation context by the use of a mesh deformation: an initial structured mesh is first manually created to ensure the best quality, and is then deformed to fit the geometry change. Although the mesh need not be recreated at each step, the deformation is limited to small movement to preserve the mesh quality. Diffusion is predominant in the wake regions, and is very dependant on the mesh quality. In an optimisation, the deformation may lead to change in the mesh quality in those regions, inconsistently

changing the drag estimation. Such mesh deformation approach is not suitable for high-lift aerofoil optimisation.

Newman III et al. (1999) states that triangles and tetrahedra are best suitable to fill the space in complex geometries. Moreover, as cells have not a fixed number of neighbours, their size can be varied independently, allowing to increase the number of cells only where needed. Complex geometries are easier to mesh, and automation easily implementable, especially in two dimensions (Melber-Wilkending et al. 2005). But despite this control on the element size, unstructured mesh generally requires more cells to reach the same level of accuracy. Cell orthogonality, very important for accuracy in high acceleration regions, is difficult to ensure when using tetrahedrons. This issue is overcome by using a prismatic boundary layer near the wall to have a good orthogonality and a higher cell concentration in the boundary layer region. This conserves the flexibility while allowing a good resolution of viscous effects. However, the generation of this type of mesh is usually time consuming and leads to a high number of cells, increasing the computation cost (Brezillon, R. Dwight, and J. Wild 2008). Finally, a structured mesh in simple geometric regions and an unstructured mesh in more complex regions can be merged together to form another type of hybrid mesh. This permits to speed up the RANS simulation while conserving the accuracy of structured meshes at the cost of increasing meshing complexity to connect the meshes.

A multielement aerofoil is a complex geometry: it has sharp edges where the elements are cut (slat and rear of the main element), changes in geometry curvatures in cove area, and elements are close to each other creating difficulties for the prismatic boundary layer. The aerofoil has a very strong curvature, and viscosity is not only constrained to the boundary layer but in all wake interaction regions. Thus the mesh should be fine in the vicinity of the aerofoils and in the large wake area. Finally, mesh deformation is ruled out requiring the automatic regeneration of the mesh at each optimisation iteration. Two test cases are used in chapters 5 and 6, both are using an unstructured mesh generation.

The previous chapter highlighted the aerodynamic phenomena appearing in multielement aerofoil flows: locally transonic flows where compressibility plays an important role, separation areas and wake interactions where viscosity is predominant. Those two components must be found in the resolution method used for both fidelities. RANS equations includes those features but are expensive to solve, requiring multiple Computational Power Units (CPUs) and several hours. The next sections

are dedicated to the presentation of alternative methods to solve NS equations with a smaller computational power requirement at the cost of reducing the generality of the method (more assumptions need to be fulfilled for the method to be applicable); the following methods will be considered in the rest of the thesis as low-fidelity methods.

3.1.3 Euler

Outside the boundary layer and the wake, viscosity can be ignored while conserving a good accuracy. Removing the viscous stress tensor and the heat conduction from equations (3.2) and (3.3) leads to the inviscid Euler equations (3.1), (3.9) and (3.10):

$$\text{Momentum} \quad \frac{\partial \rho_f u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho_f u_i u_j + p_r \delta_{ij}) = 0 \quad (3.9)$$

$$\text{Energy} \quad \frac{\partial \rho_f E}{\partial t} + \frac{\partial}{\partial x_j} (\rho_f E u_j + p_r u_j) = 0. \quad (3.10)$$

Euler equations are computationally cheaper to solve than the NS equations since the viscosity does not need to be solved nor modelled. A space discretisation is still required and its quality primordial to the accuracy of the solution but unlike RANS equations, the boundary layer, where viscosity is predominant, is not resolved. It is clear however that the solution is not directly applicable to the study of single- or multielement aerofoil flows, but only in regions where viscosity is not predominant: far from the walls and the wakes. Hence Euler equations need to be coupled with other set of equations to resolve viscosity and this is the subject of the section 3.1.6.

3.1.4 Potential Flow and Panel Method

Recall figure 3.1: we started from the top with the NS equations, added a non-viscosity assumption to derive the Euler equations. The next step is to assume an irrotational flow to obtain the potential flow equations. As the flow vorticity (the curl of the velocity vector) is assumed to be null, the velocity can be expressed as the gradient of a potential ϕ_f , see equation (3.11):

$$u_i = \frac{\partial \phi_f}{\partial x_i}, \quad (3.11)$$

and considering the compressibility to be negligible, using (3.11) in the mass (3.1) and momentum (3.2) equations yield to the Laplace equation (3.12) and Bernoulli

integral (3.13):

$$\text{Mass} \quad \frac{\partial^2 \phi_f}{\partial x_i^2} = 0 \quad (3.12)$$

$$\text{Momentum} \quad \frac{\partial \phi_f}{\partial t} + \frac{1}{2} u_i^2 + \frac{p_r}{\rho_f} = f(t), \quad (3.13)$$

with f a function of time. Introducing the potential ϕ_f reduces the number of unknown, and once the solution to the Laplace equation (3.12) is found, equation (3.13) determines the pressure everywhere in the flow.

The absence of vorticity is related to the absence of entropy production and therefore the use of a potential ϕ_f means the flow is isentropic (A. Jameson 1989). Aside from viscosity, entropy is created through shocks hence the potential methods fail to model shocks. Only weak shocks, satisfying the continuity equation but not the momentum one, are accurately modelled. Although a flow around a multielement aerofoil is expected to involve compressibility, its effect remains small.

The Laplace equation is linear allowing the flow to be decomposed in multiple elementary flows. This property has made this method successful to solve complex flow around two and three dimensional geometries (Tuncer Cebeci, Besnard, and H. H. Chen 1998; Mark Drela 1989; Le Balleur and Neron 1991), as long as the viscosity, close to the walls, are modelled with a different method and coupled to the potential flow solution. Despite the no-vorticity hypothesis made, the singularities include sources, doublets and vorticities. The geometry is discretised in panels, on each of them a distribution of singularities is assumed, and control points are introduced to apply a no-flow condition on the boundary to represent the geometry (Erickson 1990). The system of linear equations does not have a unique solution so additional boundary conditions are applied: the Kutta condition is imposed: the rear stagnation point should be on the sharp trailing edge (no flow around the trailing edge). The condition is enforced by the addition of panels in the aerofoils wake to model potential jump at the trailing edge. The wake position is not known a priori, and the panel positions must be manually imposed. This is an issue in flows around a multielement aerofoil as the wake affects the flow on downstream elements.

3.1.5 Boundary Layer Equations

When the Reynolds number is sufficiently large and the flow remains attached, the flow around a single- or multielement aerofoil can be decomposed in an inviscid part,

modelled by the Euler or potential equations for example, and a small viscous region restricted to the vicinity of the aerofoils. The size of this region is considered small compared to the scale of the aerofoil and the wake, and the Reynolds number is considered high, translating in conditions (3.14) and (3.15):

$$\delta_{bl} \ll c \quad (3.14)$$

$$Re \geq 1/\delta_{bl}^2 \quad (3.15)$$

with δ_{bl} the boundary layer thickness (part where the viscous effects are restricted to, defined as the distance normal to the wall where the streamwise speed $u = 0.99u_\infty$), c the aerofoil chord and Re the Reynolds number.

Using those conditions on the NS equations (3.1), (3.2) and (3.3), and restricting them to two dimensions, the boundary layer equations are derived and read as follow: (Anderson Jr 2010):

$$\text{Mass} \quad \frac{\partial \rho_f u_j}{\partial x_j} = 0 \quad (3.16)$$

$$\text{x-momentum} \quad \rho_f u_x \frac{\partial u_x}{\partial x} + \rho_f u_y \frac{\partial u_x}{\partial y} = -\frac{dp_e}{dx} + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) \quad (3.17)$$

$$\text{y-momentum} \quad \frac{\partial p}{\partial y} = 0. \quad (3.18)$$

The energy equation can be found in Anderson Jr (2010); the point here is that the stress tensor is gone, leading to a much simpler set of non-linear equations. Moreover, the pressure is constant in the direction normal to the wall, meaning that it is fully defined by the non-viscous flow outside the boundary layer. The equation does not have a closed solution, so it must be solved numerically, usually with the Runge-Kutta method.

3.1.6 Coupling Inviscid and BLE

The previous two sections presented a method to solve the inviscid part of a flow on an aerofoil and a method to solve the viscid part. The coupling can be performed in different manners affecting the solution robustness and accuracy (M. Drela 1986). The most robust algorithm is usually referred as fully-simultaneous, and consists in considering unknown variables from both the inviscid and viscid systems of equations, solved with the Newton method. The advantage of such resolution is twofold: the

coupling is very robust and the convergence rate rapid, marginally increasing the inviscid calculation cost. This approach is used in the MSES solver, described in the next paragraph. The effect of viscosity is to effectively displace the inviscid flow by the boundary layer thickness, which first slightly increases the lift produced by the aerofoil due to the increase in effective thickness but secondly decreases the camber leading to a larger decrease in lift. Hence, a more loosely coupling approach is to obtain the inviscid solution considering the displacement thickness as the new geometry, and use the obtained pressure as boundary conditions for the direct solving of the boundary layer equations. A new thickness is calculated and the inviscid flow needs to be recomputed. The coupling requires a high underrelaxation coefficient leading to a slow convergence and a high computational cost; separation is badly handled and results in strong divergence. This iterative method is referred as direct-coupling. A more common approach however, situated in between, is called semi-inverse coupling, proposed by Carter (1979). In this method, the boundary layer calculation is performed in inverse mode, the displacement thickness being prescribed. The obtained pressure is then compared with the inviscid solution and the difference is used to change the boundary layer displacement. The method is more robust when separation occurs and can predict accurately high-lift aerofoil maximum lift (Tuncer Cebeci, Besnard, and H. H. Chen 1998; Cebeci et al. 1986).

The work carried out in this thesis relies on two different implementations of a viscid/inviscid code suitable for multielement aerofoils near maximum lift: the MSES solver, from Professor Drela and the HiLi solver used with the sponsor.

MSES solver Professor Drela is the author of several world-famous CFD codes for two-dimensional flow around aerofoils. He started by the development of the ISES code during for his PhD thesis (M. Drela 1986). Initially thought to be applied to transonic flows, the code solves the steady Euler equations to obtain the inviscid and is coupled to a boundary formulation. The choice of the Euler equations is motivated by the wish of accurately modelling the shocks. The mesh is generated using streamlines to simplify the continuity and energy Euler equations resolution thanks to a reduction of unknown for each grid cells. Entropy diffusion is removed as the quantity is convected along the streamlines. The transonic shocks are handled by the addition of a bulk viscosity and a stable Newton solver for the coupling with the boundary layer equations. Following the ISES code, Drela worked on the MSES solver, for multielement aerofoils (Mark Drela 1990). The same philosophy is used

to solve the Euler equations, with streamlines that evolves during the resolution. Moreover, the use of such streamline-based grid allows to implicitly determine the wake positions during the Newton process, improving robustness compared to other grids; the author emphasis again the importance of the wake position on the downstream flow. The boundary conditions are imposed to fill the mass defect from the shocks and viscosity at the exit while taking into account the solution behaviour, allowing to reduce the domain size covered by the grid. Despite the grid evolving during convergence, an initial grid is generated prior to the iteration procedure using an incompressible panel method. The meshing procedure requires only some parameters to be set by the user to obtain the best grid quality. Manual tweaks may be needed when streamlines greatly vary during convergence leading to negative grid cell volumes. The integral boundary layer method is used with a multilayer velocity profile as it provides a fast and robust convergence when coupled with an inviscid solver. Initial tests performed by Prof. Drela show a good drag prediction and a slight over prediction of the maximum lift due to a failure to capture the slat and the main elements' wake merge.

HiLi High-Lift (HiLi) uses a two-dimensional panel method for the inviscid flow coupled to a integral boundary layer method. The coupling is performed with a semi-inverse procedure allowing prediction of moderate flow separation. Transition is empirically predicted and the compressibility effects are included via a Karman-Tsien correction. Finally, the cove areas need to be manually treated to avoid divergence.

Xfoil The method preliminary test is performed on a single element at low-speed. The code used is the famous Xfoil implementation, from Professor Drela (Mark Drela 1989). The inviscid flow is modelled by a panel method: the aerofoil and wake are discretised in panels, with a linear vorticity associated along with a constant source strength. The equations along with the Kutta condition is solved with a Gauss elimination. The boundary layer is modelled the same way as in MSES and the coupling procedure is also done with a Newton method. Xfoil can model low Reynolds flow and can predict transition from laminar to turbulent with a e^9 -type amplification formulation. More suitable for low-speed flow, compressibility is implemented with a Karman-Tsien correction.

3.1.7 ESDU Datasheets

Another approach to predict an aerofoil maximum lift is to use semi-empirical methods for which wind-tunnel experiments are augmented with algebraic equation to increase their range of applicability. The ESDU datasheet 94026 (1995b) presents the method to compute the maximum lift increment due to the high-lift devices deployment with the help of the ESDU datasheet 94027 (1994) and ESDU datasheet 94030 (1995a) using the aerofoils geometries, deployment parameters and flow conditions. Similarly, the ESDU datasheet 87005 (2008) and ESDU datasheet 87024 (2006) use the same information to evaluate the drag increment at maximum lift due to the deployment of the devices. The application of such a method is restricted to the type of aerofoils and flow conditions tested in the wind tunnel experiments. However it relies mainly on carpet plots and algebraic equations therefore the computational complexity is negligible which allows a quick estimation of a configuration performances.

3.2 Optimisation

3.2.1 Generalities

An optimisation is the research of the design variables that will give the smallest (or largest) response while satisfying the required constraints. Mathematically speaking, we will seek to solve the problem (3.19):

$$\min_{\mathbf{x} \in \mathcal{D}} \mathbf{f}(\mathbf{x}) \quad \text{Subject to} \quad \begin{cases} \mathbf{c}_i = \mathbf{0}, & i = 0, \dots, m' \\ \mathbf{c}_i \geq \mathbf{0}, & i = m' + 1, \dots, m \end{cases} \quad (3.19)$$

Where $\mathbf{f} : \mathcal{D} \rightarrow \mathbb{R}^l$ is the continuous objective function, $\mathcal{D}$ is a subset of $\mathbb{R}^n$ defining the boundaries of the problem, $\mathbf{x} \in \mathbb{R}^n$ is the vector of design variables and $\mathbf{c}_i$ is the set of equality and inequality constraints (Nocedal and Wright 1999). Generally speaking the objective function can return more than one component and this type of problem is called *multiobjective* optimisation. For single-objective and when the objective function f is scalar, the point $\mathbf{x}_1$ is a better solution to (3.19) than $\mathbf{x}_2$ if $f(\mathbf{x}_1) < f(\mathbf{x}_2)$. For the multiobjective counterpart, the concept of *Pareto optimality* needs to be used and is introduced in the next section.

Complex functions, such as aerodynamic-based functions, can present multiple

regions with minima leading to the definition of *local* and *global* minima. A local minimum $\mathbf{x}_{\text{loc}}^*$ is a point where the function is minimal in the vicinity of $\mathbf{x}_{\text{loc}}^*$: $\forall \mathbf{x} \in \mathcal{N}, f(\mathbf{x}_{\text{loc}}^*) \leq f(\mathbf{x})$, with $\mathcal{N}$ the neighbourhood of $\mathbf{x}_{\text{loc}}^*$. A global minimum is the point (or points) $\mathbf{x}_{\text{glob}}^*$ where the function has its minimal value over the whole design space: $\forall \mathbf{x} \in \mathcal{D}, f(\mathbf{x}_{\text{glob}}^*) \leq f(\mathbf{x})$. Classical deterministic optimisers tend to converge quickly to a local optimum whereas metaheuristic optimisers (defined in section 3.2.4) are designed to approximate the global optimum (Bianchi et al. 2009). Indeed, there is no proof that an optimiser will converge to the global optimum, one can only hope to approach it. Before choosing the optimisation algorithm, the problem must be defined: if an engineer wants to find some improvements on an already well-designed configuration, a local optimiser should be used but if an engineer starts from a generic design and wants to explore the entire design space to find the best solution(s), a global optimiser shall be selected. As described in chapter 2, section 2.1, those two concepts are encountered at two stages of the design process. This thesis aims at bringing the two distinct stages together, and therefore an approach aiming at global space exploration while ensuring a satisfactory accuracy for the optimum is sought.

3.2.2 MultiObjective

Often in aerodynamics, and many other optimisation applications, the problem cannot be reduced to a single objective because of objectives competing between each other, leading to the use of multiple objectives ($l > 1$), as demonstrated in chapter 2. One way to address this issue is the use of a weighted sum (or product) of the multiple objectives to create a single-objective problem f in equation (3.20):

$$f(\mathbf{x}) = \sum_{i=1}^{i=l} \alpha_i f_i(\mathbf{x}). \quad (3.20)$$

Weights need be introduced to specify how much a decrease in one of the objective is worth the increase of others. However, weights need to be chosen (manually or automatically) involving expert judgement and introducing bias in the optimisation problem (J. Wild and Brezillon 2005).

To keep both objectives during the optimisation and delay the choice of trade-off between them until solutions are obtained, multiobjective optimisers are required; most of them implement the concept of Pareto optimality stating that a solution is

optimal if it is not possible to find a solution that increases all objectives at the same time. It leads to the two following concepts:

Domination $\mathbf{x}_1$ dominates $\mathbf{x}_2$, if:

$$\forall i \in [1, \dots, l] \quad f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \text{ and } \exists k \in [1, \dots, l], f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2) \quad (3.21)$$

Pareto equivalence If some of the $\mathbf{f}(\mathbf{x}_1)$ components are less optimal than $\mathbf{f}(\mathbf{x}_2)$ components, and others are more optimal, the two points $\mathbf{x}_1$ and $\mathbf{x}_2$ are Pareto equivalent.

Therefore the use of the Pareto-optimality concept no longer leads to a single solution to the problem (3.19) but rather a set of solutions representing the different compromises between each objective. The solution selection is left to the decision maker a posteriori. The Pareto optimal set forms a curve when two objectives are considered, but it forms a surface for three objectives. It is clear that the more objectives are considered, the higher the number of points in the Pareto set and therefore the higher the computational cost of the optimisation.

There exist methods based on the weighted sum that leads to a Pareto front: the methods consist in varying the weighted parameters α_i to sweep the whole Pareto front (Hancock and Mattson 2013). First the *anchor points* are found by considering each objectives independently: $\alpha_k = 1, \alpha_{j \neq k} = 0$. The line in-between anchor points is discretised and multiple suboptimisation problems are solved from those points. Each suboptimisation problem is independent and can be run in parallel; they can use single-objective techniques but are limited to the Pareto front discovery in the vicinities of the starting points.

3.2.3 Local Methods

Two philosophies for the optimisation methods can be distinguished: metaheuristic methods, that may need randomness and do not converge to the exact optimum but to a good approximation of the global optimum, and local methods which, for most of them, a proof of convergence to the closest optimum exists. Those methods are deterministic and require most of the time the function derivatives. This section focuses on the last group, local methods, and introduces the line search and trust region strategies needed for the multifidelity chapter. The main optimisation algorithms are introduced as well.

Line search To solve the problem (3.19), the line search strategy chooses a search direction, function of the first function derivative such as the steepest descent algorithm or possibly of the second function derivatives—the Hessian—like the Newton method or an approximation of the Hessian such as SQCP. Then a one-dimensional subproblem is solved (usually approximatively) to find the step size that can yield a sufficient decrease of the objective function (Nocedal and Wright 1999) while ensuring stability.

Different methods exist to find the direction of search, requiring at least the first-order derivatives and sometimes second-order ones. As an example, the steepest descent uses the direction given by the first derivative. Despite being quite intuitive, it is shown to exhibit bad convergence properties on difficult problems.

If the second derivatives are known, they can be used to create the *Newton* direction. The method is derived from the Taylor expansion and uses the definite positive Hessian matrix to correct the downhill direction. Despite being very fast to converge, the need to have the Hessian matrix makes it prohibitive to be used with a computationally expensive function. To overcome this issue, *Quasi-Newton* search directions can be used. The Hessian is not required, and an approximation is used instead; this approximation is done using information from previous steps: $\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k$. The approximated Hessian is updated at each step according to some formulae, the most famous ones being the *symmetric-rank-one* and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms.

Trust region Conversely, trust region methods can be seen as first choosing a step size, called the trust region and then solving the subproblem to find a direction leading to an improvement. The information gathered from the function from previous iterations is used to construct an approximation model that has a similar local behaviour than the true function. The trust region is selected so that the subproblem is only solved (approximatively) in the accurate region of the model (Yuan 1999). The trust region size is usually changed according to the agreement between the model and the true function.

In the original trust region strategy, the approximation model m_k ¹ is quadratic and defined by:

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T \mathbf{B}_k p \quad (3.22)$$

where f_k is usually the function value at x_k , ∇f_k the gradient values at x_k and $\mathbf{B}_k$ can

¹For approximatively solving the subproblem.

be the Hessian matrix or its approximation. Finally, the usual trust region is defined as a ball defined by the Euclidean or any other norm from the centre, whose size is varying. The method was used for the gradient-based optimisation of a high-lift aerofoils by Richard P. Dwight and Brezillon (2006). They concluded that the Quasi-Newton Trust-Region (QNTR) algorithm outperforms the Conjugate Gradient (line search) in terms of CPU time, but QNTR is more sensitive to poor quality gradients.

The trust-region method being the starting point of the multifidelity optimiser used in this thesis, more details are provided in chapter 5.

Gradient calculation Calculating gradients is very challenging for aerodynamic problems (see all the points from chapter 2, section 2.5.2). One way of computing the gradient is by using a finite-difference approach: for each design variable, a perturbation is applied to the function. However it requires $n + 1$ function evaluations to obtain an approximation of the gradient. This becomes prohibitive for high-dimensional problems and is quite unreliable for non-smooth functions. A method that is gaining interest nowadays is using the solution of the adjoint problem (Kim, J. J. Alonso, and Antony Jameson 2004; Newman III et al. 1999; John C. Vassberg and Antony Jameson 2014). The cost to compute the gradient components for one objective is similar to the one needed to solve the flow field. Despite being very efficient for high-dimensional problems, the implementation is complex and is usually problem-dependant which is against point 4 from section 2.5.2. To obtain the adjoint equations, the flow solver needs to be differentiated, via hand-differentiation, complex variable or automatic differentiation (Richard P. Dwight and Brezillon 2006). Often, an approximation of the solver (first order discretization, thin-shear layer viscous fluxes, frozen turbulence model, etc.) is used for the differentiation due to the high complexity involved, degrading the gradients' accuracy.

Deterministic non-gradient-based method The Simplex algorithm (Rowan 1990) lies in the deterministic algorithm since it does not use any randomness. However, it has the particularity of not requiring derivatives information and can avoid being trapped in a local optimum. It is based on a *simplex* (a regular body) of $n + 1$ points. According to the vertices values, the shape is transformed (reflection, contraction, expansion) while moving in the design space until collapsing on a single point, the optimum. Not as efficient as gradient-based algorithms especially in high-dimensional

spaces, the Simplex algorithm is robust against noisy objective functions (J. Wild 2008). The simplex complexity rises linearly with the number of variable and becomes inefficient for more than 5 design variables (Rowan 1990). To overcome this issue, the Subplex (Rowan 1990) algorithm is preferred over the Simplex. The method uses the non-linear Simplex method on a subset of the dimensions. It also includes a restart strategy to overcome premature convergence. According to J. Wild (1998), the Subplex handles well the noise during optimisation and enables the Subplex optimiser to leave local optima. The Subplex implementation from the nlopt library (Johnson 2016) is integrated within the framework.

3.2.4 Metaheuristic Methods

The second philosophy regarding optimisation usually mimics physical, chemical, biological or even sociological behaviour. The use of randomness prevents the convergence provability but allows the exploration of the whole design space in order to find approximation of the global optimum; this comes also at the price of a relatively low efficiency. *Metaheuristic* combines *heuristics*, that searches the design space to find a *good* solution (but rarely the best) and are problem specific, in algorithms that are more generic. All those algorithms provide a different trade-off between exploration (also called diversification) and exploitation (also called intensification) (Bianchi et al. 2009). The following sections introduce metaheuristic algorithms that will be used throughout the thesis.

Evolutionary Algorithm Evolutionary Algorithm (EA) are inspired by the Darwinian evolution theory. An iteration of this algorithm can be seen as a *generation* to which certain operations are applied to generate the next generation. Those operations mimic natural selection: parents are used to create offspring through recombination. The design variables are similar to chromosomes (called *strings* here), and are modified mainly by *crossover* (combination of parents —selected according to their fitness— substrings to create offspring) and *mutation* (spontaneous modification of the strings). Randomness can appears in both operations.

Two main algorithm classes are found in the literature: *Genetic Algorithm* (GA) and *Evolutionary Strategies* (ES). The first one is described in the previous paragraph while the second one includes tuning parameters in the evolutionary process avoiding the need to tune the algorithm for each problem. If badly tuned, GA converges prematurely and requires an excessive number of evaluations. Despite their high

computational costs, EA are noise-resistant, more likely to find global optimum, and they are population-based which make them suitable for multiobjective optimisation and parallelism (K. Giannakoglou 2002).

The most famous implementation of such algorithm is the Nondominated Sorting Genetic Algorithm-II (NSGA-II) from Deb et al. (2002) that is usually used as a reference to be compared with. Each individual is sorted according to the domination level of the Pareto front they are belonging to, and are assigned a rank value. Distances between individuals are also computed, referred as crowding distance. The rank and crowding distance are then used for the parent selection. Those two properties push the solution to be Pareto optimal while preserving good diversity in the population for global exploration and to limit early convergence. NSGA-II was applied to the design of multielement aerofoil by Trapani, Kipouros, and Savill (2012) without reaching the MultiObjective Tabu Search performance, introduced below.

Tabu search The Multiobjective Tabu Search (MOTS) implementation (Glover and Laguna 1997; Jaeggi, G T Parks, et al. 2008) from Cambridge University is introduced. The algorithm uses a Hooke and Jeeves move for the local search (Hooke and Jeeves 1961). To keep the number of function evaluation low, at each iteration, a random subselection of all directions are tested, then compared and the best one, if allowed, is chosen for the next iteration. To avoid being trapped in a local optimum, MOTS uses *memories*:

short term memory records all the visited points. They become *Tabu*, meaning that already evaluated points cannot be revisited. This allows the optimiser to *climb away* from local optimum.

medium term memory records the best points found during the search. In multi-objective optimisation, those points are Pareto-equivalent and form the Pareto front. They are used to restart local search once no new optima are found; this move is called *intensification*.

long term memory records all the visited points in order to explore under-explored regions once no new optima are found; this move is called *diversification*.

MOTS presents a good compromise between global exploration and local search while being robust against noise. It is worth noting that MOTS does not require the function gradients which are hard to obtain for difficult aerodynamic problem.

Particle Swarm Optimiser Particle Swarm Optimisers (PSO) are based on birds behaviour in flocks (Eberhart and J. Kennedy 1995). As for GA, a random initial population is required to provide a global exploration. Each particle (individual) then evolves according to its past history (fitness function values) and the rest of the swarm (population) with the introduction of a global leader; the role of the global leader is to drive the swarm in the area of interest. The trade-off between cognitive and social behaviour is controlled when the particle velocities are updated which define their position evolution between iterations. Application to aerodynamic optimisation is done by Wei and Meijian (2014) and Jansen and Perez (2011).

The PyOpt library (Perez, Jansen, and Martins 2012) only provides a single-objective implementation of its Swarm optimiser. To use the strength of PSO within the author's multifidelity framework, a new implementation of the multi-objective PSO algorithm is done by the author and Spyridon Kontogiannis according to Alvarez-Benitez et al. (Alvarez-Benitez, Everson, and Fieldsend 2005) without constraint handling. The main algorithm is presented in algorithm 1 and the particle position and velocity update in algorithm 2. A succinct overview of the implementation is shown in figure B.1.

The optimisation starts with a Latin Hypercube Sampling (LHS) with n_{swarm} points to create the initial particle population $\mathbf{P}$. During the optimisation, a global leader is selected from the Pareto front PF with one of the following method:

prob the number of points dominated by each Pareto front point is counted and used as weights to select randomly the new global leader,

round the number of points dominated by each Pareto front point is counted and the point with the minimum value is used as the global leader,

maxdist prob the distances between each consecutive Pareto front point are computed and used as weights to select randomly the new global leader,

maxdist the distances between each consecutive Pareto front point are computed and the point with the maximum distance is used as the global leader,

random a uniformly random point is selected and used as the new global leader.

Then the position and velocity of each particle are updated according to algorithm 2 with the personal c_1 and social c_2 factors controlling the effect of the particle and the swarm best positions; the parameter w being the inertia, r_e the relaxation parameter

for the position update and ε_t the random turbulence parameter to avoid being stuck in a local optimum but remains set to zero in this study. Each particle best position is also updated accordingly and a new iteration can start.

Table 3.1 shows the results of performing a multiobjective optimisation on functions (3.23) and (3.24) with MOSPO using various leader selection techniques and swarm sizes. The performance is evaluated in terms of hypervolume indicator² value, introduced later in chapter 6, section 6.1.5, averaged for 10 runs and of standard deviation to evaluate the robustness of the random process. The results are also compared against the commercial MACROS tool, introduced in the next section; the latter is deterministic. The higher the value, the more optimal the Pareto front. Regarding the leader selection, the deterministic selection *round* is the one giving the worst results, with a higher variation. *Maxdist-prob* seems to give the best results, but when tested on a more complex function (chapter 6, section 6.4), important parts of the Pareto front are partially missed. Overall, the *prob* leader selection is selected and a study on the swarm size effect is performed on a similar function. The larger the swarm size, the more accurate and more consistent the Pareto front, but the more function evaluation is needed. As a compromise a size of 10 individuals is selected.

$$f_1(x_0, x_1) = \exp \left(\left(\frac{x_0 - 1}{\sqrt{2}} \right)^2 + \left(\frac{x_1 - 1}{\sqrt{2}} \right)^2 \right) \quad (3.23)$$

$$f_2(x_0, x_1) = \exp \left(\left(\frac{x_0 + 1}{\sqrt{2}} \right)^2 + \left(\frac{x_1 + 1}{\sqrt{2}} \right)^2 \right) \quad (3.24)$$

3.2.5 MACROS

MACROS v2.4 GTOpt (DATADVANCE 2016) is an optimisation library from Datadvantage that blends different optimisation algorithms through an automatic selection. MACROS optimisers (GTOpt) can be split into two main categories: classical gradient based optimisers for standard problems, and Surrogate Based Optimisers (SBO) for optimisation involving expensive functions. GTOpt SBO uses a surrogate to model the expensive function after an automatic sampling. Depending on a user-specified input, MACROS is alternating between global search, which involves maximum expected improvement criteria, and local search with a Quasi-Newton (Gill and

²The signification of the hypervolume indicators are introduced in section 6.1.5, but here we simply consider it as a measure of performance, the closer to 1 the better.

Algorithm 1 MultiObjective Particle Swarm Optimiser

```

 $\mathbf{X}_0 \leftarrow \text{LHS}(n_{\text{swarm}})$  {LHS sampling}
 $\mathbf{F}_0 \leftarrow \text{evaluate}(\mathbf{X}_0)$ 
 $\mathbf{V}_0 \leftarrow \mathbf{0}$ 
 $\mathbf{P} \leftarrow \text{create\_particle}(\mathbf{X}_0, \mathbf{F}_0, \mathbf{V}_0)$  {Create the swarm of particles}
for  $i = 0$  to  $\text{max\_iter}$  do
   $\mathbf{PF} \leftarrow \text{get\_pareto\_front}(\mathbf{P})$  {Obtain non-dominated points}
  for  $\mathbf{p}$  in  $\mathbf{P}$  do
     $\text{leader} \leftarrow \text{get\_leader}()$  {Obtain swarm leader}
     $\mathbf{p}.\text{update\_particle}(\text{leader})$  {See Algorithm 2}
     $\mathbf{p}.\text{evaluate}(\mathbf{p}.\mathbf{x})$  {Evaluate particle fitness value}
     $\mathbf{p}.\text{get\_best}(\mathbf{p})$ 
  end for
end for

```

Algorithm 2 Particle position and velocity update at iteration k

```

Require:  $\text{leader}_k$ 
 $r_1 \leftarrow \text{uniform}(0, 1)$ 
 $r_2 \leftarrow \text{uniform}(0, 1)$ 
 $\mathbf{v}_{k+1} = w\mathbf{v}_k + c_1r_1(\mathbf{p}.\text{leader}_k.\mathbf{x} - \mathbf{p}.\mathbf{x}_k) + c_2r_2(\text{leader}_k.\mathbf{x} - \mathbf{p}.\mathbf{x}_k)$ 
for  $i$  in  $[0, \dots, n-1]$  do {Limit velocities within bounds}
  if  $\mathbf{v}_{k+1}^{(i)} \geq \mathbf{v}_{\text{max}}^{(i)}$  then
     $\mathbf{v}_{k+1}^{(i)} = \mathbf{v}_{\text{max}}^{(i)}$ 
  else if  $\mathbf{v}_{k+1}^{(i)} \leq \mathbf{v}_{\text{min}}^{(i)}$  then
     $\mathbf{v}_{k+1}^{(i)} = \mathbf{v}_{\text{min}}^{(i)}$ 
  end if
end for
 $\mathbf{p}.\mathbf{x}_{k+1} = \mathbf{p}.\mathbf{x}_k + r_e\mathbf{v}_{k+1} + \varepsilon_t$ 
for  $i$  in  $[0, \dots, n-1]$  do {Limit positions within bounds}
  if  $\mathbf{x}_{k+1}^{(i)} \geq \mathbf{x}_{\text{max}}^{(i)}$  then
     $\mathbf{x}_{k+1}^{(i)} = \mathbf{x}_{\text{max}}^{(i)}$ 
  else if  $\mathbf{x}_{k+1}^{(i)} \leq \mathbf{x}_{\text{min}}^{(i)}$  then
     $\mathbf{x}_{k+1}^{(i)} = \mathbf{x}_{\text{min}}^{(i)}$ 
  end if
end for

```

Name	Swarm size	Value	Std
MACROS		0.529	
MACROS expensive		0.525	
MOPSO Round	10	0.487	0.055
MOPSO Maxdist-prob	10	0.537	0.007
MOPSO Maxdist	10	0.532	0.015
MOPSO Random	10	0.529	0.015
MOPSO Prob	5	0.452	0.087
MOPSO Prob	10	0.529	0.021
MOPSO Prob	20	0.546	0.003

Table 3.1: Hypervolume indicator values and standard deviation (Std) for different MOPSO parameters and comparison with MACROS.

Murray 1972) gradient-based optimiser; the local searches for SBO and for non-SBO as well are using a Quasi-Newton method.

The MACROS GTOpt philosophy in a multiobjective problem context is to “*avoid evaluation away from the Pareto front*”. It relies on two search types: a local search, using a Quasi-Newton method extended to the multiobjective concept, and a global search using the concept of *diffusion* along the Pareto front according to the following sequential algorithm (Pospelov and Gubarev 2014):

1. Anchor search: each objective is first optimised independently to find the anchor points. The search is performed as an iterative process to ensure not to find only weak Pareto optimal points (if objective $f^{(i)}$ is optimised, some other points could lead, for the same $f^{(i)}$, to smaller $f^{(j)}, j \neq i$, see figure 3.2).
2. Diffusion: from a Pareto point, a small shift is performed in the direction of the Pareto set tangent plane (in the design space) as shown in figure 3.2. A local search is then performed to find the Pareto-optimal point from the shifted point (in blue in figure 3.2).

The final number of point on the Pareto front is an input from the user; MACROS then deduces a step size for the shift (ε in figure 3.2) according to the user input and to the anchors extent.

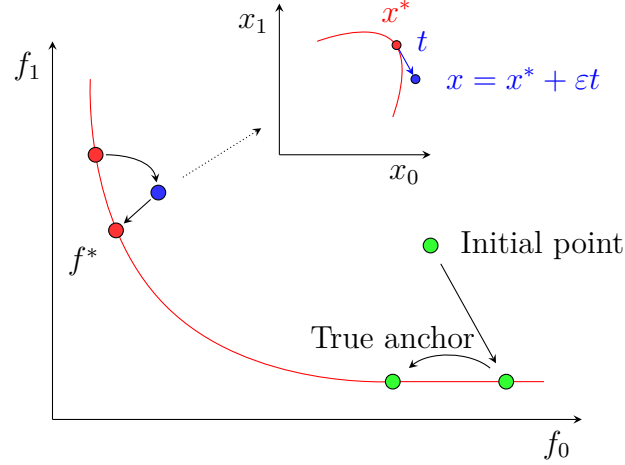


Figure 3.2: MACROS GTOpt anchors search

3.3 Surrogate Modelling

Despite the numerous methods to obtain an approximate solution to the Navier-Stokes equations, the low-fidelity methods cannot solely be used because of the differences in confidence between results. Nowadays the industry relies on the RANS solutions for the multielement aerofoils design, but their high computational cost associated makes it intractable inside an optimisation loop. To alleviate this cost, a mathematical model using sparse points of the true simulation can replace the expensive model. The points are usually selected via a sampling process, presented in section 3.3.1. Various models are found in the literature but can generally be categorised into parametric models, assuming a fixed form of the underlying function, and non-parametric models without prior knowledge of the function. First parametric models are quickly introduced in section 3.3.2 and the focus is made on non-parametric models with the Radial Basis Functions in 3.3.3, Kriging models in section 3.3.4 and co-Kriging models in section 3.3.5. Their implementation is described through a simplified Unified Modelling Language (UML) diagram in figure B.2.

3.3.1 Sampling Methods

To construct a metamodel, information from the true function is required. This is done by sampling the function to be approximated, with what is called a Design

of Experiments (DoE), at smartly chosen points so that the amount of information gathered is maximised. Since the overall objective is to obtain a good approximation (in terms of accuracy and space exploration) at a fraction of the original cost, the sampling needs to be reasonably used (Ruichen Jin, Wei Chen, and Sudjianto 2002).

Typical methods include *full factorial* (Forrester, Sóbester, and Andy J. Keane 2008), *Latin hypercube* (Lian, Oyama, and Liou 2010) which is the most widely used thanks to its good behaviour in high dimension spaces, and more advanced techniques based on optimality (Forrester, Sóbester, and Andy J. Keane 2008; Ye 1998). Full factorial technique splits each dimension in a fixed number of bins, and the full grid is used as sampling points. It provides the best uniformity but the number of points increases sharply with the space dimension. For that reason, when a design of experiments is required in this thesis to initialise a population or to build a complete surrogate, the Latin Hypercube Sampling technique is used. Instead of using the complete grid, for each bin, one point per dimension is selected, so that the projection on each axis is uniform. This work uses the pyDoE implementation (Baudin et al. 2017), and specifically the maximin criterion that maximises the minimum distance between points to select the best LHS sampling.

3.3.2 Parametric Models

Parametric models assume a fixed form of the underlying function reducing the required number of points to train the model. Polynomial models (linear, quadratic or higher orders) are the most used ones; they are computationally inexpensive but limited in terms of flexibility. Therefore, those models are usually preferred for local approximation or when the underlying function form is known a priori (Songqing Shan and G. G. Wang 2010b).

A polynomial model of order k for the scalar function $f : \mathbb{R} \rightarrow \mathbb{R}$ can be expressed as:

$$\hat{f}(x) = w_0 + w_1x + w_2x^2 + \dots + w_kx^k = \sum_{i=0}^k w_ix^i. \quad (3.25)$$

The model training requires the estimation of the $\mathbf{w} = \{w_0, w_1, \dots, w_k\}^T$ with a least square method. It is important to note that this is a non-interpolating method thus estimate is not exact at interpolation points.

This method can be generalised for multivariate functions. Let us consider $f :$

$\mathbb{R}^n \rightarrow \mathbb{R}$. The second-order expression of $\hat{f}$ becomes:

$$\hat{f}(\mathbf{x}) = w_0 + \sum_{i=0}^n w_i x_i + \sum_{j=0}^n w_{jj} x_j^2 + \sum_{i=0}^n \sum_{j=0}^n w_{ij} x_i x_j. \quad (3.26)$$

The importance of each variable can be estimated (screening) by examining the coefficients (R. Jin, W. Chen, and Simpson 2001). Also, when using relatively low-order polynomials, the noise can be eliminated. On the contrary, using high order polynomials may result in an overfitted model: the model predicts correctly the points used for the training but fails to predict independent points (Forrester, Sóbester, and Andy J. Keane 2008). Finally, as the order increases, the number of parameters to determine increases also largely; this prevents the use of polynomial regression for high-dimension problems.

More flexible, non-parametric models do not assume any general shape of the function to model. Requiring more internal parameters to determine as part of the training, they are usually more complex to train but very promising for the prediction of unknown non-linear function. The following sections introduce the models used in the rest of the thesis.

3.3.3 Radial Basis Functions

To approximate an one-output function with no prior information, a linear combination of basis functions is used so that the expression of the model prediction $\hat{y}$ is:

$$\hat{y}(\mathbf{x}) = \sum_{k=0}^{p-1} \lambda_k \phi(\|\mathbf{x} - \mathbf{d}_k\|) = \boldsymbol{\lambda} \boldsymbol{\Phi}. \quad (3.27)$$

The training involves the estimation of the vector $\boldsymbol{\lambda}$ so that the combination is exact at the p interpolation points $\{\mathbf{d}_0, \dots, \mathbf{d}_{p-1}\}$:

$$y(\mathbf{d}_i) = y^{(i)} = \sum_{k=0}^{p-1} \lambda_k \phi(\|\mathbf{d}_i - \mathbf{d}_k\|), \quad i \in [0, \dots, p-1] \quad (3.28)$$

which can also be written as a linear system in (3.29) and since $\boldsymbol{\Phi}$, called the *Gram matrix*, is positive definite (Forrester, Sóbester, and Andy J. Keane 2008), the inverse can be calculated:

$$\mathbf{Y} = \boldsymbol{\lambda} \boldsymbol{\Phi} \quad \Leftrightarrow \quad \boldsymbol{\lambda} = \mathbf{Y} \boldsymbol{\Phi}^{-1}, \quad (3.29)$$

with $\mathbf{Y} = [y^{(0)}, \dots, y^{(p-1)}]$, $\Phi = \{\phi_{i,j} \mid i, j \in [0, \dots, p-1]^2\}$, $\phi_{i,j} = \phi(\|\mathbf{d}_i - \mathbf{d}_j\|)$ and $\boldsymbol{\lambda} = [\lambda_0, \dots, \lambda_{p-1}]$.

Radial basis function values depend only on the norm from the origin; the Euclidean norm is consistently used in this work. However the introduction of the same norm for each variable may be unsuitable if the variables are representing quantities of different magnitude. Radial Basis Function does not address the scaling issue since the same parameter controlling the norm scaling applies to all directions. It is therefore expected that the model will not perform consistently in each direction (each variable); but the negligible training time remains a strong incentive to initially work with RBFs.

Various basis functions can be found in the literature such as the linear, Matern and Gaussian. The linear basis function ($\phi(\|\mathbf{x}\|) = \|\mathbf{x}\|$) diverges to infinity for high distances which is not suitable for a correction. The Matern and the Gaussian basis functions showed to provide similar robustness in approximation. In Wild's RBF implementation (S. M. Wild, Regis, and Shoemaker 2008) a linear tail is added to prevent the Gram matrix to be ill-conditioned but Gutmann (2001) shows that the Gaussian RBF does not require such tail. The Gaussian basis function without linear tail is therefore used (equation (3.30)), with the correlation factor ϵ_c set equal to the mean distance between points and is computed during the training.

$$\phi(\|\mathbf{x}\|) = e^{-\frac{\|\mathbf{x}\|}{\epsilon_c}} \quad (3.30)$$

The RBF method can also be considered as an artificial neural network: its form is similar to a single-layer neural network, with an input layer, a single hidden layer and an output layer with a single unit (Haykin 1999). Typical neural networks require many parameters to be optimised which necessitates more training points than RBF.

3.3.4 Kriging

The original Kriging model considers that the unknown function f to approximate is the realisation of a Gaussian process, even though the computer code is deterministic (Forrester, Sóbester, and Andy J. Keane 2008). It is assumed that, at the p sampling points $[\mathbf{d}_1, \dots, \mathbf{d}_p]$, the function values $\{\mathbf{y}_i = f(\mathbf{d}_i), i \in [1, \dots, p]\}$ are correlated with each other according to the Gaussian correlation function (Forrester,

Sóbester, and Andy J. Keane 2008; Sacks et al. 1989; Walter 2014):

$$\text{cor}[\mathbf{y}_i, \mathbf{y}_j] = \exp \left(- \sum_{k=1}^{n-1} \xi^{(k)} |d_i^{(k)} - d_j^{(k)}|^{q_k} \right), \quad i, j \in [1, \dots, p] \quad (3.31)$$

with ξ_j and q_j the Kriging hyperparameters in the j^{th} direction. They are estimated by maximising the concentrated ln-likelihood (Jones 2001) of y , as in equation (3.32):

$$\max_{\xi, q} \ln(L). \quad (3.32)$$

The concentrated ln-likelihood $\ln(L)$ is defined in equation (3.33):

$$\ln(L) \approx -\frac{n}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} \ln |\Psi| \quad (3.33)$$

with Ψ the correlation matrix of the sampling data defined in equation (3.34) and $\hat{\sigma}$ the maximum likelihood estimate of the variance defined in equation (3.35) where $\mu = \mathbf{1}\mu$ is the vector containing the mean of the random outputs $\mathbf{y}$ on all its rows.

$$\Psi = (\text{cor}[\mathbf{y}_i, \mathbf{y}_j])_{1 \leq i \leq p, 1 \leq j \leq p} \quad (3.34)$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mu)^\top \Psi^{-1} (\mathbf{y} - \mu)}{p} \quad (3.35)$$

Similarly, the maximum likelihood estimate for the mean $\hat{\mu}$ can be expressed by equation (3.36):

$$\hat{\mu} = \frac{\mathbf{1}^\top \Psi^{-1} \mathbf{y}}{\mathbf{1}^\top \Psi^{-1} \mathbf{1}}. \quad (3.36)$$

The maximisation of (3.32) is usually a difficult problem since the ln-likelihood is most likely multimodal (Rasmussen 2006; Sacks et al. 1989). Heuristic optimisation algorithms are preferred to solve such problems (Forrester, Sóbester, and Andy J. Keane 2008); my implementation uses the Augmented Lagrangian Particle Swarm Optimizer (ALPSO) algorithm (Jansen and Perez 2011) from the PyOpt library (Perez, Jansen, and Martins 2012).

Finally, to calculate the prediction from the model at the point $\mathbf{x}$, the maximum likelihood estimate $\hat{y}$ expression (Forrester, Sóbester, and Andy J. Keane 2008) is used, defined in equation (3.37):

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r} \Psi^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (3.37)$$

with $\mathbf{r}$ is the correlation vector between the new prediction and the sampling point as shown in (3.38):

$$\mathbf{r} = (\text{cor}[f(\mathbf{d}_i), f(\mathbf{x})])_{1 \leq i \leq p}. \quad (3.38)$$

3.3.5 Co-Kriging

The co-Kriging method is an extension of the Kriging interpolation method but correlates different sets of data (M. C. Kennedy and O'Hagan 2001) associated with different confidence levels: the low- and high-fidelity functions f_{low} and f_{high} outputs can be correlated to improve the prediction. We use the *auto-regressive* model, assuming that the high-fidelity model does not have any error embedded and only the low-fidelity model has inaccuracies (Forrester, Sóbester, and Andy J. Keane 2007, 2008; M. C. Kennedy and O'Hagan 2001): no information can be extracted from the low-fidelity function when the value of the high-fidelity is known.

As for ordinary Kriging interpolation (Anand Amrit et al. 2016; Cressie 1992; Demange, A. Mark Savill, and Timoleon Kipouros 2017b; Lam 1983; O'Hagan 2006), the unknown one-output (the error surrogate model $\mathbf{e}_k$ assumes that the outputs are independent so that one surrogate for each objective can be constructed at the same time) functions $f_{\text{low}} : \mathbb{R}^n \mapsto \mathbb{R}$ and $f_{\text{high}} : \mathbb{R}^n \mapsto \mathbb{R}$ to approximate are seen as Gaussian processes (respectively G_{low} and G_{high}), even though the computer codes are deterministic (Forrester, Sóbester, and Andy J. Keane 2008). Let's define the p_h sampling points $\mathbf{D}_{\text{high}} = [\mathbf{d}_{h,1}, \dots, \mathbf{d}_{h,p_h}]$ having the high-fidelity functions values:

$$\mathbf{y}_{\text{high}} = \{y_{\text{high},i} = f_{\text{high}}(\mathbf{d}_{h,i}), i \in [1, \dots, p_h]\} = f_{\text{high}}(\mathbf{D}_{\text{high}}) \quad (3.39)$$

and the p_l sampling points $\mathbf{D}_{\text{low}} = [\mathbf{d}_{l,1}, \dots, \mathbf{d}_{l,p_l}]$ having the low-fidelity function values:

$$\mathbf{y}_{\text{low}} = \{y_{\text{low},i} = f_{\text{low}}(\mathbf{d}_{l,i}), i \in [1, \dots, p_l]\} = f_{\text{low}}(\mathbf{D}_{\text{low}}). \quad (3.40)$$

All the high-fidelity points are sampled with the low-fidelity function such that $\mathbf{D}_{\text{high}} \subset \mathbf{D}_{\text{low}}$; the two output vectors are concatenated to a single one: $\mathbf{y} = [\mathbf{y}_{\text{low}}^\top \mathbf{y}_{\text{high}}^\top]^\top$. The high-fidelity function is approximated by a linear combination of the low-fidelity Gaussian process and a second one representing the difference as shown in equation (3.41):

$$G_{\text{high}} = \eta G_{\text{low}} + G_{\text{diff}}. \quad (3.41)$$

According to Forrester, Sóbester, and Andy J. Keane (2007), the expression of the

Gaussian processes leads to the construction of four covariance matrices —(3.42), (3.43), (3.44) and (3.45)— expressing the covariance of the low-fidelity data between itself and the high-fidelity one, and of the high-fidelity data between itself and the low-fidelity one.

$$\text{COV}(\mathbf{y}_{\text{low}}, \mathbf{y}_{\text{low}}) = \sigma_l^2 \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{low}}) \quad (3.42)$$

$$\text{COV}(\mathbf{y}_{\text{low}}, \mathbf{y}_{\text{high}}) = \eta \sigma_l^2 \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{high}}) \quad (3.43)$$

$$\text{COV}(\mathbf{y}_{\text{high}}, \mathbf{y}_{\text{low}}) = \eta \sigma_l^2 \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{high}})^\top \quad (3.44)$$

$$\text{COV}(\mathbf{y}_{\text{high}}, \mathbf{y}_{\text{high}}) = \eta^2 \sigma_l^2 \Psi_l(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}}) + \sigma_d^2 \Psi_d(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}}), \quad (3.45)$$

It results in the following square covariance matrix $\mathbf{C}$ of dimension $(p_l + p_p) \times (n_l + n_p)$:

$$\mathbf{C} = \begin{bmatrix} \sigma_l^2 \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{low}}) & \eta \sigma_l^2 \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{high}})^\top \\ \eta \sigma_l^2 \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{high}}) & \eta^2 \sigma_l^2 \Psi_l(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}}) + \sigma_d^2 \Psi_d(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}}) \end{bmatrix}. \quad (3.46)$$

The correlation matrices $\Psi_z(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{low}})$, $z \in [l, d]$ is defined in equation (3.47):

$$\begin{aligned} \Psi_z(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{low}}) &= (\text{cor}_z[\mathbf{y}_{\text{low}, i}, \mathbf{y}_{\text{high}, j}])_{1 \leq i \leq p_h, 1 \leq j \leq p_l} \\ &= \left(\exp \left(- \sum_{k=1}^{n-1} \xi_z^{(k)} |d_{h,i}^{(k)} - d_{l,j}^{(k)}|^{q_{z,k}} \right) \right)_{1 \leq i \leq p_h, 1 \leq j \leq p_l}. \end{aligned} \quad (3.47)$$

Similarly to the Kriging method, the hyperparameters ξ_l, q_l for the low-fidelity model and ξ_d, q_d, η for the difference model are determined by maximising the concentrated likelihood. Since the low-fidelity model is assumed independent of the high-fidelity one, the low-fidelity hyperparameters are found by maximizing (3.48) (Forrester, Sóbester, and Andy J. Keane 2007):

$$\ln(L_l) \approx -\frac{p_l}{2} \ln(\hat{\sigma}_l^2) - \frac{1}{2} \ln |\Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{low}})|; \quad (3.48)$$

with the maximum likelihood estimates for the variance $\hat{\sigma}_l^2$ and for the mean $\hat{\mu}_l$ being expressed by equations (3.49) and (3.50):

$$\hat{\sigma}_l^2 = \frac{(\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)^\top \cdot \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{low}})^{-1} \cdot (\mathbf{y}_l - \mathbf{1}\hat{\mu}_l)}{p_l}, \quad (3.49)$$

$$\hat{\mu}_l = \frac{\mathbf{1}^\top \cdot \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{low}})^{-1} \cdot \mathbf{y}_l}{\mathbf{1}^\top \cdot \Psi_l(\mathbf{D}_{\text{low}}, \mathbf{D}_{\text{low}})^{-1} \cdot \mathbf{1}}. \quad (3.50)$$

The high-fidelity parameters $\boldsymbol{\xi}_d, \mathbf{q}_d, \eta$ are then determined by maximising the difference model likelihood function (Forrester, Sóbester, and Andy J. Keane 2007):

$$\ln(L_d) \approx -\frac{p_h}{2} \ln(\hat{\sigma}_d^2) - \frac{1}{2} \ln |\boldsymbol{\Psi}_d(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}})|. \quad (3.51)$$

This time, the maximum likelihood estimates for the difference model variance $\hat{\sigma}_d^2$ and mean $\hat{\mu}_d$ is expressed by (3.52) and (3.53) respectively:

$$\hat{\sigma}_d^2 = \frac{(\boldsymbol{\Delta} - \mathbf{1}\hat{\mu}_d)^\top \cdot \boldsymbol{\Psi}_d(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}})^{-1} \cdot (\boldsymbol{\Delta} - \mathbf{1}\hat{\mu}_d)}{p_h}, \quad (3.52)$$

$$\hat{\mu}_d = \frac{\mathbf{1}^\top \cdot \boldsymbol{\Psi}_d(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}})^{-1} \cdot \boldsymbol{\Delta}}{\mathbf{1}^\top \cdot \boldsymbol{\Psi}_d(\mathbf{D}_{\text{high}}, \mathbf{D}_{\text{high}})^{-1} \cdot \mathbf{1}}. \quad (3.53)$$

with the vector of differences $\boldsymbol{\Delta}$ defined by equation (3.54) at the high-fidelity sampling points:

$$\boldsymbol{\Delta} = \mathbf{y}_{\text{high}} - \rho f_{\text{low}}(\mathbf{D}_{\text{high}}). \quad (3.54)$$

The same method as in the previous section is used for the maximisation of (3.48) and (3.51). Despite the matrix $\mathbf{C}$ not appearing in equation (3.51), its quality is of primal importance to compute the new points. Therefore, the matrix numerical quality is checked during the maximisation, and the hyperparameters leading to bad matrices are discarded.

Once the more likely parameters are found with the successive likelihood maximisations, the estimated prediction mean $\hat{\mu}$ is evaluated with equation (3.55):

$$\hat{\mu} = \frac{\mathbf{1}^\top \cdot \mathbf{C}^{-1} \cdot \mathbf{y}}{\mathbf{1}^\top \cdot \mathbf{C}^{-1} \cdot \mathbf{1}}. \quad (3.55)$$

Finally, to calculate the prediction from the model at the point $\mathbf{x}^*$, the maximum likelihood estimate $\hat{y}$ expression is used, defined in equation (3.56) (Forrester, Sóbester, and Andy J. Keane 2008):

$$\hat{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}(\mathbf{x}^*)^\top \cdot \mathbf{C}^{-1} \cdot (\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (3.56)$$

with $\mathbf{r}$ is the correlation vector between the already known outputs (our random variables used to construct the model) and the new prediction considered as a random

variable having $G_{\text{high}}(\mathbf{x}^*) = y^*$ as output:

$$\mathbf{r}(\mathbf{x}^*) = \begin{pmatrix} \hat{\rho}\hat{\sigma}_l^2\text{cor}_l[\mathbf{y}_l, y^*] \\ \hat{\rho}^2\hat{\sigma}_l^2\text{cor}_l[\mathbf{y}_h, y^*] + \hat{\sigma}_h^2\text{cor}_h[\mathbf{y}_h, y^*] \end{pmatrix}. \quad (3.57)$$

The prediction from the co-Kriging is used to estimate the corrected function for each output $f_{\text{corr}}(\mathbf{x}^*) = \hat{y}(\mathbf{x}^*)$ such that: $\mathbf{f}_{\text{corr}} = [\hat{y}^{(1)}, \dots, \hat{y}^{(l)}]^\top$.

The co-Kriging method provides as well an estimation of the error from the sampling at the point $\mathbf{x}^*$ with the mean squared error $\hat{s}^2(\mathbf{x}^*)$, defined (Forrester, Sóbester, and Andy J. Keane 2008) in equation (3.58):

$$\hat{s}^2(\mathbf{x}^*) \approx \rho^2\hat{\sigma}_l^2 + \hat{\sigma}_h^2 - \mathbf{r}(\mathbf{x}^*)^\top \cdot \mathbf{C}^{-1} \cdot \mathbf{r}(\mathbf{x}^*) + \frac{1 - \mathbf{1}^\top \cdot \mathbf{C}^{-1} \cdot \mathbf{r}(\mathbf{x}^*)}{\mathbf{1} \cdot \mathbf{C}^{-1} \cdot \mathbf{1}} \quad (3.58)$$

Once the model is trained, no information relative to the prediction reliability is available. How could we be sure that the parameters we chose give a good approximation? Since all available points are used for the training, the prediction capabilities need to be estimated via a cross-validation (Jaeggi, Geoffrey T. Parks, et al. 2008). In the p -fold cross validation, the training set of points is divided into p subsets. Then, leaving each time a different subset, a model is fitted to the remaining $p - 1$ subsets. Predictions at the excluded points are then compared to the true training points with any error metric. This is repeated for each subset. If all subsets of size k are tested, this is called the *leave-k-out* but is more computationally expensive than the p -fold method. For RBF, G. G. Wang and S. Shan (2006) showed that $k = 1$ provides a good error prediction. For the Kriging method, a value taking into account the size of the training set is preferred (eg. $k = \sqrt{N}$ or $k = 0.1N$). However, the leave-one-out method should be considered with care since rather than estimating the absolute accuracy of the model, the method evaluates the sensitivity of the loss of a training point.

3.3.6 Curse of Dimensionality

As the problem dimension increases, the computational cost also rises. This is known as the *curse of dimensionality* (Songqing Shan and G. G. Wang 2010b). Also called the *problem of size* by Lewis (1996), this issue appears on three levels. Firstly, it is seen on the number of variables: more variables are needed to be sampled and the

optimiser needs to vary more the design variables; this leads to a loss of efficiency but also accuracy. Secondly, the number of simulations is exponentially increased, particularly excruciating for expensive simulations. And thirdly, the optimisation space for the parameters defining the model increases in complexity, requiring more time to train and reducing chances to find the best global optimum.

Solutions are proposed in the literature to tackle this issue. Songqing Shan and G. G. Wang (2010b) compiled an exhaustive literature review which can be summarised as follow:

Decomposition This includes three types of decomposition: *product decomposition* which divides the problem into physical components, *process decomposition* which includes the design process and *problem decomposition* which aims at simplifying a complex problem by solving simpler subproblems; this is what is traditionally used for multidisciplinary problems.

Screening By identifying design variables relative effect, it is possible to identify the predominant ones and to drop the ones with the smallest effect. Screening should be used with care as it has a direct effect on accuracy (Schonlau and William J Welch 2006).

Mapping Mapping aims at reducing the dimensionality of a problem by creating a set of uncorrelated variables from the original correlated ones. The goal is to reduce the number of design variables while keeping most of the initial information such that the true optimum is kept unchanged. This includes non-linear mapping and projection. This method category is also used in multifidelity analysis as presented in the next chapter.

Space reduction The individual range of each design variable can be adjusted to limit the space exploration. This can be achieved by sampling few points and reduce design space accordingly. This method is at the heart of any *trust region* based optimisation algorithm (Alexandrov, Dennis, et al. 1998).

Visualisation Designers are at the best position to reduce dimensionality of a problem thanks to their experience. Visualisation techniques can also help them to direct the optimisation toward the most promising places. This is a hot topic in the research world.

Figure 3.3 shows the recommended process for performing an optimisation on a high dimensional problem using expensive function. To deal with those functions,

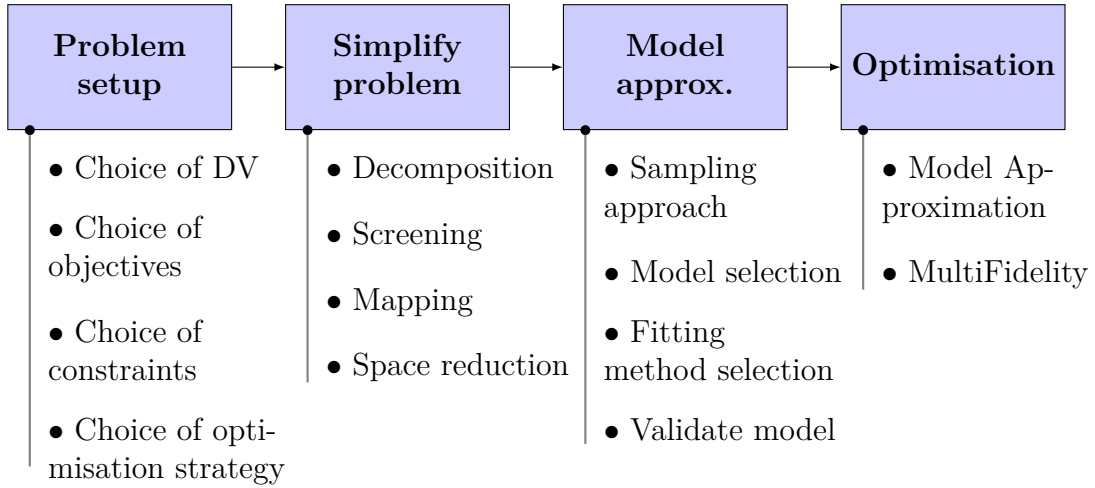


Figure 3.3: High dimensionality problems optimisation using meta-models, adapted from Songqing Shan and G. G. Wang (2010b).

it can be necessary to use surrogate models to lighten the computational power requirements. The first three boxes in figure 3.3 were already described in the previous sections. The last one, which is at the heart of the project presented here, contains two ways of dealing with the optimisation: the first one is only with the help of a model approximation and the second method is using multiple levels of approximations to solve the physical problem. Those methods are the subject of the next chapter.

3.4 Implementation

The tools used in this work is a collection of publicly and commercially available tools, as stated in this chapter, but also the author's own implementation (described in Appendix B). In particular, the author implemented the following codes:

1. The MultiObjective Particle Swarm Optimiser (MOPSO), with the contribution of Spyridon Kontogiannis;
2. The Radial Basis Function, Kriging and co-Kriging surrogate models;
3. The single- and multiobjective multifidelity optimisation frameworks;
4. The plotting library providing automatic plotting capabilities from the multifidelity optimisation results file.

Multifidelity Literature Review

The previous chapter introduced the aerodynamic models mainly based on the Navier-Stokes equations. Because of their non-linearities, they are extremely expensive to solve with computers. Approximations can be introduced to alleviate this cost at the expense of decreasing the model domain of applicability and reducing the level of accuracy. A compromise needs to be made between computational cost and overall accuracy. This chapter will answer the following question: how can the synergies between those approximations be used in the optimisation world? It is important to note that the following sections are an attempt of providing a classification of the different methods, but as any classification, some methods could be classified differently, and subjectivity had to be introduced.

4.1 Introduction

Multifidelity, also known as variable-fidelity, encompasses any method that uses at least two *models* representing the same *quantities* of interest. The term *model* needs to be defined as well: as introduced in Peherstorfer, Willcox, and Gunzburger (2016), we will consider that a model is anything that relates the outputs (quantities of interest) to the inputs. For example in CFD, a RANS simulation of an aerofoil is a model that, given the geometry and the flow conditions, will return the lift of such aerofoil. Similarly, using a surrogate as presented in chapter 3, section 3.3 is as well a model relating the training data (e.g. the shape of the aerofoil) to the surrogate response (e.g. the lift). Section 4.2 will describe those models more in details to answer the question: *what are the different fidelities?*

Figure 4.1 shows the process of using multifidelity for optimisation. It has 3 sections, that will be explained in details below. After having chosen the different models, they need to be blended together: methods range from simple filtering, where the low-fidelity substitute the high-fidelity in certain region where it is deemed accurate enough, to surrogate models inherently integrating multiple levels of fidelity such as the co-Kriging method. Section 4.3 is focusing on the different methods and answers the question: *how to fuse multiple fidelities?*

Most of the time the goal of the multifidelity is to rely on the accuracy of the

higher fidelity model while importing cheap (in the sense of computation cost) information from a lower fidelity model. Ideally, one would want to have the high-fidelity accuracy with the low-fidelity computation cost. This will rarely be achieved but multifidelity will bring improvement anywhere in-between. At present, multifidelity is used for problems where the model needs to be used recurrently and according to Fernández-Godino et al. (2016) the main ones are: 1. optimisation, 2. uncertainty quantification, 3. optimisation under uncertainty. This thesis is solely focused on the use of multifidelity for optimisation hence the final section, section 4.4 presents more in details the use of multifidelity applied to optimisation to answer the question: *how to use multifidelity for optimisation?*

4.2 Fidelity Types

Literature has many definitions for the term *multifidelity*. As an example, Fernández-Godino et al. (2016) presents a non-exhaustive table with the work done under the name of multifidelity for fluid mechanics, and they come to 4 main groups, described more in details below: 1. variation in the modelling of the physics, 2. variation in the numerical accuracy of a particular model, 3. the use of surrogate model, and 4. the use of experiment along with numerical models. The fidelity in the geometry can also be added to this classification although it can be distributed to all the other categories since having a coarse geometry is forcing the use of a coarser physic simulation.

4.2.1 Physics

Physics-based multifidelity is very popular because of the history of numerical computation: as explained in chapter 3, section 3.1, CFD has always been challenging to numerically model due to the huge computational resources required. Hence many types of simplification were used in early models in order to be suitable to the available computational power. Nowadays, those models are well understood and most of the time finely tuned with previous flying aircraft reinforcing their presence in industry. Using different models (with or without viscid effects, with or without compressibility, with or without unsteadiness, ...) gives access to a large range of fidelity as shown in figure 3.1 from chapter 3. The most common models are RANS, Euler and potential methods (Han, Görtz, and Zimmermann 2013). However, any model presented in section 3.1 can be considered as a different model fidelity includ-

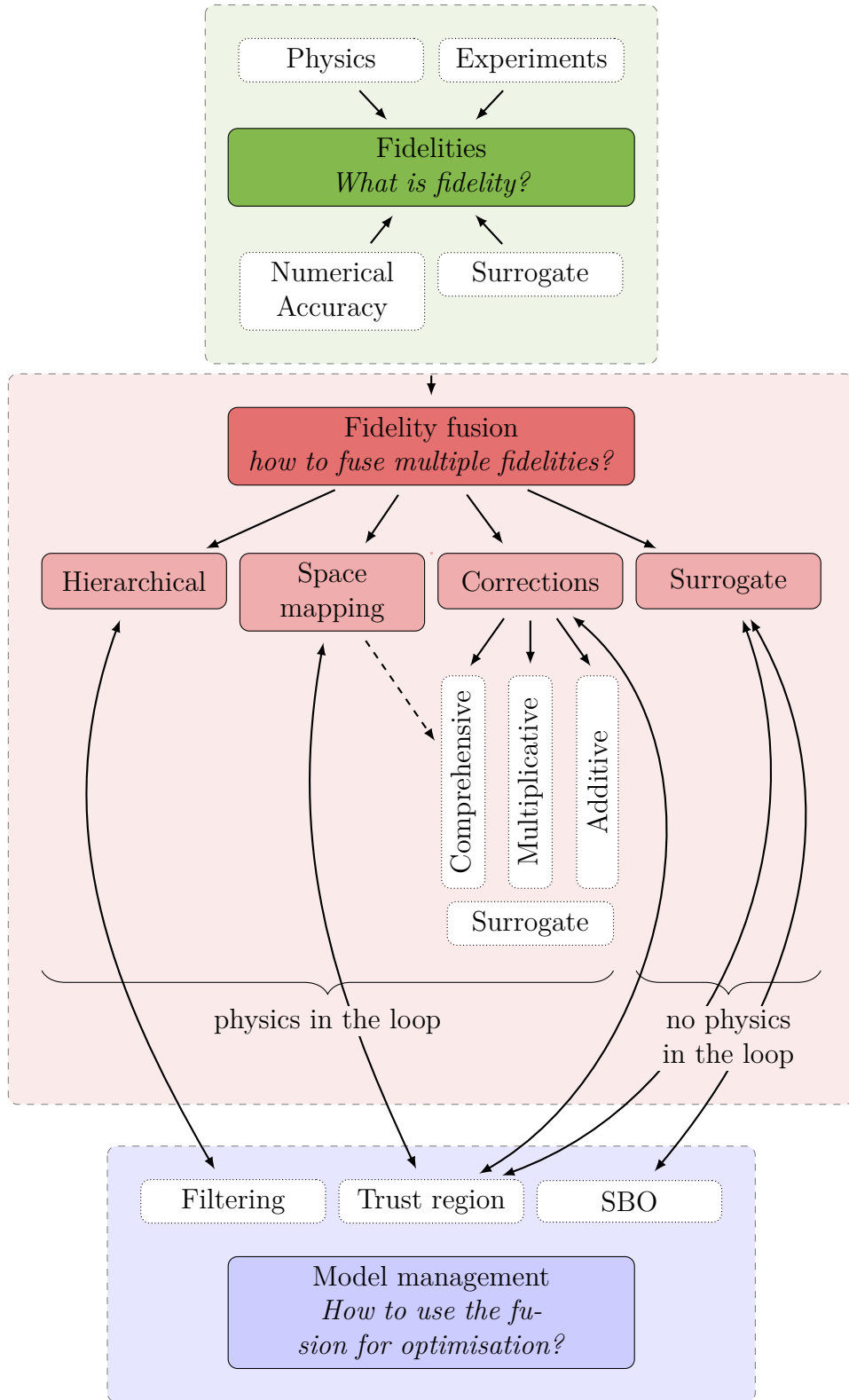


Figure 4.1: The 3-parts process to use multifidelity for optimisation.

ing Large Eddy Simulations that could be used as a high-fidelity model if unsteady features are important in the problem considered. As an example, Huang, Gao, and Zhang (2013) used a full potential method with a coupled boundary layer as low-fidelity model and a RANS simulation as high-fidelity for the three-dimensional optimisation of an aircraft. But a low-fidelity model can as well be considered high-fidelity for a different application: Umakant et al. (2007) used an inviscid model as high-fidelity and an oblique shock theory as low-fidelity.

The multifidelity work presented in this thesis (chapters 5, 6 and 8) relies partly on the physics-based multifidelity: the high-fidelity will use RANS simulations and one of the low-fidelity model will be MSES, presented in chapter 3, section 3.1.6, which uses an Euler solver along with a 2-equations boundary layer solver.

4.2.2 Numerical Accuracy

Despite their wide availability, the use of multiple numerical models in multifidelity optimisation requires the creation of separate pre- and post-processing procedures. It also requires, amongst other, a space (and time) discretisation, and using a single model with different parameters avoids duplicate work. From a reference model, the accuracy, and most of the time the wall-clock time, can be artificially decreased by varying the main parameters of the numerical schemes, namely (non exhaustive list): space (and/or time) discretisation, order of the numerical scheme, turbulence model, residuals tolerance of the iterative process (Peherstorfer, Willcox, and Gunzburger 2016). An example of such multifidelity optimisation can be found in Madsen and Langthjem (2001) and Mifsud, MacManus, and Shaw (2016) where a coarser mesh discretisation is used and Jonsson et al. (2015) and Courrier, Boucard, and Soulier (2014) where a relaxed convergence is used. The main drawback of this approach is that the meshing and the iterative process are not avoided for the low-fidelity model, forcing to use the heavy CFD process (meshing, pre- and post-processing, iterative solution) for both fidelities. It therefore involves a wall-clock time but also a disk space burden. Chapter 9 is using such approach for the optimisation of a three-dimensional wing.

4.2.3 Surrogate

The term *multifidelity* is not only used when the low-fidelity model relies on physics or numerical simplifications but also when the model is created from scratch only

based on the high-fidelity model at some control point locations. Two different types of surrogate models can be distinguished (Peherstorfer, Willcox, and Gunzburger 2016): *data-fit* and *dimension reduction*. Data-fit models use the high-fidelity model as a black-box and requires only few sampling points, as described in section 3.3 (chapter 3). Depending on the problem considered and the high-fidelity function shape knowledge, polynomials, RBFs or Gaussian processes can be used. This type of approach is generally called Surrogate Based Optimisation (SBO) in the literature. As a single model is required, this method is popular in the literature: Queipo et al. (2005) used a polynomial model on the optimisation of a liquid-rocket injector, A. J. Keane (2003) reduced the computational cost of a wing optimisation using a Kriging model, and similarly Emmerich, K. C. Giannakoglou, and Naujoks (2006) optimised a RAE aerofoil to demonstrate the capabilities of a Gaussian Process. Dimension reduction models are used when more underlying information about the function is known. Proper Orthogonal Decomposition (POD) (Berkooz, Holmes, and Lumley 1993) is popular amongst the CFD literature due its wide range of applicability and relative good performances for non-linear problems despite its linear procedure. The main idea is to find a lower-dimension orthogonal basis that can express the main properties of the function considered. This method includes as well the Principle Component Analysis (PCA), the Karhunen-Loève Decomposition (KLD) and the Singular-Value Decomposition (SVD) (Liang et al. 2002). Examples are as well widely spread in the literature (D. Alonso, Vega, and Velazquez 2010; Epureanu 2003). This approach will not be considered in the present work.

4.2.4 Experiments

On the other side of the spectrum of physics fidelity is the use of data fusion involving experimental data. The data can either be used as a high-fidelity source if the experiment is specific to the case studied and remains in its range of applicability, or used as a low-fidelity model when converted to carpet plots useable for a wide range of geometries, sometimes augmented by simplified method as employed in the Engineering Sciences Data Unit (ESDU) datasheets; this kind of approach is usually referred as semi-empirical models. Experiments require a considerable amount of time and can hardly be part of an optimisation loop, which makes their use not relevant in the present work. However, chapters 5, 6 and 8 are using ESDU datasheets, introduced in section 3.1.7 (chapter 3), as low-fidelity model and are compared to the MSES approach.

4.3 Fusing Fidelities

This section focuses on the methods to fuse the multiple levels of fidelity. We will restrict our self to the use of only two fidelity models. Figure 4.1 presents in the center a summary of the fusing methods: 1. hierarchical, 2. correction 3. space mapping and 4. pure surrogate models.

4.3.1 Hierarchical

The most intuitive method consists in using only one model at a time. Most of the time, the low-fidelity model helps reducing the design space and when criteria are met, the model management sequentially switches to a more accurate model (Choi, J. J. Alonso, and Ilan M Kroo 2009; Choi, J. J. Alonso, Ilan M Kroo, et al. 2008; Dalle and Fidkowski 2014). More than two model fidelities can be used in this case, but each model needs to have the same trends in order not to discard good design in the exploration phase. Sefrioui, Srinivas, and Periaux (2000) extended the method to the use of evolutionary optimisers. The method, called Hierarchical Genetic Algorithm (HGA), uses a tree-like structure where the top node, the more accurate, is fed by the node below with their best candidates. The lower nodes use models with a lower fidelity which allows to reduce the exploration cost while the upper nodes contribute to the exploitation with the same accuracy as traditional EA (K. Giannakoglou 2002).

Hierarchical methods rely solely on the model without the use of any surrogate. The following methods use a surrogate model to blend the low-fidelity with the high-fidelity models.

4.3.2 Correction

To use the low-fidelity model for each point evaluation in order not to rely only on a mathematical model, a surrogate model can be used to match at some high-fidelity control points the two models. The most common method available in the literature is using an additive, multiplicative or a combination of both correction models. In the case of an additive correction, at each high-fidelity sampled points, the low-fidelity function is as well evaluated and the difference between each model is used to construct a surrogate of the error $e(\mathbf{x})$. The correction function value is

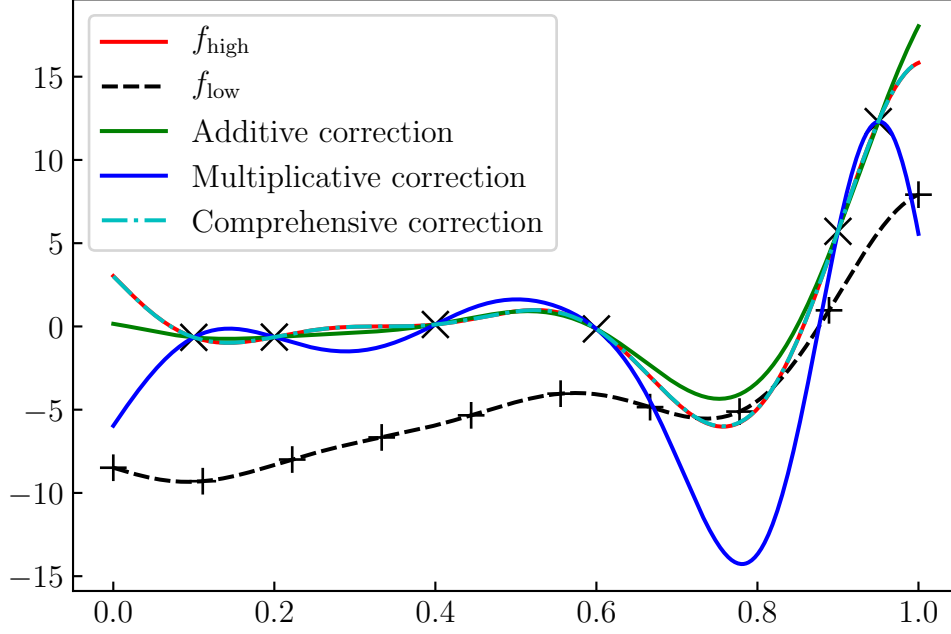


Figure 4.2: Comparison of multifidelity additive, multiplicative and comprehensive corrections

then obtained with equation (4.1):

$$f_{\text{corr}}(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) + e(\mathbf{x}). \quad (4.1)$$

Similarly, the ratio between the low- and high-fidelity functions can be computed instead of the difference, and a surrogate $r(\mathbf{x})$ can be used, as in equation (4.2):

$$f_{\text{corr}}(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) \cdot r(\mathbf{x}). \quad (4.2)$$

Finally, combining both approaches, the corrected function can be obtained according to equation (4.3):

$$f_{\text{corr}}(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) \cdot r(\mathbf{x}) + e(\mathbf{x}), \quad (4.3)$$

referred as *comprehensive* method in the literature (Fernández-Godino et al. 2016). An example of such corrections is shown in figure 4.2 on a one-dimensional function.

4.3.3 Space Mapping

The previous approach focuses on correcting the function responses. Alternatively, space mapping techniques (Bandler et al. 1994) are designed to solve similar problematic but acting on the design variables. A space mapping technique can be described as follow:

$$f_{\text{corr}}^{\text{SM}}(\mathbf{x}) = f_{\text{low}}(\boldsymbol{\zeta}_{\mathbf{z}}(\mathbf{x})), \quad (4.4)$$

with $\boldsymbol{\zeta}_{\mathbf{z}}(\mathbf{x}) = \boldsymbol{\zeta}(\mathbf{x}, \mathbf{z})$ the transformation function parametrised by $\mathbf{z}$, usually found by solving the least-square minimisation showed in equation 4.5 (Leifur Leifsson and Slawomir Koziel 2015):

$$\mathbf{z} = \arg \min_{\mathbf{r}} \sum_{k=1}^p \|f_{\text{high}}(\mathbf{x}_k) - f_{\text{low}}(\boldsymbol{\zeta}_{\mathbf{r}}(\mathbf{x}_k))\|. \quad (4.5)$$

The method is applied to CFD application to predict an aerofoil lift and drag for two mesh sizes in Leifur Leifsson and Slawomir Koziel (2015), but as pointed out by Alexandrov and Lewis (2001) space mapping is applicable to problem where the low- and the high-fidelity function show similar behaviour and trends. As the correction is applied on the design space, it also supposes that the low-fidelity models the same quantity as the high-fidelity and uses a similar parametrisation, but there is also no guarantee that such a transformation exists.

4.3.4 Surrogate

Methods introduced above always keep the simulation model in the prediction loop. It has the advantage of keeping the physics in the prediction leading to a better understanding of the corrected model. However the major drawback is the increase in the prediction time if the low-fidelity computational time is not negligible (more than a couple of seconds for example). To alleviate this problem, but as well to improve the correction of the low-fidelity model in order to reach the high-fidelity model accuracy, multifidelity surrogate models have been proposed. The most popular is the co-Kriging model, already introduced in section 3.3.5 in chapter 3. This model uses statistical tools to better capture the correlation between the low- and the high-fidelity models generally leading to a good prediction quality (Forrester, Sóbester, and Andy J. Keane 2007). Despite reducing the evaluation cost of the corrected function (since the low-fidelity model is not used in the loop anymore), the model

training cost must be considered as a computational overhead each time new low- or high-fidelity points are added.

4.4 Model Management Strategy

Previous sections were directed toward the definition of fidelity and the different methods to blend them. But the optimisation requires many model evaluations and the multifidelity method aims at using the low-fidelity model for speed up and the high-fidelity for the accuracy. Hence a proper model management is needed to balance the different fidelity models use. The model management is also required to obtain the optimisation convergence to the high-fidelity optimum. This section is focusing only on the use of a model management for optimisation, but different methods are also available for uncertainty quantification and statistical analysis.

The lower part of figure 4.1 shows the three main management techniques: 1. filtering that uses some sort of criteria on the low-fidelity confidence to switch to the high-fidelity model, 2. trust-region method that uses a trust-region to define accurate design space regions in order to decide when to sample the high-fidelity model, and 3. Surrogate Based Optimisation (SBO) that uses only a surrogate model that is being refined at each iteration.

4.4.1 Filtering

Aircraft industries rely already on the use of multiple levels of fidelity. Most of the time, the lower fidelity models are used in the preliminary stages and the more confidence in the configuration the higher the fidelity model. Hence a natural approach in using multiple levels of fidelity for the optimisation of aerodynamic function is to extensively use the lower-fidelity model to down select configuration in the regions the low-fidelity model is known to have a good accuracy, and then use the higher fidelity models to increase the low-fidelity model application range. One example of such method is provided by Choi, J. J. Alonso, and Ilan M Kroo (2009) where a two-stages optimisation is used: first a gradient free optimisation is performed on a large design space with the low-fidelity model to select most promising configurations, then followed by a more detailed optimisation with a local gradient-based algorithm performed on the high-fidelity method. The method provided improvements compared

to existing designs but the study concluded that the sequence of low and high-fidelity steps needed to be selected carefully.

4.4.2 Surrogate Based Optimisation

When a surrogate model is constructed, from a set of training points, a natural approach is to perform the optimisation on the model directly without using the true model. Despite the speed of the optimisation, the optimum found is not the true model optimum. If the surrogate is accurate enough, the direction will be correct. To alleviate this problem the surrogate model can be updated with the real model at each new candidate; such approach is referred as adaptive sampling or Sequential Approximate Optimisation (SAO) in the literature. But as explained by March and Willcox (2012a), without a proper model management method, also called infill criteria in the context of SBO, there is no guarantee to converge to the real model optimum.

Global optimisation, that is the search of an optimum in the whole design space, does not usually have a proof of convergence. It will always be possible to find an esoteric function where the minimum can be located only by a full design space search. However to alleviate the convergence problem arising from the adaptive SBO methods, Jones, Schonlau, and William J. Welch (1998) proposed an Efficient Global Optimisation (EGO) method that balances exploration and exploitation with the help of stochastic processes (Björkman and Holmström 2000). Models that are built by considering them as the realisation of a stochastic process, such as Kriging and co-Kriging models, provide on top of the mean prediction, the standard error at any point. Those two quantities can be combined to force the exploration of promising design space regions by constructing the Expected Improvement (EI) $E[I(x)]$ (Forrester, Sóbester, and Andy J. Keane 2008; Jones, Schonlau, and William J. Welch 1998), according to equation (4.6):

$$E[I(x)] = (y_{\min} - \hat{y}(\mathbf{x})) \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{\hat{s}\sqrt{2}} \right) \right] + \hat{s} \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{(y_{\min} - \hat{y}(\mathbf{x}))^2}{2\hat{s}^2} \right), \quad (4.6)$$

with erf being the error function: $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dx$. As pointed out by Forrester and Andy J. Keane (2009), the expected improvement is most likely the best infill

criteria for finding the global optimum explaining its popularity for expensive black-box functions optimisation (Meunier 2009; Timme, Marques, and Badcock 2011). A drawback of such method is the need for an accurate model over the whole design space, and therefore, as the optimisation progresses, the number of points will increase leading to a higher model construction complexity favouring the appearance of singular matrices during the model training (Jones, Schonlau, and William J. Welch 1998). Finally, since the simulation models (low- or high- and low-fidelity) are not used within the infill selection process, the surrogate must be constructed on the whole design space from scratch; a design of experiments must therefore be used prior starting the optimisation, even in locations that are not promising.

Literature is replete with SBO examples. Iuliano (2017) presents a methodology using a Proper Orthogonal Decomposition (POD) supplemented by a RBF surrogate model. The model management is performed with a EI-like method: since no variance prediction is provided by the RBF, a similarly shaped function is used for the EI calculations. The method is applied on a traditional two-dimensional aerofoils such as the RAE 2822 aerofoil, providing similar accuracy than EGO or genetic algorithm for 5% of the computational cost. Another example is given in (D. Wang et al. 2014) where a Kriging model is used along with a genetic algorithm.

4.4.3 Trust Region

In chapter 3, the section 3.2.3 presented the trust-region optimisation concept: it is usually based on a quadratic approximation of the single-fidelity model for solving the subproblem inside an expendable and shrinkable region. Lewis (1996) generalised this idea to any type of approximation while proving the same convergence properties (true local optimum). The same management can be applied to multifidelity optimisation as demonstrated in the work of Alexandrov, Lewis, et al. (2001) where a first-order Approximation and Model Management Optimiser (AMMO) is presented. Using a generic model enhanced by the gradients gives a first-order accurate model that, when part of a trust region management, converges to the true optimum. The method is applied to various problems including a 2-elements aerofoil optimisation where the low- and high-fidelity models do not necessarily presents the same trends due to the viscosity (Alexandrov and Lewis 2001; Alexandrov, Lewis, et al. 2001).

Chapter 2 explained the reasons the gradients are not easily accessible. We therefore require a method that can use multiple fidelity models that presents different trends but that do not necessarily require gradient informations. AMMO is extended,

along with the proof of convergence, to the use of general multifidelity approximation models without the use of gradients to form a 0th order model. Starting with Ouevray and Bierlaire (2009), the trust region method has been extended so that the convergence to the high-fidelity value is still verified but without the need of explicitly matching the derivatives. To obtain such agreement, Conn, Scheinberg, and Vicente (2009) proved that the model and its sampling needed to verify some constraints that have been implemented in the ORBIT framework by S. M. Wild, Regis, and Shoemaker (2008) and S. M. Wild and Shoemaker (2013) using a RBF interpolation as surrogate. This work was extended by March et al. with a Bayesian approach for the model calibration ie the surrogate parameters (March and Willcox 2012a,b). Here again the surrogate used was the RBF with Gaussian basis functions. The method was also successfully applied to a supersonic aerofoil design with more than two levels of fidelity using a maximum likelihood estimator. A downside of this approach is the model creation that becomes increasingly difficult as the input dimension increases (Alexandrov and Lewis 2001).

This approach while converging to a local optimum can also be combined with an expected improvement criterion to allow a more global exploration. In addition, the low-fidelity model can be kept in the suboptimisation for the candidate selection in the initial optimisation stage when not enough points are available to build the surrogate model; there is therefore no need to perform a design of experiment prior the optimisation, and all points evaluated are suggested either by the high- or the low-fidelity model information. Due to the versatility of such approach, the method is retained for the remaining of the thesis. The method available in the literature is presented in chapter 5 and extended to multiobjective application in chapter 6.

4.5 Multifidelity Multiobjective

Aerodynamic problems can be considered multiobjective when conflicting quantities are used (J. Wild 2002). Thus to extend the use of multifidelity optimiser in the aerodynamics community, the methods should be used with the concept of Pareto optimality. Benini, Ponza, and Massaro (2011) and G. Trapani, T. Kipouros, and A.M. Savill (2012) showed that the design of multielement aerofoil in a multiobjective context allows a greater freedom for the decision maker. It is thus natural to look for multiobjective application of multifidelity optimisation. Yang, Yeun, and Ruy (2002) present an Adaptive Approximation in Single Objective (AASO) optimiser

which is described by the SBO box of figure 4.1 but also an Adaptive Approximation in Multi Objective (AAMO) optimiser. The method is built around a MultiObjective Genetic Algorithm MOGA: a sampling is first done on the high-fidelity function and an approximation model is constructed; MOGA is then run on this model and some of the Pareto points are validated with the high-fidelity model (taking care that no clustering in a single portion of the design space appears); then the new points are used to update this model and finally the MOGA is re-run starting from the previous Pareto front. A single objective optimisation is also performed on the final Pareto points to increase the exploitation. This does not use any physics based low-fidelity tool but the method can be extend to include it.

More than two models can be used either for single- or multiobjective optimisation. Singh et al. (2017) used a three-level multifidelity co-Kriging model that blended analytical, CFD and experimental models of a cyclone separator. They used a EGO multiobjective implementation, namely the Efficient Multiobjective Optimisation (EMO), from Couckuyt, Deschrijver, and Dhaene (2014) to manage the SBO. However the whole optimisation was performed *offline*, without any adaptive sampling.

A standard approach for multiobjective optimisation is the use of evolutionary algorithms as presented in section 3.2.4 from chapter 3. Such optimisation cost is prohibitive for expensive function optimisation such as CFD. L. Leifsson, S. Koziel, and Tesfahunegn (2016) uses a space mapping technique to build a model approximating the high-fidelity function on which the evolutionary optimisation is performed. The model is then regularly updated with the new Pareto candidate points until termination criteria. This approach does not use a model management and therefore one cannot expect to find the exact high-fidelity Pareto front.

Output space mapping technique is extended to multiobjective in the work of S. Koziel et al. (2017). The Pareto ranking bisection algorithm (PRBA) method is used to find the new Pareto candidate points. The mapping uses a second order polynomial method, applicable only for similar low- and high-fidelity function. A similar work applied on an aerofoil optimisation using a Euler equation solver for 2 mesh sizes is presented in (A. Amrit, L. Leifsson, and S. Koziel 2017). The Pareto-front is sequentially approximated with a Kriging model and refined with a co-Kriging multifidelity model. They show that the multifidelity framework provides the same Pareto front for a lower computational cost. The framework is quite complex and involves simple aerodynamic solver and objectives. It may be difficultly applicable to problems involving more complex aerodynamic features.

Zhu, Y.-J. Wang, and Collette (2014) extended their initial work with a physics-based low-fidelity tool including a modification so that the Pareto points chosen for validation fill their cluster available slots in order to use the maximum computational power available. The link between the low and the high-fidelity tools is ensured by a Kriging model.

Finally, Mejía-Rodríguez, Renaud, and Tomar (2010) implemented a trust region based multiobjective optimisation. However the necessity to compute the gradients to obtain the first order agreement makes it hardly applicable to expensive functions with inaccurate gradients.

No extension of the trust-region-based multifidelity optimisation method to multiobjective problems was found in the literature. It seems that such approach would alleviate the computational cost of evolutionary algorithms while preserving better convergence properties. These methods were applied to various problems ranging from mathematical functions to marine structure optimisation but they remain to be tested on more challenging problems that is the aerodynamic high-lift aerofoils design.

4.6 Multifidelity Applied to High-Lift Devices Optimisation

The flow around an high-lift configuration involves many complex flow features as described in chapter 2. March and Willcox (2012b) highlight the fact that trust-region based multifidelity methods are sensitive to the low-fidelity tool quality. Since it is harder to find a reliable tool for multi-element aerofoil especially at maximum lift angle of attack, application of multifidelity methods to multi-element design optimisation is particularly challenging.

Some attempts of using multifidelity for high-lift design has been done. Regarding Surrogate Based Optimisation, Kanazaki et al. (2006) performed an expected improvement optimisation on a Kriging metamodel using RANS simulation as the high-fidelity model. They used statistical methods to highlight the importance of the flap deflection and gap on the multielement aerofoil maximum lift without conducting any optimisation. Regarding multifidelity using two models, the work of Alexandrov, Lewis, et al. (2001) shows an application of their first order consistent method in (Alexandrov, Nielsen, et al. 2000) to a two-elements configuration comprising a main element and a flap. The optimisation focuses on drag improvements

by modifying the angle of attack and the y displacement of the flap. The low-fidelity model is an Euler solution whereas the high-fidelity one is using a RANS model, both are using unstructured tetra meshes. The sensitivities needed for first order agreement are obtained with the adjoint method. Despite the high dissimilarities between the models, the method yields a 5 times reduction on the number of high-fidelity calls. Results are encouraging but the evaluation of the sensitivities remain a challenge nowadays. Moreover, high-lift aerofoil design should involve high-angle of attack as encountered during take off and landing for maximum lift identification. Increasing the angle of attack increases the discrepancies between Euler and RANS solver due to the increase in separation and wake interaction.

Finally, Greenman and Roth (1999) performed another work on multielement aerofoil but without the use of low-fidelity based models. They used a neural network to serve as a surrogate model. Incompressible CFD with a Spalart-Allmaras model is used for estimating the lift. The objective is the lift maximisation and the design variables are the flap position and deflection, and the angle of attack. An empirical correction is also used for estimating the maximum lift value, but no information on the integration is given. Despite the inherent design space sampling necessary to train the model, large reduction in terms of high-fidelity calls are achieved.

4.7 Method Choice

The method will be based on the zeroth-order trust region based method because:

- the first order version was successfully applied to high-lift aerofoil optimisation;
- the method keeps the physics in the centre of the suboptimisation, rendering an initial design of experiments not necessary, amongst other;
- the method remains efficient in terms of high-fidelity function calls while converging to the optimum;
- the method is modular: it is not bound to any particular surrogate model, aerodynamic solver nor optimiser;
- the correction model can be adjusted depending on model correlations;
- the method can be extended to multiobjective problems.

Multifidelity Optimisation

Methodology: Application to Single-Objective Optimisation

Prior to the introduction of the multiobjective concept to multifidelity optimisation, the method from the literature is implemented and applied to a single-objective optimisation. The trust-region based multifidelity method specificities are described in section 5.1, followed by the details on the Python implementation in section 5.2. To demonstrate the possibilities and the method correct behaviour, the optimisation of one- and two-dimensions functions are shown in section 5.3. Then, a simple test case from the literature is used to introduce aerodynamic optimisation into the problem, in section 5.4. This includes as well a study on the initial trust region effect, not found in the literature. The novelty of the chapter is as well to bring derivative-free trust-region based multifidelity optimisation to the complex aerodynamic optimisation of a multielement aerofoil. This is done on an academic test case in section 5.5 and then on an industrial one in section 5.6. Finally, conclusions are drawn in section 5.7.

5.1 Method

Two approximations of the aerodynamic physics are used in order to speed up the optimisation convergence while retaining accuracy from the most accurate model. The first model is relatively fast (less than a minute to obtain the converged solution) but relies more on empirical corrections; it will be called the *low-fidelity* estimator. One can argue that this model includes many empirical correction that are more accurate in terms of physics than purely numerical solutions, yet the accuracy decreases when going away from the applicability range; this is when the higher fidelity predictor, which has a more consistent response, will come into play. This model is more accurate than the low-fidelity one but requires more time to run (from 15 minutes for single point 2D simulations to several hours for industrial applications requiring a

full polar and 3D computations); it will be referred in the rest of the document as the *high-fidelity* estimator.

Let's define the low-fidelity function as $\mathbf{f}_{\text{low}} : \mathbb{R}^n \rightarrow \mathbb{R}^l$ and the high-fidelity function as $\mathbf{f}_{\text{high}} : \mathbb{R}^n \rightarrow \mathbb{R}^l$; where n is the design space size and l is the objective space size which is equal to 1 in this chapter: $l = 1$. The goal is the minimisation (or maximisation) of the high-fidelity function as defined in equation (5.1), where $\mathbf{x}$ is a vector of dimension n and $\mathcal{D}$ is a subset of $\mathbb{R}^n$ defined by the user-provided optimisation boundaries:

$$\min_{\mathbf{x} \in \mathcal{D}} \mathbf{f}_{\text{high}}(\mathbf{x}). \quad (5.1)$$

Similarly to the standard trust-region optimisation algorithm presented in chapter 3, section 3.2.3, instead of directly solving the high-fidelity problem, an approximation model is used. However, in contrast to the standard trust region, since the gradients are not easily and accurately obtainable, the derivative-free trust region method is used as a starting point. At iteration k the problem (5.2) is solved under a trust region $\mathcal{T}_k$:

$$\min_{\mathbf{x} \in \mathcal{T}_k} \mathbf{f}_{\text{corr}_k}(\mathbf{x}). \quad (5.2)$$

The trust region can either be defined in the design variables space or in the objective space; more details on the implementation is given in section 5.1.4.

The model $\mathbf{f}_{\text{corr}_k} : \mathcal{T}_k \rightarrow \mathbb{R}^l$ is defined by equation (5.3) at iteration k . It is formed by the sum of a surrogate model $\mathbf{e}_k$ representing the difference between the low- and the high-fidelity functions (the error in other terms) and of the low-fidelity function.

$$\mathbf{f}_{\text{corr}_k}(\mathbf{x}) = \mathbf{f}_{\text{low}}(\mathbf{x}) + \mathbf{e}_k(\mathbf{x}) \quad (5.3)$$

Figure 5.1 shows the method flowchart; the following sections focus on each elementary procedure. The low-fidelity information is available from the beginning of the multifidelity optimisation which, unlike SBO methods, makes an initial sampling of the high-fidelity function not mandatory. Therefore, the method starts with a single sampled point, the datum, provided by the user, defining a constant correction at iteration 0. The subproblem (5.2) needs to be solved at each iteration in the trust region. Section 5.1.1 deals with the surrogate model $\mathbf{e}_k$ that provides the correction to the low-fidelity function, and section 5.1.2 presents the suboptimiser. Once the suboptimisation is solved, the predicted improvement needs to be compared to the actual improvement by sampling the high-fidelity function at the problem (5.2) so-

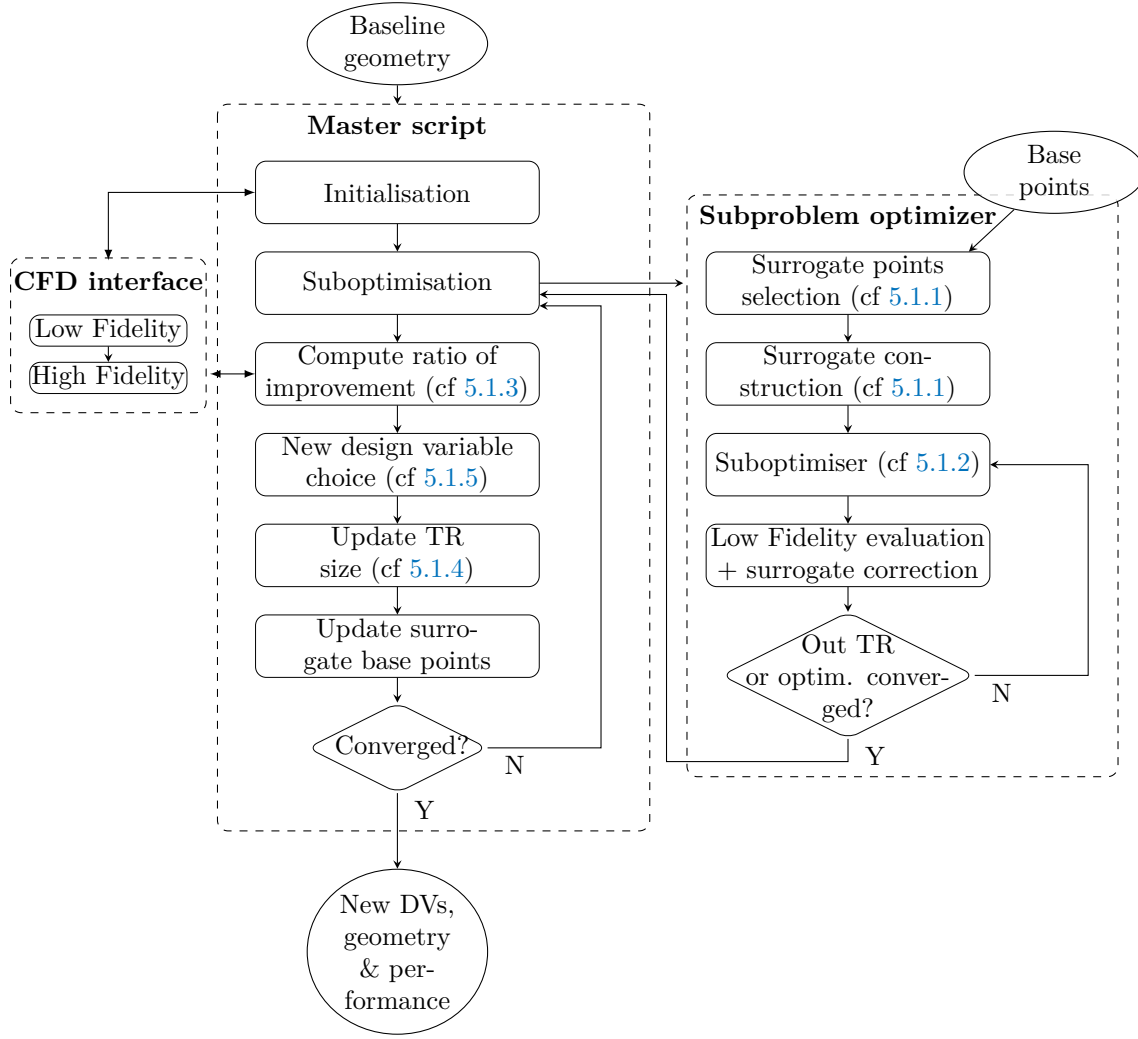


Figure 5.1: Single-objective multifidelity optimisation framework.

lution point; this part is explained in section 5.1.3. According to the high-fidelity value, the trust region size needs to be updated (section 5.1.4) and the next iteration starting point needs to be selected (section 5.1.5).

5.1.1 Surrogate Model

Standard trust-region algorithm uses a local quadratic approximation (first two terms from the Taylor-series expansion) that requires the gradients. To circumvent the lack of gradients, more generic surrogate models are used. The surrogate model that corrects the low-fidelity function must be constructed with sparse data points since a

limited number of high-fidelity sampled points is available in the first iterations. Most of the computation cost must be in the high- and the low-fidelity models evaluation, hence the surrogate model should not require a significant time to evaluate new points nor to be constructed. Moreover, the data used to build the surrogate comes from CFD near maximum lift, a non-steady phenomenon; the convergence achieved is usually lower than for a steady flow, introducing noise to the results. RBF methods are well suited for highly non-linear functions (Fang and Horstemeyer 2006) and cheap to tune (setting the correlation parameter) but only one variable is used for all directions reducing their ability to model complex high-dimensional functions. The simplicity of the training procedure and the good performances makes them suitable to solve the subproblem until a more advanced surrogate is required. In the work done for sections 5.5 and 5.6, RBF showed limits in accuracies requiring a more advanced correction model. To extend the RBF method, and more specifically the Gaussian RBF, different parameters for each direction (variable) can be used to provide fine control on the Gaussian functions width in all dimensions. Furthermore, the Gaussian kernel exponent, fixed at two for the Gaussian RBF, can be replaced by a different value for each direction. That's why a Kriging model, described in chapter 3, section 3.3.4, is included and used in the more complex cases.

No sampling is performed prior the optimisation but the Kriging model is used when at least two points are available. When the database is composed of only one suitable point, the error $\mathbf{e}_0$ is constant across the whole design space and its value is: $\mathbf{e}_0(\mathbf{x}) = \mathbf{f}_{\text{high}}(\mathbf{d}_0) - \mathbf{f}_{\text{low}}(\mathbf{d}_0)$. When more points are available, they are filtered so that the distance between each point is greater than ϵ_{neigh} in order to avoid points clustering and to reduce numerical instabilities (Lian, Oyama, and Liou 2010). The remaining points, used as training points, are called *base points* because they define the basis for the surrogate model.

5.1.2 Subproblem Optimiser

The subproblem optimisation solves the problem (5.2) with the error surrogate being constructed for the objective function as described in section 5.1.1. For each function call, the surrogate is evaluated at the current point to obtain $\mathbf{e}_k(\mathbf{x})$ along with the low-fidelity function at the same point. The corrected function $\mathbf{f}_{\text{corr}_k}$ is obtained by adding for each objective the low-fidelity function and the correction model (equation 5.3). Once the trust region is exceeded or the suboptimiser convergence criteria fulfilled,

the subproblem is considered solved and the point, considered as a candidate point for the trust-region management algorithm, is fed back to the master script.

The method is not bounded to any particular suboptimiser. So any optimiser can be used to fit any problem considered. As highlighted previously, the problem types considered in this thesis involve most of the time multiple minima and complex functions, therefore advanced optimisers are to be used to provide a good trade off between design space exploration and local search in order to get closer to the global optimum, without requiring gradients that are difficult to obtain. Finally, since the low-fidelity (that may need a minute to evaluate) is still in the loop, the optimiser should be efficient relative to the number of function evaluations.

The different optimisers used throughout the thesis were presented in chapter 3 and will not be reintroduced here.

5.1.3 Ratio of Improvement Computation

Once the subproblem (5.2) is solved, the candidate point is evaluated by the high-fidelity function in order to compute the *ratio of improvement* at the heart of the trust-region based algorithm. A candidate point $\mathbf{x}_k$ is accepted or rejected and the trust region size is increased or decreased according to the ratio of improvement ρ_k , defined as the actual improvement over the predicted improvement:

$$\rho_k = \frac{f_{\text{high}}(\mathbf{x}_k) - f_{\text{high}}(\mathbf{x}_s)}{f_{\text{corr}_k}(\mathbf{x}_k) - f_{\text{corr}_k}(\mathbf{x}_s)}. \quad (5.4)$$

An improvement of the function f_{high} translates to a ratio of improvement greater than zero, meaning that for $\rho_k < 0$, the corrected function does not have the correct trends but becomes more accurate as the ratio tends to one. Typically, for complex aerodynamic functions, this ratio will stay relatively low, meaning that the corrected function will rarely be perfectly accurate.

5.1.4 Trust Region Management

The trust region is usually defined as an Euclidean ball in the design space, centred on the subproblem initial point (Conn, Scheinberg, and Vicente 2009; Nocedal and Wright 1999). However this definition is local and does not apply well to multi-objective problems, when the points are scattered over the design space. A second definition based on the number of improvements of the corrected model is used (Jar-

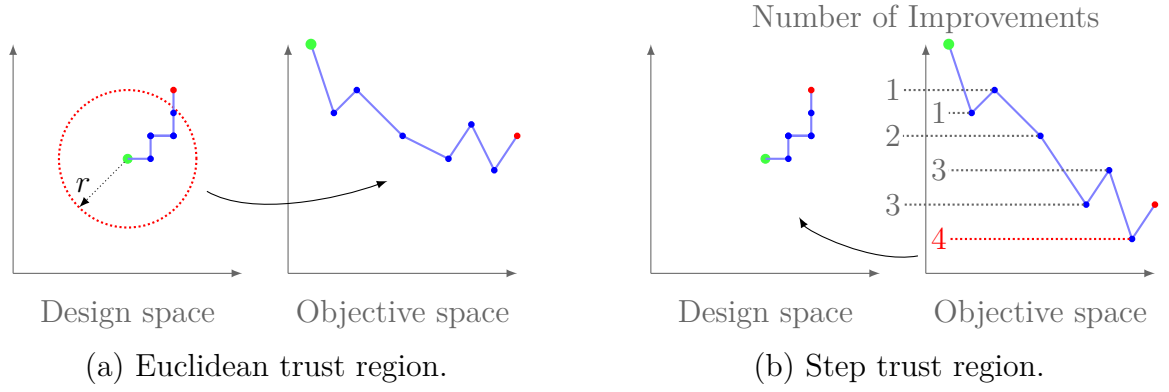


Figure 5.2: Illustration of the Euclidean (a) and of the step (b) trust regions in the design and objective spaces.

rett and Ghisu 2012): the trust region is defined as the number of improving points found, or for a multiobjective objective optimisation, the number of dominating or Pareto-equivalent points found during the suboptimisation. The distance in the design space, from the subproblem starting point, is no longer fixed but in this case, the model is trusted only for a certain number of improvements. On the one hand, figure 5.2a illustrates the Euclidean trust region that applies on the design space and the other hand figure 5.2b illustrates the step-based trust region applying on the objective space. The latter requires more explanation: starting from the subproblem initial point (in green), the first move in the design space leads to an improvement, increasing the counter. The second move does not provide any improvement and therefore the counter remains at 1. Each improvement increases the counter, until the limit, in this example 4, is reached. The step-based trust region is particularly suited to step-based optimisers such as Tabu Search since the design and objective spaces are related. However, nothing prevents its use with more generic optimisers.

For both methods, when the trust region is violated (limited space is exited or the number of improvements required is exceeded), the subproblem is considered as resolved and the new optimum point is returned to the master optimiser for the ratio of improvement computation. The trust region size is then increased or decreased according to the ratio of improvement value as follow:

$$\Delta_{k+1} = \begin{cases} \Delta_k / \gamma_s & \text{if } \rho_k \leq \rho_{\text{bad}} & \text{(bad prediction)} \\ \Delta_k & \text{if } \rho_{\text{bad}} < \rho_k \leq \rho_{\text{good}} & \text{(moderate prediction)} \\ \Delta_k \times \gamma_e & \text{if } \rho_k > \rho_{\text{good}} & \text{(good prediction)} \end{cases} . \quad (5.5)$$

5.1.5 New Point Selection

Finally, the ratio of improvement is used to select the next iteration subproblem starting point (see (5.6)): if the ratio of improvement is too low (below ρ_{bad}), the corrected function is not accurate enough and the suboptimiser was led in the wrong direction. Therefore, the trust region algorithm dictates that the corrected function must be improved, which is done by the high-fidelity point already sampled for the ratio of improvement calculations. The suboptimisation is also restarted from the previous optimum. Alternatively, if the high-fidelity function was correctly approximated, the new point $\mathbf{x}_s$ is accepted and used as the next iteration subproblem starting point.

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_s & \text{if } \rho_k > \rho_{\text{bad}} \\ \mathbf{x}_k & \text{if } \rho_k \leq \rho_{\text{bad}} \end{cases} \quad (5.6)$$

5.2 Implementation

The method implemented is aimed to be used with various aerodynamic and surrogate tools. The code is made with the objective of being extensible and modular. Appendix B.3 presents multiple UML diagrams of the trust-region-based multifidelity single-objective framework implementation, done in Python (see figures B.3, B.4 and B.5).

The multifidelity core program is designed to run in an academic environment with the Tabu Search (TS) C++ implementation. Hence only the TS job submission, monitoring and results parsing is implemented in the *run_optim_and_wait* method. The correction model is handled by the suboptimisation process. The Multifidelity core however handles all the files output (for data transfer to the suboptimisation, and for logging purposes), the CFD (low- and high-fidelity) and the trust-region algorithm. Each CFD function has a *cached version* in order to avoid rerunning points already evaluated. Finally, the Multifidelity core can restart an optimisation after the maximum number of iterations is reached or if a unexpected problem stops the process before completion.

The method is as well used in the industrial sponsor environment. To adapt the program to any environment, the Multifidelity core class can be inherited and specific methods can be overridden: in this case, the CFD functions are rewritten to fit the industrial system (software and queue system are different). The *run_*-

optim_and_wait method is also entirely rewritten to include the suboptimisation and the surrogate model for the correction¹. Lastly, new class methods are included to perform high-fidelity- or low-fidelity-only optimisations.

5.3 Mathematical Test Functions

To demonstrate the single-objective multifidelity trust-region algorithm behaviour, firstly a one-dimensional and secondly a two-dimensional test functions are minimised in this section. Both examples are using a RBF model for the correction and the Nelder-Mead Simplex optimiser (see chapter 3).

5.3.1 One-Dimensional Example

The test function from Forrester, Sóbester, and Andy J. Keane (2008) is used, as well as its low-fidelity counterpart, see equations (5.7) and (5.8):

$$f_{\text{high}}(x) = (6x - 2)^2 \sin(12x - 4), \quad (5.7)$$

$$f_{\text{low}}(x) = \frac{1}{2}f_{\text{high}}(x) + 10\left(x - \frac{1}{2}\right) - 5. \quad (5.8)$$

Those functions have a local minimum and a different global minimum, as shown on the left hand side of figure 5.3. The low-fidelity function presents similar trends but accentuates more the local minimum than the global one.

The function is optimised with both trust-region definitions and with the optimiser calling only the high-fidelity function. Figure 5.3 presents the convergences comparison: the Euclidean-based optimisation converges to the local minimum due to the small initial trust region and due to the local behaviour of the low-fidelity function. Increasing the initial trust region size allows to find directly the global optimum. The step-based optimisation however finds quickly the global optimum thanks to the liberty offered by the trust region defined on the objective space.

5.3.2 Two-Dimensional Example

The demonstration is done as well on the two-dimensional Rosenbrock function (Rosenbrock 1960), with the low-fidelity function being a polynomial, according to

¹The same methods were reimplemented in Python.

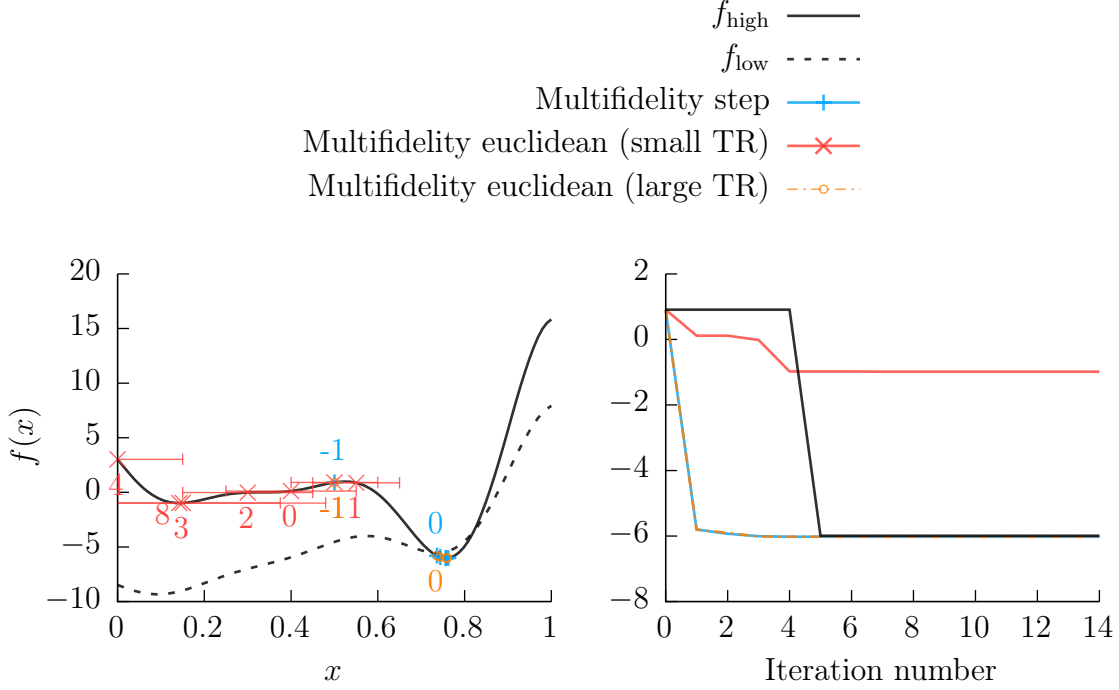


Figure 5.3: Single- and multifidelity optimisations on the 1D function shown on the left (numbers are the iteration number) and convergence plot shown on the right.

equations (5.9) and (5.10):

$$f_{\text{high}}(x_1, x_2) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2, \quad (5.9)$$

$$f_{\text{low}}(x_1, x_2) = x_1^4 + x_2^2. \quad (5.10)$$

The optimisation is started from the point $\mathbf{x}_0 = [-2, 2]$ for all methods, the lower bounds are both set to -3 and the upper ones to 3 . The high-fidelity-only optimisation is done with the Simplex algorithm, which is as well the suboptimiser for the multifidelity runs.

Figure 5.4 shows the convergence of both trust regions definitions compared with the high-fidelity-only one. On the left, the convergence is shown with respect to the number of high-fidelity function calls and no matter the trust-region definition, the multifidelity requires less function calls, at the expense of more low-fidelity function calls. On the right, the convergence is shown with respect to the elapsed time: no time difference is visible between the two trust-regions. In this case both low- and high-

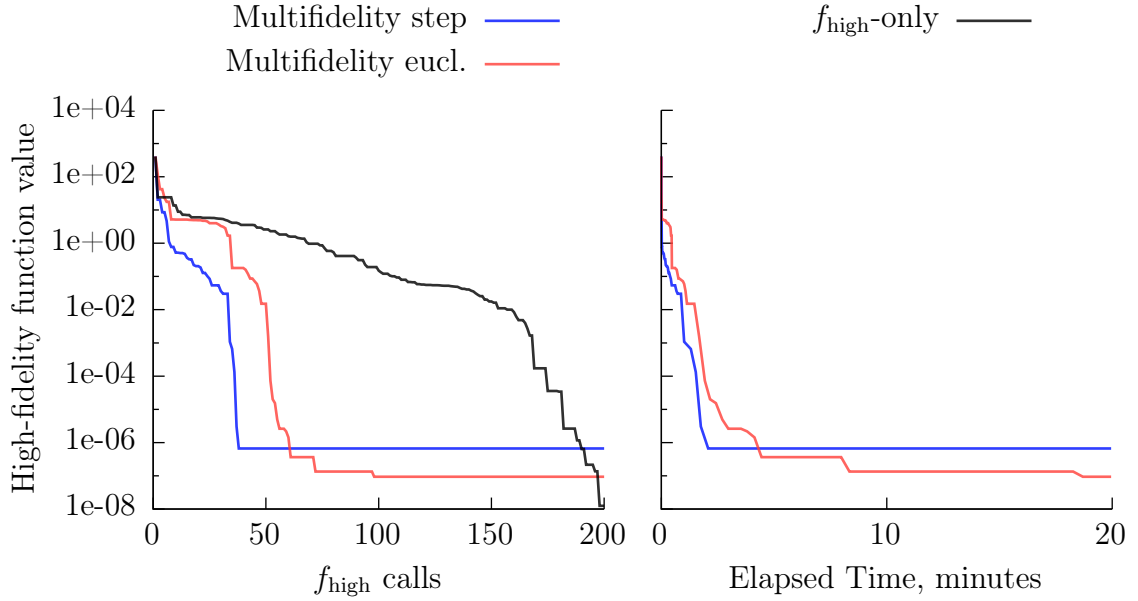


Figure 5.4: Convergence history of the two-dimensional high-fidelity-only and multi-fidelity optimisations (step and Euclidean) in function of the number of high-fidelity function on the left and in function of the time in the right.

fidelity function are very fast to compute, and are not considered *expensive* functions since their computational time is negligible. The overhead time induced by the multifidelity method to build the surrogate (including point sorting and duplicated points removal) and to perform the suboptimisation prevents any wall-clock time comparison against the high-fidelity-only optimisation.

The Rosenbrock function is difficult to optimise efficiently because of the shallow curved valley that leads to the optimum, shown in figure 5.5. The figure shows as well the low-fidelity function and its optimum: the direction is correct on the valley left side (relative to the low-fidelity optimum) but gets inverted in the right hand side. The optimisation paths are shown for each method: the faster convergence from the multifidelity comes from the help of the low-fidelity function that points in the right direction from the beginning whereas the high-fidelity-function-only optimisation struggles in the first iterations to find the right direction. The step-based optimisation path explores the high-fidelity function once on the low-fidelity optimum whereas, the Euclidean trust region optimisation, being more local, explores and corrects the low-fidelity function first and then follow the valley to the high-fidelity optimum.

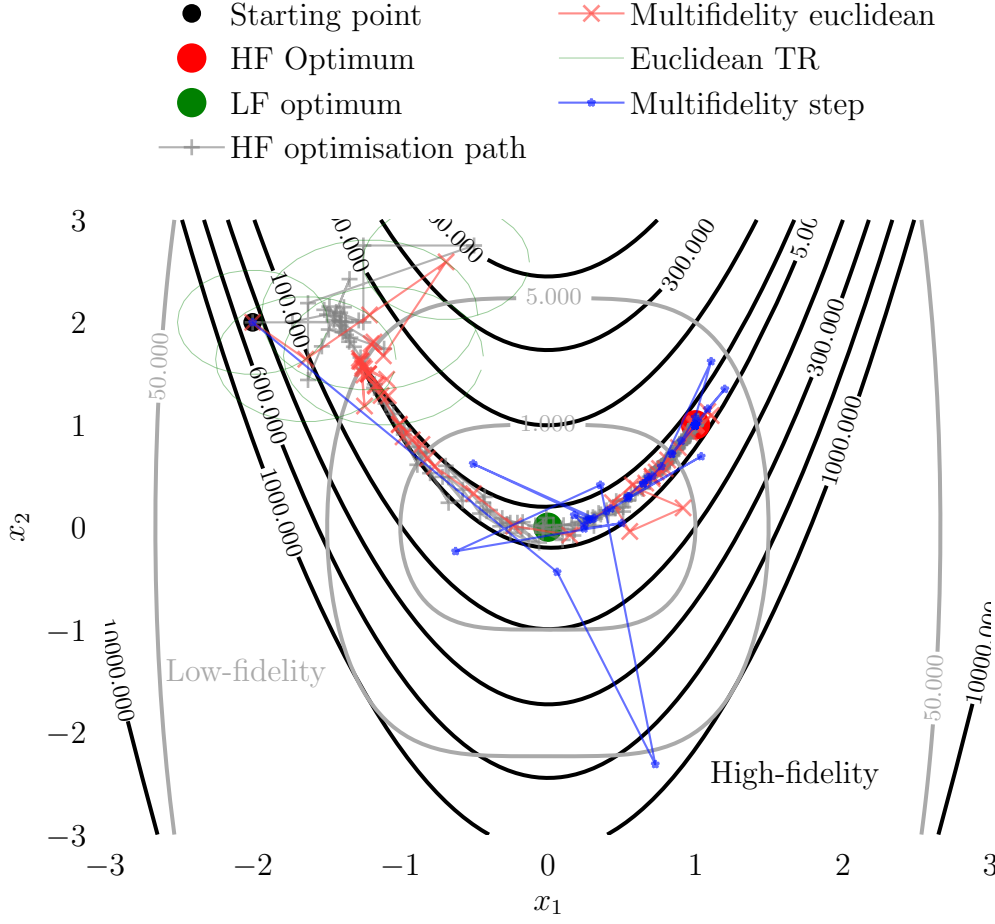


Figure 5.5: Two-dimensional optimisations path for the high-fidelity-only and the multifidelity optimisation (step and Euclidean); the Euclidean trust region is shown for the first 5 iterations.

Table 5.1: NACA 23012 optimisation parameters from (Jarrett and Ghisu 2012).

Parameters	Notation	Value
Reynolds number	Re	$1.8 \cdot 10^6$
Mach number	M	0.18
Fixed Angle Of Attack	α	4 [deg]

5.4 Single Element Aerofoil Test Case

Rising in complexity, the framework is first applied to a simple aerodynamic case from the literature (Jarrett and Ghisu 2012): the lift-over-drag ratio optimisation of a single-element aerofoil at low-speed. First the optimisation problem is presented along with the aerodynamic tools used in section 5.4.1. Then results are shown in section 5.4.2, first comparing the convergence against the high-fidelity-only optimisation, then showing the effect of the initial trust-region size.

5.4.1 Problem Formulation

The objective is the lift-over-drag ratio L/D maximisation of a NACA 23012 aerofoil at low speed with the flow conditions presented in table 5.1. The parametrisation is done with a Free Form Deformation (FFD) method (Sederberg and Parry 1986), the grid is made of 6 points on the x axis and 2 points on y axis, as presented in figure 5.6. The first and last two points are fixed to keep the chord constant and to prevent any angle of attack modification. Hence, 16 design variables are used to deform the initial shape. The quarter- and the half-chord thicknesses are constrained (minimum equal to the baseline thickness) to prevent the aerofoil thinning.

High-fidelity function The high-fidelity estimation is performed using the commercial code Fluent (ANSYS, Fluent 14.5 2013b) and its associated mesher ICEM-CFD (ANSYS, ICEM CFD 14.5 2013) from ANSYS. The geometry is relatively simple but may suffer large variation of shape. Hence the meshing philosophy retained is a C-grid structured mesh that is regenerated for every new candidate. The mesh is around 30,000 cells including a fine boundary layer resolution ($y^+ = 1$). The flow is considered fully turbulent. Lift-over-drag comparison between CFD and wind tunnel (Abbott, Von Doenhoff, and Stivers Jr 1945) is in figure C.1 (appendix C.1): the high-fidelity function represents accurately the wind tunnel experiment, but may

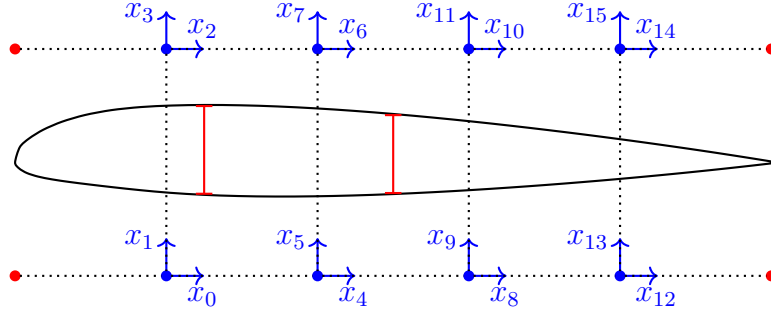


Figure 5.6: NACA 23012 FFD parametrisation: red dots are fixed FFD control points and blue ones are the design variables. The thickness constraints are shown in red.

Table 5.2: Multifidelity parameters for the NACA test case.

Parameter	Min/shrink	Initial/Value	Max/grow
Eucl. trust region size	0.1	0.1 to 1.0	5.0
Step trust region size	2	2 to 70	inf.
Eucl. TR ratio	2.0		1.5
Step TR ratio	2		2
Number of DVs		16	
Correction surrogate		RBF	
Suboptimiser		Tabu Search	
Low-fidelity		XFOIL	
High-fidelity		ICEM & Fluent	

not be accurate near maximum lift-over-drag conditions. The transition was free in the experiment, which may explain the discrepancies.

Low-fidelity function The low-fidelity analysis is performed with XFOIL (Mark Drela 1989), presented in chapter 3, section 3.1.4, with a transition location fixed at $x/c = 0.3$.

Subproblem An RBF surrogate model is used for the correction and the Tabu Search optimiser is used in this test case.

The main parameters for the multifidelity optimisation are summed up in the table 5.2.

5.4.2 Results

The original Tabu Search algorithm performs a full Hook & Jeeves move at each iteration (Glover and Laguna 1997) which requires $2 \times \text{number-of-variables}$ function evaluations. Consequently, each iteration requires a large number of function evaluation. The implementation used in this work reduces this high computational cost by evaluating only a random selection of those moves. This, however, introduces a stochastic behaviour into the optimiser. Moreover, diversification and intensification moves use randomness for the selection of their respective regions. The optimisation is therefore performed once with the high-fidelity-function only with a sufficient number of iterations to remove this stochastic effect from the Tabu Search. The left side of figure 5.7 shows the averaged convergence history from the two trust regions implementations: the one using the Euclidean distance, marked as *Eucl.*, and the one using the number of improvements, marked as *Step*. As concluded in Jarrett and Ghisu (2012), the convergence is faster by a factor of two for a fixed computational cost of 500 iterations. The two different trust region implementations have the same behaviour but the Euclidean implementation is more accurate.

Unlike the results presented in (Jarrett and Ghisu 2012), the high-fidelity-only optimisation is left running for 2000 function evaluations. After 700 function evaluations, the high-fidelity-only optimisation decreases more the objective function than the multifidelity method. The best value from the high-fidelity-only optimisation is compared in table 5.3 along with the time estimation for the whole optimisation: the improvements are also visible in terms of time. Even though the final optimised value is lower than the optimum found by the high-fidelity-only optimiser, the shapes shown in the right side of figure 5.7 are very similar: the camber and the leading edge radius are increased to increase the lift. The main trends are captured by the multifidelity optimiser. The same optimum geometry are obtained as in (Jarrett and Ghisu 2012). It is important to note also that the new optimum geometry may not be robust with respect to small condition variations such as the angle of attack. A multipoint optimisation should be performed to ensure a robust optimum; this would remove the specific waves on the aerofoil lower surface.

A study on the influence of the initial trust region size is performed. Figure 5.8 presents the mean history convergences, based on 10 identical runs, for 4 initial trust region sizes for the two trust-region implementations; the standard deviation is as well shown. Both implementations present a relatively high variability in the convergence: the fastest and the slowest convergences are different in terms of number of high-

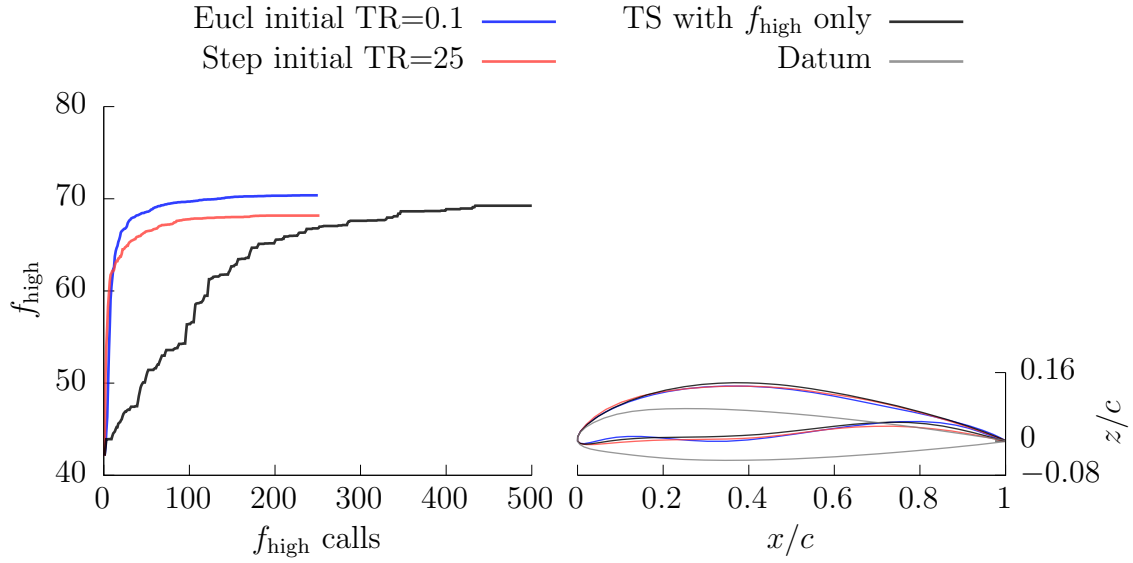


Figure 5.7: Mean and standard deviation convergence history for one initial trust region (TR) size using Eucl. and step trust region management compared to f_{high} -only optimisation on the left and optimised geometries on the right.

Table 5.3: Comparison between high-fidelity-only and multifidelity optimisations for the NACA test case. Multifidelity values are averaged over 10 runs.

	Initial	High-Fidelity	Multifidelity	
			Eucl. TR	Step TR
L/D	42.13	72.14	70.41	68.81
Time (h)		48.1	13.2	13.5

fidelity function calls but also in terms of accuracy. Jarrett and Ghisu (2012) showed also that the multifidelity was increasing the variability in convergence compared to the high-fidelity-only optimisation. This is a drawback of using a stochastic optimiser but thanks to the non-deterministic behaviour, the global optimum is more likely to be found.

Unlike the one-dimensional example, the Euclidean trust region is insensitive to its initial value which is not true for the step one. The smallest trust region leads to a slower convergence (but still faster than the high-fidelity-only) since the high-fidelity function is evaluated much more often than with a higher initial trust region. Conversely, using the largest initial trust region leads to the quickest convergence: this happens because the low-fidelity function is accurate relative to the high-fidelity one and therefore provides accurate direction, despite the low level of correction in the first iterations. Finally, the trust region insensitivity to its initial size is a hint that the function may have a single optimum. The Euclidean trust-region was not used in (Jarrett and Ghisu 2012), nor the effect of the initial trust region size analysed; hence the conclusions drawn here cannot be compared to previous work and is considered as an original analysis.

Figure 5.9 shows the trust region convergences for both trust region definitions with various initial sizes and confirms that the Euclidean trust region is not sensitive to its initial size: all trust regions quickly converge to zero meaning that most of the improvements are done at first (as shown in figure 5.8), albeit small initial sizes lead to a trust region size increase around iteration 25, explaining the higher variance for small trust regions in figure 5.8. The step-based trust region behaviour is different: small initial trust regions fails to find improvements (trust regions increase) while larger ones decrease from first iterations at the expense of reaching a lower value latter. Overall the variance observed in figure 5.8 is similarly found in the trust regions convergence. Finally, as for the Euclidean one, the trust regions convergence to zero confirming the convergence observed in figure 5.8.

5.5 Multielement Aerofoil Academic Test Case

As presented in chapter 1, the multielement aerofoil optimisation is of interest both in the industrial and the academic world. Applying the multifidelity methodology to such difficult optimisation case is one of the objective of the thesis. Hence, now that the framework has been demonstrated on simple mathematical and aerodynamic

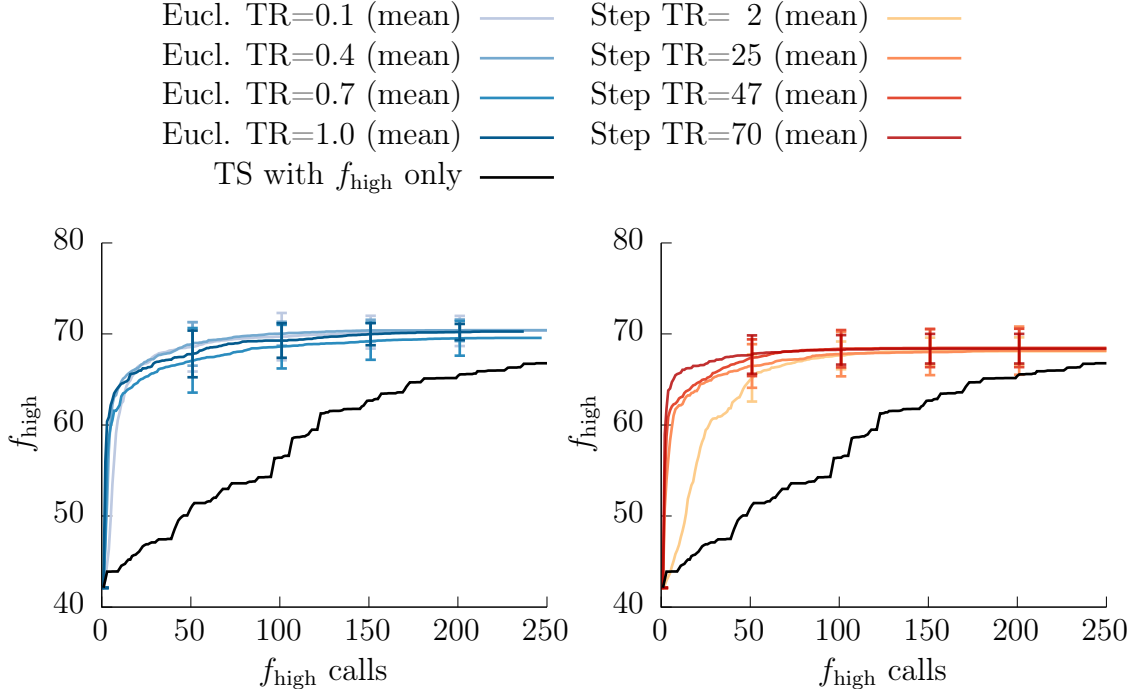


Figure 5.8: NACA multifidelity optimisation convergences for various initial trust region sizes compared with the high-fidelity-only run. The multifidelity runs are averaged over 10 runs and the standard deviation is plotted.

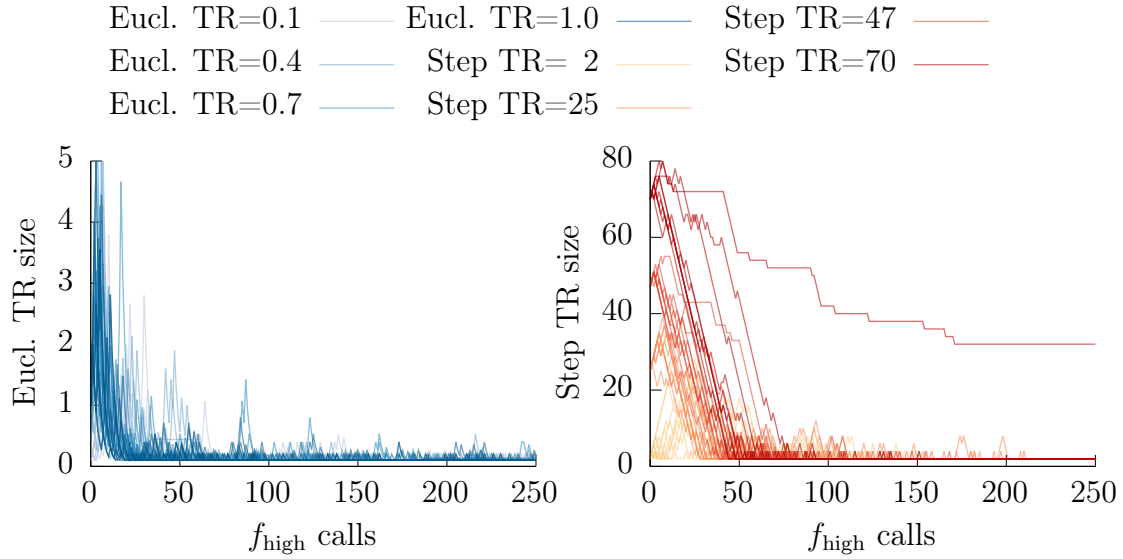


Figure 5.9: NACA trust region size convergence for the Euclidean (left) and step (right) trust regions.

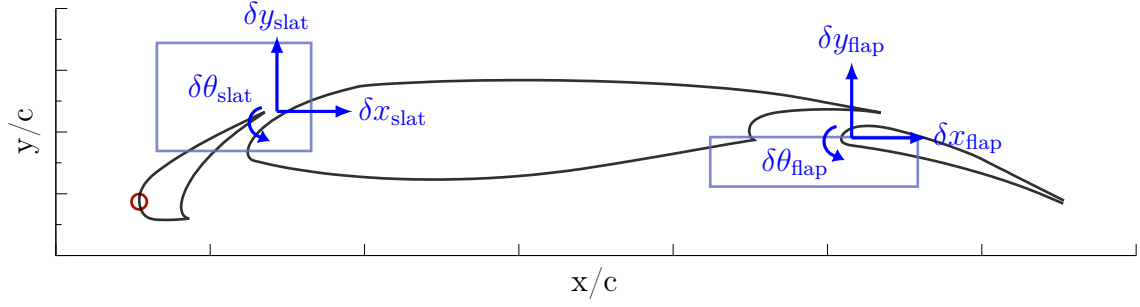


Figure 5.10: Two-dimensional GARTEUR aerofoil geometry at take-off with the flap and slat parametrisation (including optimisation boundaries), oriented in positive values.

test cases, it is applied on a 3-element aerofoil with the tools available at the university. Section 5.5.1 introduces the geometry and the optimisation set-up. Then section 5.5.2 presents the aerodynamic tool used for the optimisation along with the CFD validation. Finally, section 5.5.6 deals with the results and draws conclusion.

5.5.1 Geometry, Parameterisation and Objective

The multielement geometry used in this part is the GARTEUR aerofoil (Thibert 1991). Used during an intensive test program involving Airbus, this geometry has been subject to a lot of studies, including intensive wind tunnel validations. Based on the Airbus A310, the geometry is a realistic and relevant test case for high-lift devices aerodynamics. The geometry representing the take-off configuration is used as a baseline and is plotted in figure 5.10.

We are focusing on the position optimisation of the high-lift devices: the aerofoils shape are considered fixed. The Cartesian parametrisation is used for both the slat and the flap as shown in figure 5.10, oriented in positive values. During deployment, element clashing is checked with a sweep line algorithm (Shamos and Hoey 1976) and clashed configuration are considered unfeasible. The optimisation boundaries, plotted in figure 5.10, are selected such that the kinematics that support and deploy the devices remain feasible. Despite not being the subject of the thesis, section 10.2 shows an example of an optimisation including kinematics.

As described in chapter 3, the lift-over-drag ratio is of interest for positioning the high-lift devices at take-off. Hence the single objective f_{so} is the multielement aerofoil

Table 5.4: High-lift optimisation aerodynamic parameters.

Parameters	Notation	Value
Reynolds number	Re	$4.10 \cdot 10^6$
Mach number	M	0.20
Fixed Angle Of Attack	$\alpha_{cl_{\max}}$	24 [deg]

lift-over-drag ratio:

$$f_{so} = -\frac{L}{D}. \quad (5.11)$$

The flow conditions for the optimisation are the same as for the wind tunnel tests done in the ONERA F1 wind tunnel (Manie, Piccin, and Ray 1989), shown in table 5.4. The angle of attack is fixed and set to the angle at maximum lift for the initial take-off configuration, estimated in section 5.5.2.

5.5.2 High-Fidelity CFD Solver

The commercial tool ANSYS, Fluent 14.5 (2013b) is used as the reference tool (in terms of accuracy) for the academic study, along with its associated mesher ANSYS, ICEM CFD 14.5 (2013). As described in section 3.1, multielement aerofoils are complex geometries, even in two-dimensions. Furthermore, the movement from the slat and flap induces large distance variation between elements: this prevents the use of traditional mesh deformations and complicates the automatic meshing procedure. Lastly, the flap and slat can move freely and may be very close to the main element; this can cause problem for an automatic mesher to create a regular boundary layer of good quality needed for an accurate RANS simulation. Therefore an unstructured mesh is used and, to reduce the chance of failure during the meshing procedure, the Fluent *Enhanced Wall Treatment* is used to blend automatically the viscous layer model according to the y^+ value (ANSYS, Fluent 14.5 2013a). Hence the mesh is designed to have a y^+ of the order of 100 and the logarithmic formulation is used to resolve smaller scales. The grid domain starts 20 chords upstream the slat leading edge and extends to 50 chords downstream with 30 chords above and below to ensure free stream conditions at the boundaries. This leads to a relatively coarse mesh of 280,000 elements. It is important to recall that for optimisation, the CFD is used to compare configurations relative to the baseline as opposed to provide absolute accurate values; thus to alleviate the optimisation clock time, a coarse mesh is used. A grid convergence analysis is performed and is shown in figure C.2 (appendix C.1):

increasing the number of elements reduces the error, but for an optimisation, the medium-sized mesh is of sufficient quality; this mesh will be the same throughout the thesis.

RANS simulations are the industry workhorse (Rumsey and Ying 2002) to obtain accurately and relatively quickly the aerodynamic performances. Those equations, along with a Spalart-Allmaras turbulence model (J. Wild 2008), are used. An upwind second-order solver is used for all equations from iteration 500 to reduce divergence at the beginning. The flow is considered converged when residuals fall below 10^{-5} , or if the number of iterations is greater than 2000². Because convergence is not always achieved, the lift and drag coefficients are averaged over the last 100 iterations. The RANS simulation is run on 8 cores during 722 wall-clock seconds, or 12 minutes and 13 seconds.

The simulation with the initial geometry is validated with experimental data from the ONERA F1 wind tunnel (Manie, Piccin, and Ray 1989), with the conditions shown in the table 5.4. Firstly, the integrated values are compared for angles ranging from 0 to 27 degrees in figure 5.11. The difference between maximum lifts and angles of attack at maximum lift is relatively small, and before stall, the lift curve fits experimental data; only after stall, CFD deviates as expected due to the separation that is source of numerical error in RANS solution. Although the drag polar presents discrepancies, the general trends are respected, until separation occurs. The discrepancies are most likely coming from the coarse discretisation in the wake. Secondly, the pressure coefficients are compared to experiments for two angles: 12.2 and 20.4 degrees in figure 5.12. For the lower angle, the match is perfect for all elements, confirming the good lift match before separation. However, the higher angle of attack presents a lower suction peak on the slat leading edge but without consequences on the other elements. This explains the relatively delayed stall prediction from CFD.

5.5.3 Low-Fidelity CFD Solver

The MSES software from Professor Drela, presented in section 3.1.6 (chapter 3), is used as the low-fidelity estimator.

When used in challenging aerodynamic cases, such as high-lift conditions, a single point evaluation requires up to 2 minutes on a single core. Massive separation, that can easily appear from bad designs generated during optimisation, produces diver-

²This may be not enough to have fully converged flow but is sufficient to capture most of the flow physics

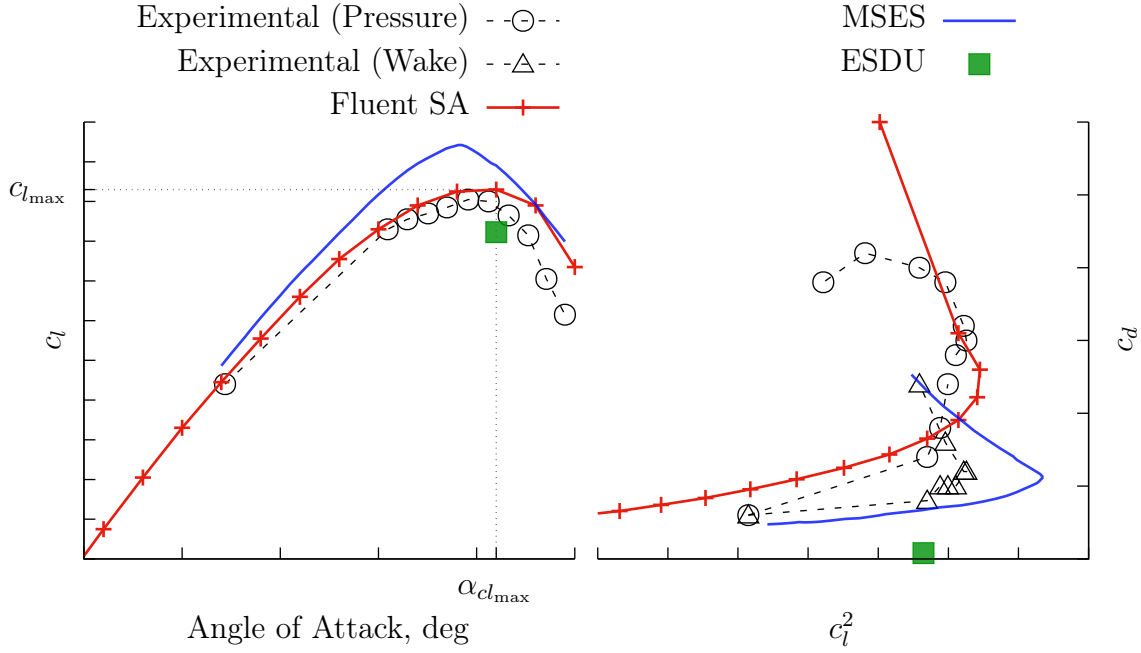


Figure 5.11: Polar comparison between experimental data, Fluent, MSES and ESDU simulations for the Garteur test case.

gence in the coupling procedure. This phenomena is amplified by the discretisation quality used to solve the Euler equations. Despite the attention given to the mesh points distribution on the profile, divergence occurs for bad designs resulting in a restricted design space.

Figure 5.11 includes as well the polar computed by MSES for the baseline configuration. MSES over-predicts the maximum lift value although the maximum lift angle is well estimated. Finally, the drag is slightly under-estimated but presents the same trends. The trends are respected but the absolute values are not as accurate as our RANS reference method. It will therefore perfectly play its low-fidelity tool role.

5.5.4 Very Low-Fidelity CFD Solver

To reduce the time needed by the low-fidelity estimator, the ESDU datasheets (see chapter 3, section 3.1.7) are implemented to predict the maximum lift and its associated drag. Although the time required for this estimation is very small (of the order of 500 ms), leading to a large reduction in the subproblem resolution time, the method applicability is restrictive, as it relies extensively on experimental carpet

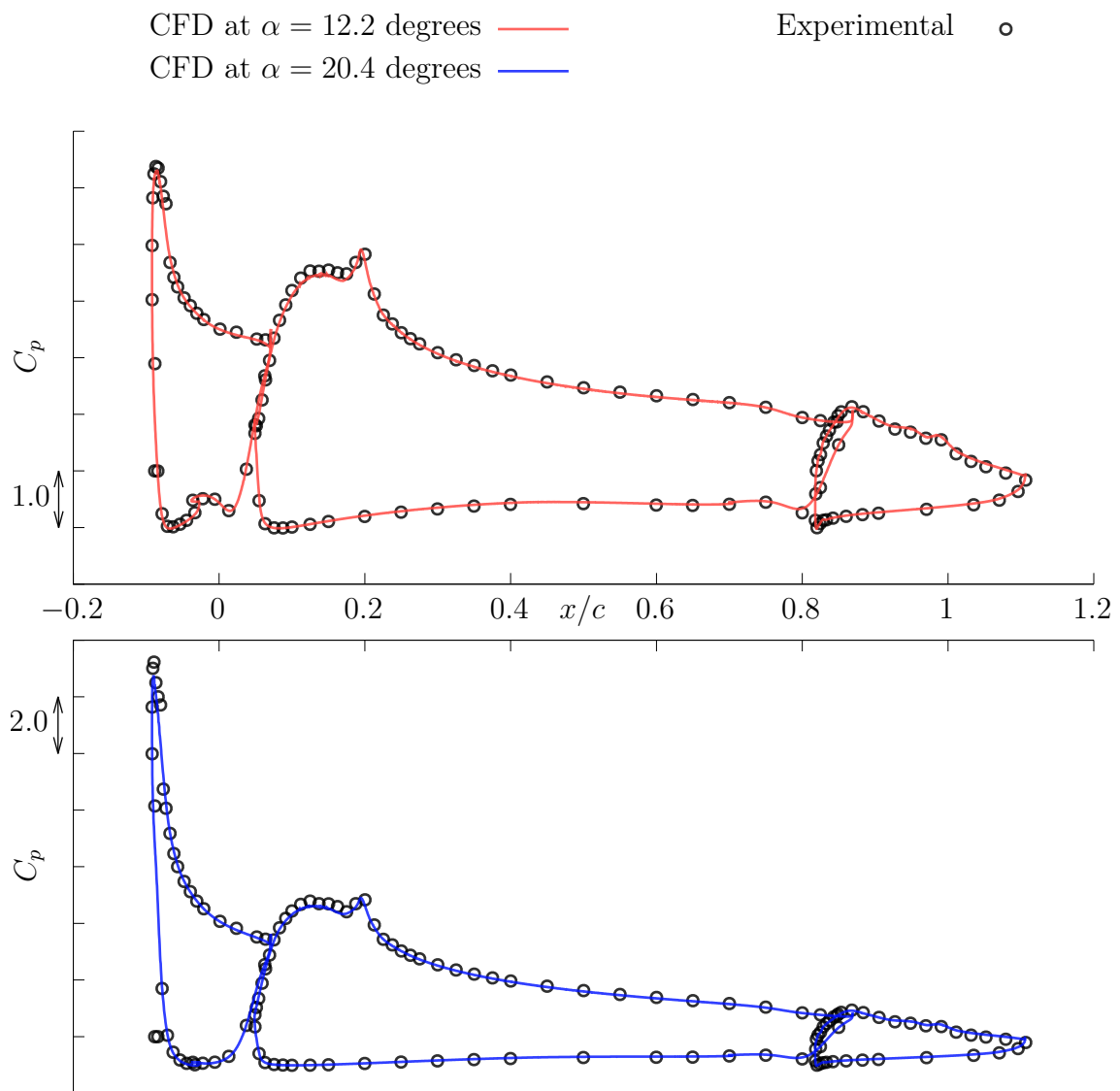


Figure 5.12: Pressure coefficients distribution from CFD on the GARTEUR aerofoil in take-off configuration for two angles of attack and comparison with experimental data.

Table 5.5: Multifidelity parameters for the Garteur single-objective test case.

Parameter	Min/shrink	Initial/Value	Max/grow
Eucl. trust region size	0.1	2.5	5.0
Step trust region size	2	6	inf.
Eucl. TR ratio	1.5		2.0
Step TR ratio	2		2
Number of DVs		6	
Correction surrogate		Kriging	
Suboptimiser		Tabu Search	
Low-fidelity		MSES / ESDU	
High-fidelity		ICEM & Fluent	

plots. Outside its design bounds, the method provides either incorrect trends, or no trends at all. The estimated maximum lift and corresponding drag value are plotted in figure 5.11. This method clearly underestimates both maximum lift³ and the corresponding drag value to the advantage of having a negligible run time.

5.5.5 Suboptimisation

Multiple tests were done and the RBF model used for the correction showed shortfall in terms of accuracy and optimum consistency as shown in Demange, A. Mark Savill, and Timoleon Kipouros (2016b). Therefore, the more advanced Kriging model is used to provide the correction. The same Tabu Search optimiser is used as in the previous test case. The main multifidelity parameters for this test case are summed up in table 5.5.

5.5.6 Results

Due to the stochastic nature of Tabu Search, 10 optimisations using only high-fidelity function calls are performed to establish a comparison. The convergence history in terms of number of high-fidelity function calls is shown in figure 5.13a (left) and in terms of wall-clock time in figure 5.13a (right); only the mean convergence μ_r and the convergence standard deviation σ_r^2 are showed instead of the 10 individual runs. The convergence is compared to the multifidelity cases: each trust region

³The method implemented does not provide an estimation of the angle at maximum lift, but estimates the maximum lift coefficient possibly achievable from the configuration.

method (Euclidean and step) is tested for both low-fidelity tools: the ESDU method in figure 5.13a and the MSES method in figure 5.13b; multifidelity optimisations are as well run 10 times. Although the single-fidelity optimisations show different convergences, they all reach the same optimum, but the number of CFD calls and therefore the time to reach the optimum can varies from 10 to 55 hours.

Regardless of the low-fidelity tool used, the Euclidean trust region method provides more consistency in the convergence than the step-based one. Using either MSES or ESDU tools, the Euclidean trust region optimisations always converge to the same value, similar to the high-fidelity-only one, whereas the step-based trust region optimisations always find a relatively worse optimum with a bigger range of variability in its final value. Furthermore, the number of CFD calls needed for convergence remains in the same range for the Euclidean trust region method unlike the step-based one. The same comment can be made on the elapsed time.

We can now compare figure 5.13a and 5.13b: the optimisations performed with MSES are more consistent than the one with ESDU, especially for the step-based trust region. Observing the graphs on the left, it seems that MSES allows the step-based trust region optimisations to be faster than the high-fidelity-only ones but because of an increased time spent in the suboptimisation, this conclusion cannot be drawn on the elapsed time (right graph). However, using MSES over ESDU allows the Euclidean based trust region to be faster than the high-fidelity-only optimisation, despite a longer low-fidelity run time; the increased accuracy improves the suboptimisation predictions reducing high-fidelity function calls need. Finally, using MSES reduces the differences in convergence for both the Euclidean and the step-based trust region methods to the point where the multifidelity convergences show less variability than the high-fidelity-only ones.

Differences in the optimum objective value are seen between the methods, but are those differences coming from a different slat and flap placement or is it only due to the numerical noise? To answer this question, the pressure coefficients and the best optimum geometries for each run are plotted in figure 5.14. All optima reduces greatly the suction peak from the slat, reducing its generated lift, in order to obtain a stronger acceleration on the main element leading edge; the slat wake is as well reduced, decreasing its drag. The deceleration are identical between each configuration, but despite having large discrepancies in the flap position, the deflection angle is always reduced to keep the pressure coefficient distribution similar to the datum. Finally figure 5.14 shows as well the lift and drag coefficients for each configuration. All

configurations had their lift reduced, especially the high-fidelity-only one. The drag as well is reduced but all multifidelity optimisations remain close to each other whereas the high-fidelity-only optimisation is slightly more offset towards the lower lift and drag region.

To conclude the single objective optimisation Garter test case, the multifidelity framework allows a more consistent and occasionally faster optimisation than using only the highest fidelity. Increasing the low-fidelity model accuracy, despite the increased computational time for a single run, allows a decrease of the overall optimisation time since the optimiser will rely more on the physics than on the correction model. From the two trust regions presented, the Euclidean trust region provides the most consistent and accurate optimum but the step-based trust region could be faster if less time was spent in the suboptimisation.

The multifidelity optimisation benefits are not significant when applied to a multielement position optimisation compared to the previous case. The increase in aerodynamic modelling complexity is first found on the function responses that are noisier and less similar but also in the low-fidelity accuracy-to-speed ratio.

5.6 Multielement Aerofoil Industrial Test Case

The method, as well as the implementation, allows the use of any objective function, optimiser and surrogate model. Hence, the method is as well applied to a test case provided by the industrial partner. The geometry and optimisation set-up are presented in section 5.6.1 followed by the low- and high-fidelity solvers introduction in section 5.6.2 and 5.6.3 and the suboptimisation details in section 5.6.4. The multifidelity optimisation is then done in three steps: first only two variables are used in section 5.6.5, second the small boundaries are used with the six design variables in section 5.6.6 and third the extended boundaries problem is treated in section 5.6.7.

5.6.1 Geometry, Parameterisation and Objective

The geometry, showed in figure 5.15 in the datum position, is more representative (as much as a two-dimensional section can be) of aerofoils used in modern transport aircraft. It features a slat that can move freely, a large recirculation area in the rear of the main element and a relatively simple flap that can as well move freely.

The parametrisation chosen for the slat and flap positioning uses the gap and over-

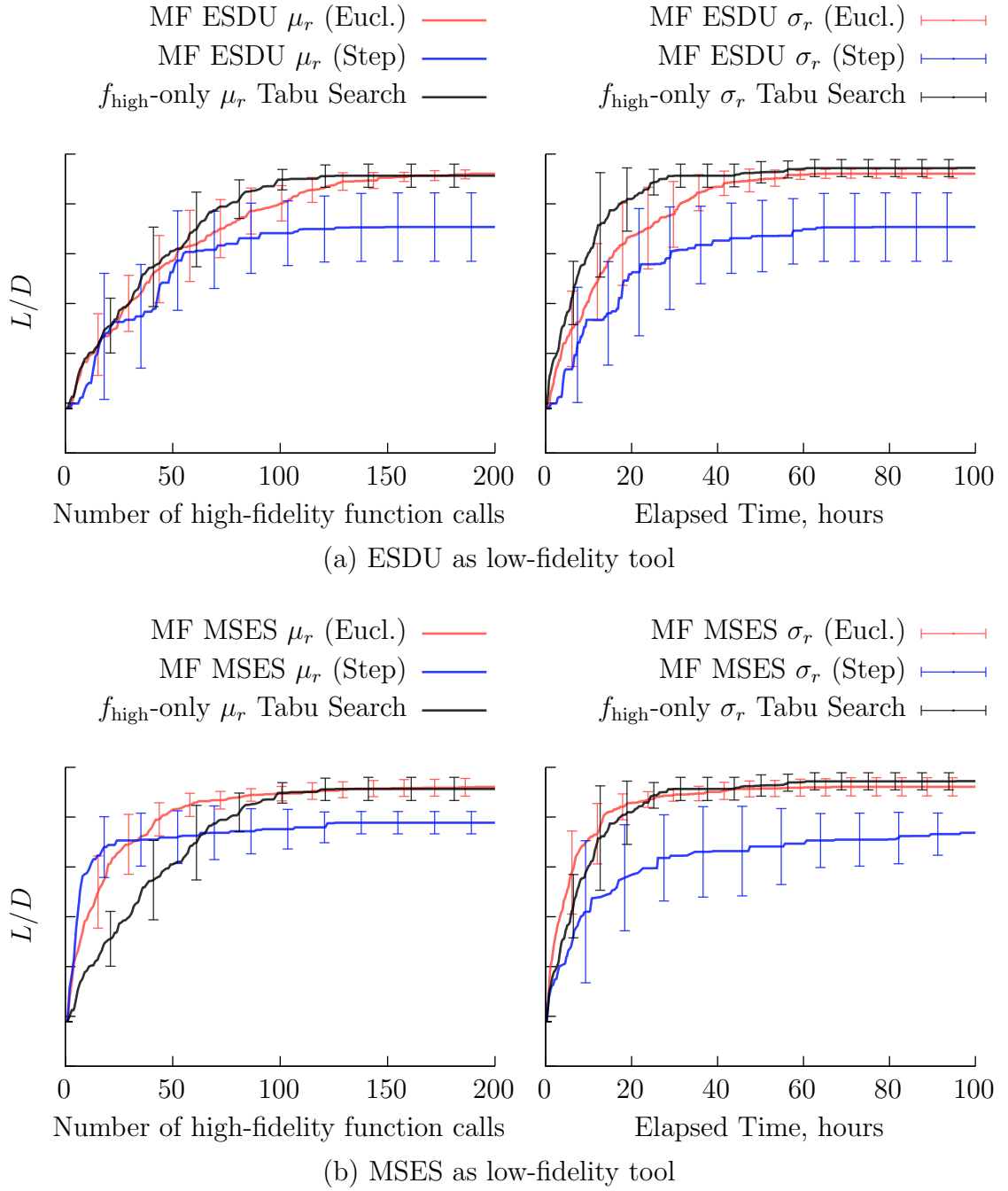


Figure 5.13: Convergence (mean μ_r and standard deviation σ_r) comparison between high-fidelity-only optimisations and multifidelity optimisation using either ESDU (a) or MSES (b) with either an Euclidean- or step-based trust region, in terms of number of high-fidelity function calls on the left and in terms of overall time on the right.

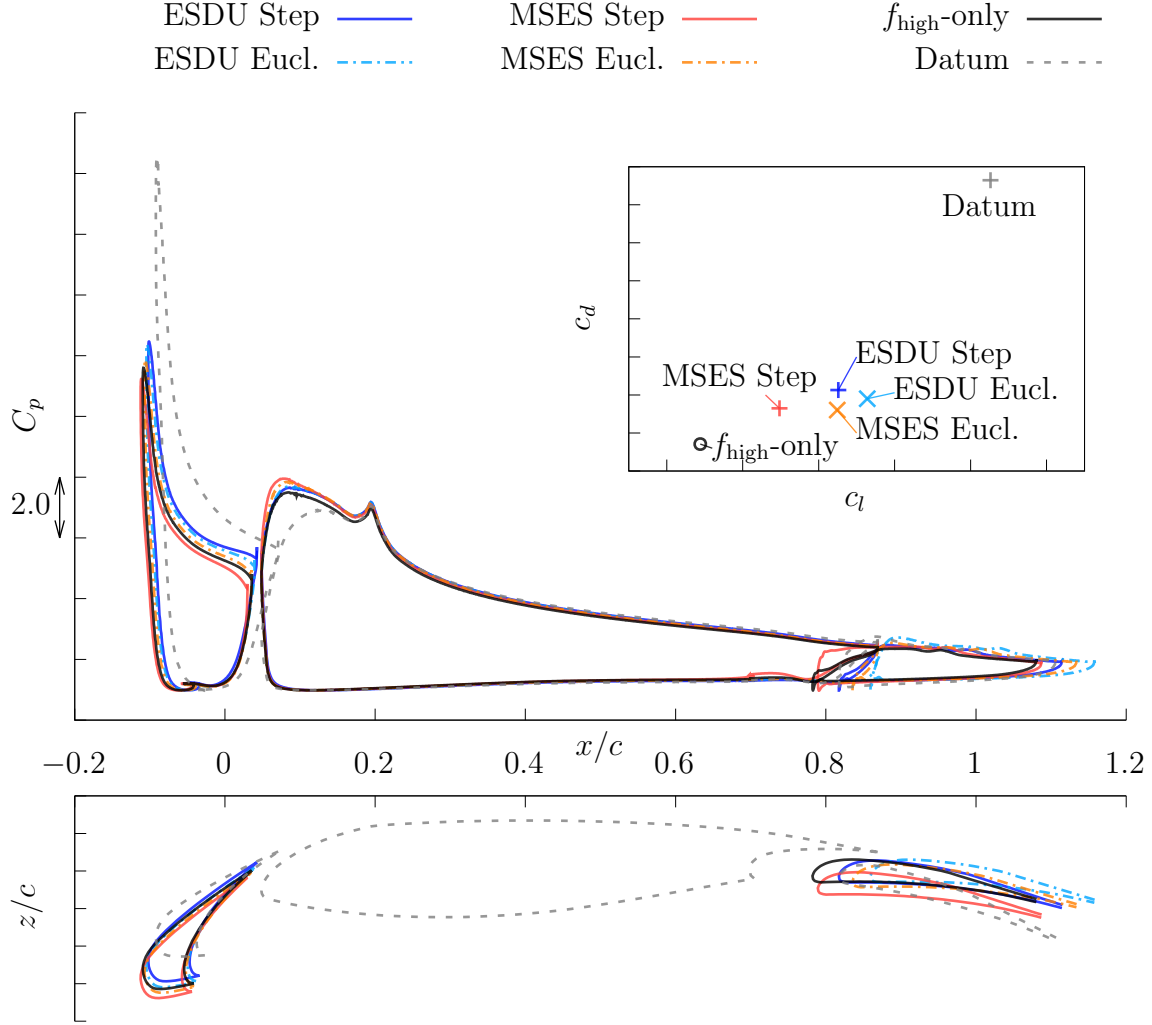


Figure 5.14: Pressure coefficient plots and shapes associated (below) for the multielement optimisations, compared with the high-fidelity-only optimum and the datum configuration. Lift and drag coefficients breakdown is also shown.

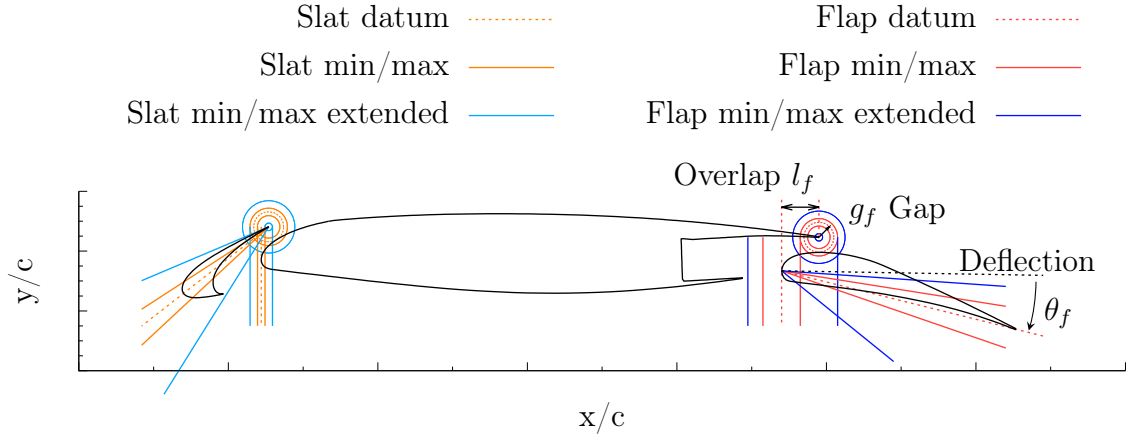


Figure 5.15: High-lift two-dimensional profile used for the industrial optimisation in datum configuration. 2 sets of boundaries, and the parametrisation are plotted.

Table 5.6: Normalized optimisation boundaries values for the industrial test case.

	Slat			Flap		
	Gap	Lap	Defl.	Gap	Lap	Defl.
Min.	-1.0	-1.0	-1.0	-1.0	-1.0	-0.5
Max.	2.0	2.0	2.5	2.0	1.4	3.0

lap definitions shown in figure 5.15; the deflection is defined positive for nose down rotation. The standard workflow for slat and flap positioning at the sponsor involves the low-fidelity tool to set the overlap and gap without running the low-fidelity aerodynamic model. A total of six design variables are used for the optimisation. The initial values are noted g_{s0} , l_{s0} , and θ_{s0} for the slat and g_{f0} , l_{f0} , and θ_{f0} for the flap. The normalised bounds for the optimisation are shown in table 5.6 relative to the starting values, and are plotted in figure 5.15.

Objectives are chosen to be representative of the high-lift design process. If we look at the take-off conditions, Rudolph (1996) highlights that the high-lift system should allow a good compromise between lift capability and lift-over-drag efficiency. For take-off, the Federal Aviation Requirements (FAR) requires the lift coefficient at climb speed to be equal or less than $C_{L_{\max}}/1.14^2$. Only the maximum lift-coefficient is used in this case:

$$f_{so} = -c_{l_{\max}}. \quad (5.12)$$

Table 5.7: High-lift aerodynamic parameters for the industrial test case.

Parameters	Notation	Value
Reynolds number	Re	$6.0 \cdot 10^6$
Mach number	M	0.20
Multiple angles	α	0.0 to 27.0 [deg]

The maximum lift is obtained by computing the aerofoil aerodynamic polar with two points in the linear region and 6 points in the remaining non-linear part. The maximum angle of attack is set to 27 degrees to ensure a complete polar. The remaining flow conditions are similar to the previous test case, representative of a take-off (see table 5.7).

5.6.2 High-Fidelity CFD Solver

The high-fidelity function is chosen to represent the most common model used in industry: solutions to the RANS equations. The unstructured DLR-TAU flow solver (Schwamborn, Gerhold, and Heinrich 2006) is used to solve the equations. Turbulence is modelled with the one-equation Spalart-Allmaras with Edwards modification (SAE) model (Edwards and Chandra 1996). A central discretisation scheme is used along with a multigrid acceleration. The flow is first converged at 0.0 degree angle of attack using different CFL numbers for 10,000 iterations, then each angle is converged for 7,500 iterations from the previous solution; this allows convergence for most angles except the last ones that are fully stalled.

Rather than a mesh deformation, a regeneration procedure is favoured to avoid stretched cells due to the large slat and flap movements. The mesh is made of roughly 200,000 points and it includes a viscous region of at least 30 layers with a y^+ of 1 to properly resolve this region. The mesh is automatically generated from the best practices available at the sponsor, and no validation is therefore added to this section.

A single polar run on the Airbus High Performance Computer (HPC) takes 74,937.62 CPU time seconds on 8 cores. Including transfer times, the polar takes 9,400 wall-clock seconds or 2 hours and 37 minutes.

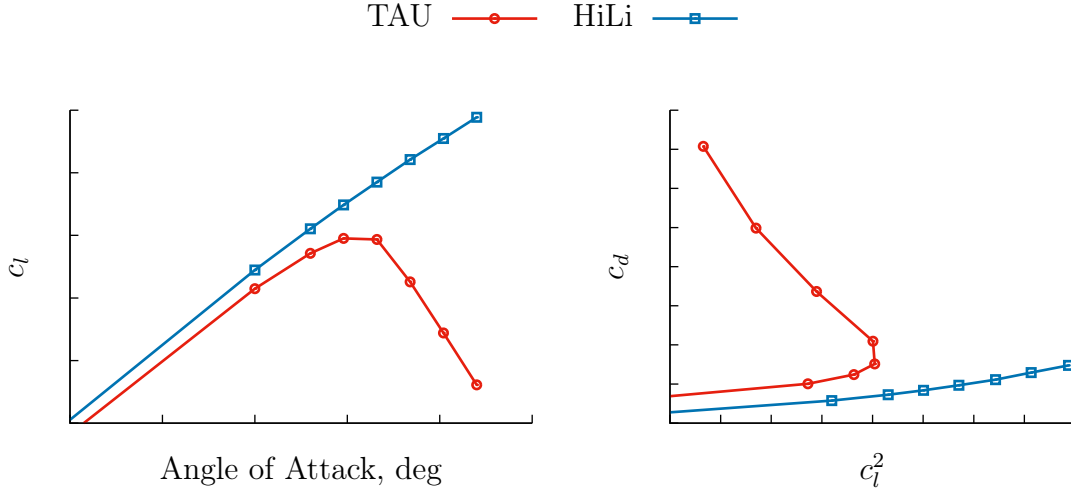


Figure 5.16: Baseline configuration aerodynamic polars for the Airbus test case from TAU and HiLi.

5.6.3 Low-Fidelity CFD Solver

To lower the computation cost at the early design stage, Airbus uses a viscous/inviscid coupled method for the prediction of the high-lift aerofoils aerodynamic performances. The tool is HiLi and uses a panel method coupled with a boundary layer equation (see chapter 3, section 3.1.6 for more details). The run time varies significantly depending on the case convergence. The polar is run on a single core, and takes between 30 seconds to 2 minutes.

The lift and drag polar is compared in figure 5.16 for both TAU and HiLi. The latter remains accurate in the linear region but fails to predict the aerofoil stalling characteristics. We hope to use the higher fidelity TAU solver to correct it.

5.6.4 Suboptimisation

A RBF surrogate model is used to provide the correction implemented in Python this time. Since the suboptimiser is directly called from Python in this implementation, different optimisers can be tested and compared. Hence, this test case uses a Subplex implementation from the NLOPT library. To increase the global exploration, the Particle Swarm Optimiser from the PyOpt library (ALPSO) is also used; both algorithm were introduced in chapter 3. Table 5.8 sums up the different multifidelity parameters used in this test case.

Table 5.8: Multifidelity parameters for the industrial single-objective test case.

Parameter	Min/shrink	Initial/Value	Max/grow
Eucl. trust region size [†]	0.1	0.05	5.0
Step trust region size	2	6	inf.
Eucl. TR ratio	1.2		1.3
Step TR ratio	2		2
Number of DVs		6	
Correction surrogate		RBF	
Suboptimiser		SBPLX / PSO	
Low-fidelity		HiLi	
High-fidelity		Solar & TAU	

[†] in % of design space

5.6.5 Initial 2-Variables Optimisation

The method is first applied on the two-variable problem, presented in this section which leads to the development of a restart strategy. Then the full six-design variables case is run and results presented in section 5.6.6.

The initial tests are performed on a quicker problem than the original one: the flap gap and overlap optimisation with the lift-over-drag ratio as objective. Convergence of the method is shown in figure 5.17. The optimiser quickly gets stuck and stops progressing.

Restart Strategy The commercial toolbox *MACROS* (see chapter 3, section 3.2.4), performing a SBO, is used to serve as reference; its convergence is as well shown in figure 5.17. The multifidelity optimisation is stalling in a different location than the high-fidelity-only one. Similarly to Tabu Search (Jaeggi, G T Parks, et al. 2008), a restart strategy is implemented to escape those “numerical” minima. The number of iterations that do not lead to an improvement is counted and when a specified trigger is reached, the restart move is done. The strategy retained is a restart from the second best optimum: the new suboptimisation starting point is chosen to be the second best optimum encountered so far. The trust-region is also reset to its initial value. This allows the optimisation to start from a known location, and therefore accurate, but with a better correction in its vicinity. The suboptimiser is therefore following a new path and does not find the same local minimum.

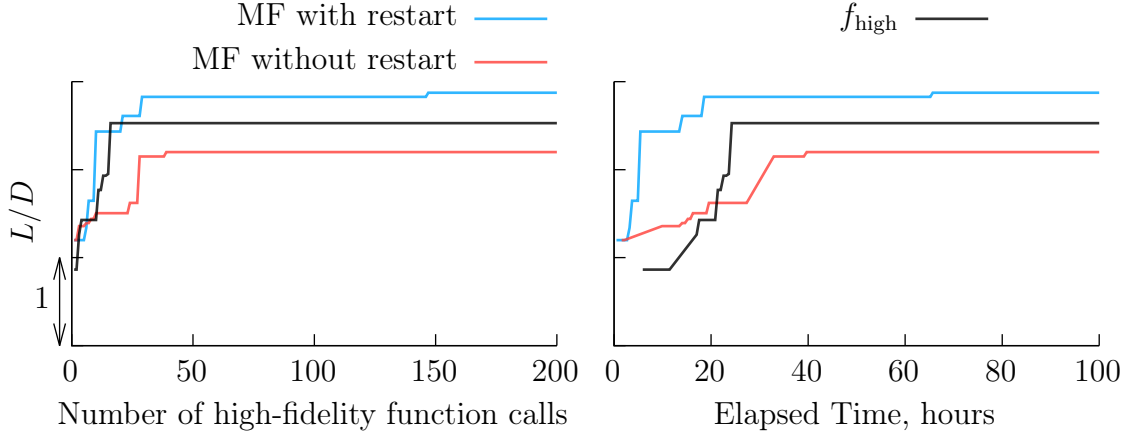


Figure 5.17: Multifidelity convergence on the 2-design variables optimisation with and without restart strategy. Left is the convergence with respect to the number of function calls, right is with respect to the wall-clock time.

Restart visualisation Figure 5.18 shows the same two-dimensional optimisation path (on the design space) after 20 iterations. At point 3, the suboptimiser does not find any improvement and is therefore stuck in one portion of the design space. The optimisation therefore restarts from the second best, and a new path is created (shown in green in the figure 5.18). The convergence is also shown in figure 5.17 with respect to the objective function: as expected, the restart allows a better design space exploration (a better lift-over-drag ratio), and in addition the wall-clock time is lower, since the restart is performed after only 3 iterations leading to a totally new path, following more closely the high-fidelity function.

High- and low-fidelity behaviour In figure 5.19 is shown the two-dimensional problem high- and low-fidelity functions (L/D); the functions output are normalised hence the optimum is toward region of highest values. The high-fidelity function seems to have its optimum in the region of high overlap and low gap (upper left). On the contrary, the low-fidelity function behaviour is opposite and the optimum seems to be in the low-overlap (lower region) with less effect from the flap gap. Our two functions do not agree in this case, leading to difficulties for the multifidelity to efficiently converge to the high-fidelity optimum.

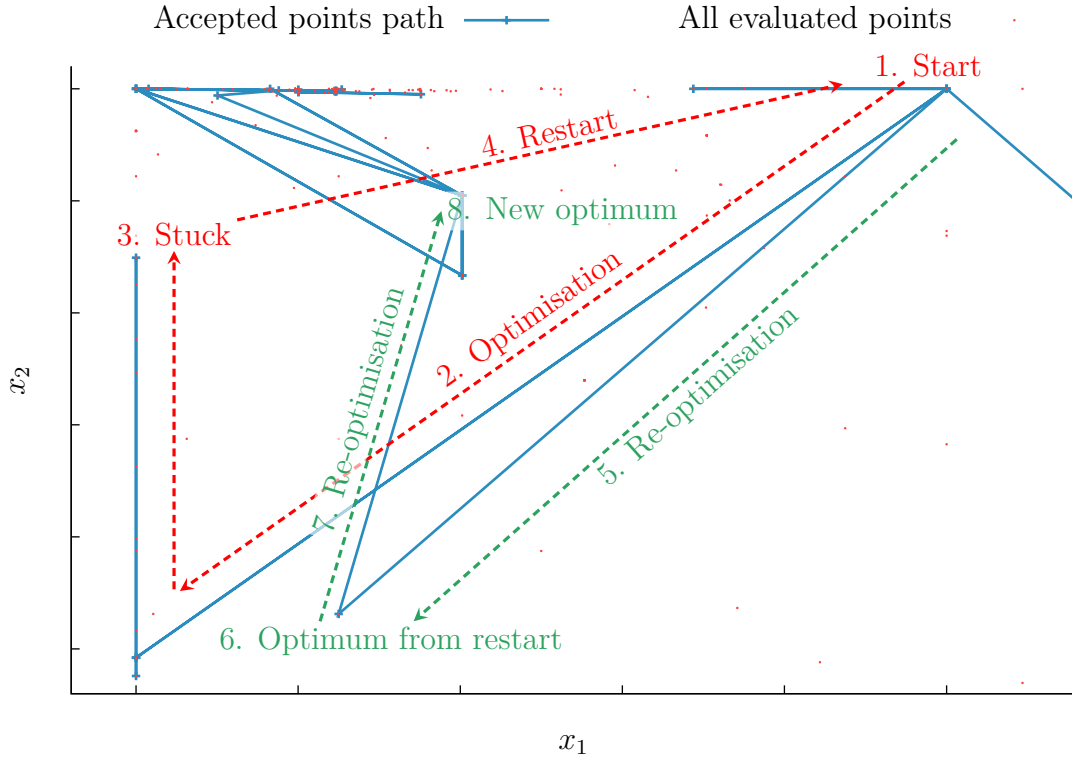


Figure 5.18: Two-dimensional design space showing the path of the multifidelity optimisation.

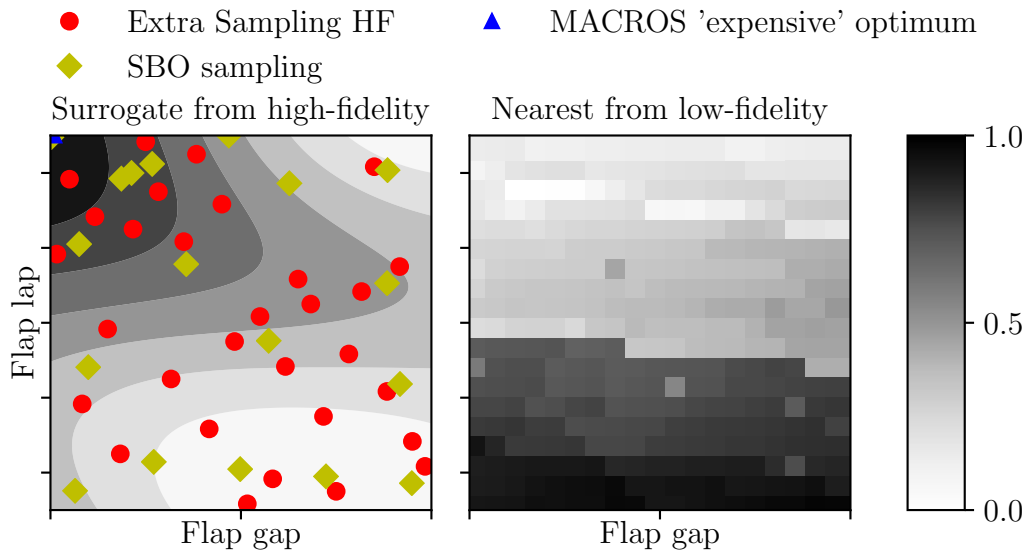


Figure 5.19: Normalised high-fidelity approximation (left) and low-fidelity sampled values (right).

Table 5.9: Optimisation starting and boundaries values for the initial boundaries case, normalised by variable range.

	Slat			Flap		
	Gap	Lap	Defl.	Gap	Lap	Defl.
Min.	0.0	0.0	0.0	0.0	0.0	0.0
Max.	1.0	1.0	1.0	1.0	1.0	1.0
Start	0.5	0.5	0.5	0.5	0.5	0.5

Table 5.10: Running time spent in the suboptimisation and in CFD for both Euclidean- and step-based trust regions using SBPLX optimiser.

<i>All times in minutes</i>		Euclidean	step
Optimisation	mean	134.1	21.2
	standard deviation	77.4	21.1
CFD	mean	219.9	281.4
	standard deviation	195.6	167.3

5.6.6 Results: Small Boundaries

The first 6-variable design optimisation is first performed on relatively restricted boundaries —shown in table 5.9— when the design space region is known and only fine tuning is required.

The two trust region definitions are compared between each other. The *Euclidean* trust region is defined as a ball centred on the starting point in the design space while the *step*-based one counts the number of improvements in the corrected objective space.

The convergence is shown in figure 5.20 both in terms of number of iterations and elapsed time. The first remark to be made is the similarities between the two trust region definitions convergence with respect to the number of iterations. However, in terms of elapsed time, the Euclidean trust region is much slower, confirmed by looking, for each iteration, at the time spent in the suboptimisation and in the CFD for the SBPLX optimisations in table 5.10: the Euclidean-based trust region optimisation spends a much higher time in the suboptimisation than the step-based one since it is exploring more thoroughly the close vicinity of the suboptimisation starting point.

Similar remarks can be made on the optimisations performed with the ALPSO

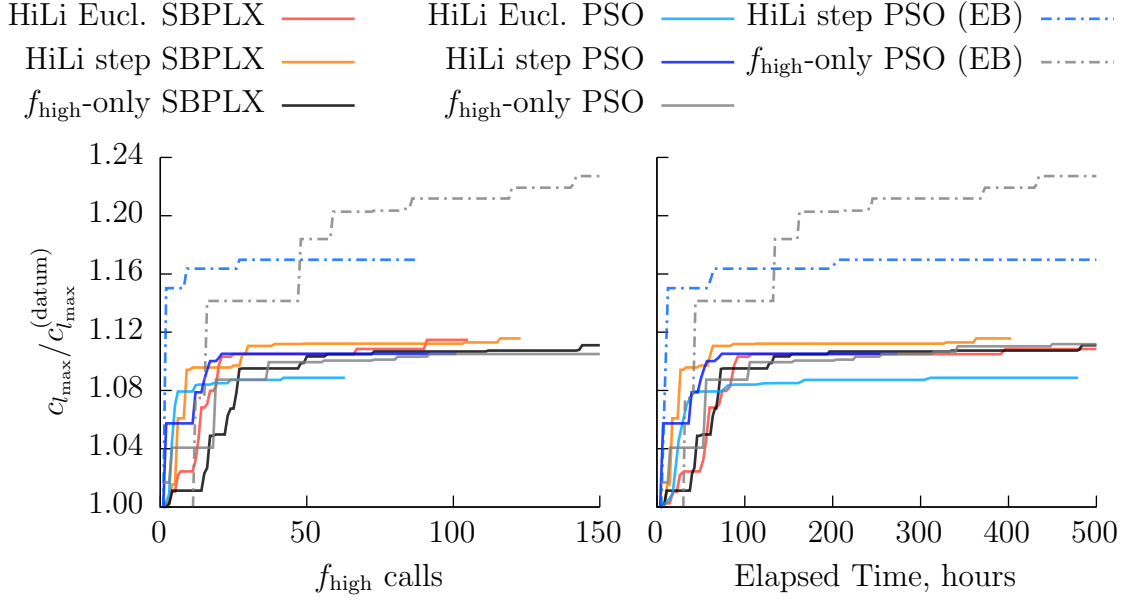


Figure 5.20: Convergence of high- and multifidelity initial, and extended (EB), boundaries optimisations with respect to the number of high-fidelity function calls (left) and in terms of elapsed time (right).

optimiser, displayed as well in figure 5.20. Again improvements in time are seen with the step-based trust-region multifidelity but also with the Euclidean one, albeit it gets stuck at a lower final optimum. The time improvements are slightly less: in the order of 7 days for the step-based method. However, one can observe that the PSO with only the high-fidelity optimisation finds a better optimum than any of the multifidelity optimisations. It seems that the multifidelity limits the global exploration of the design space, even when the trust region is not defined on the design space but on the corrected objective one.

Finally, the optima design variables can be compared for all optimisations presented above; this is done in the parallel coordinates plot shown in figure 5.21, focusing on the initial boundaries runs. Firstly the flap positioning is very consistent across all methods, except the best configuration found by PSO on the high-fidelity function only; this maximisation problem therefore seems to have multiple distant minima. Secondly, although the same trends are also observed on the slat positions, the solutions are more scattered. Only the slat deflection, with a maximum down pitching movement is consistent across all methods in order to enable the maximum energy extraction from the flow. The lower $c_{l_{\max}}$ from the multifidelity Euclidean

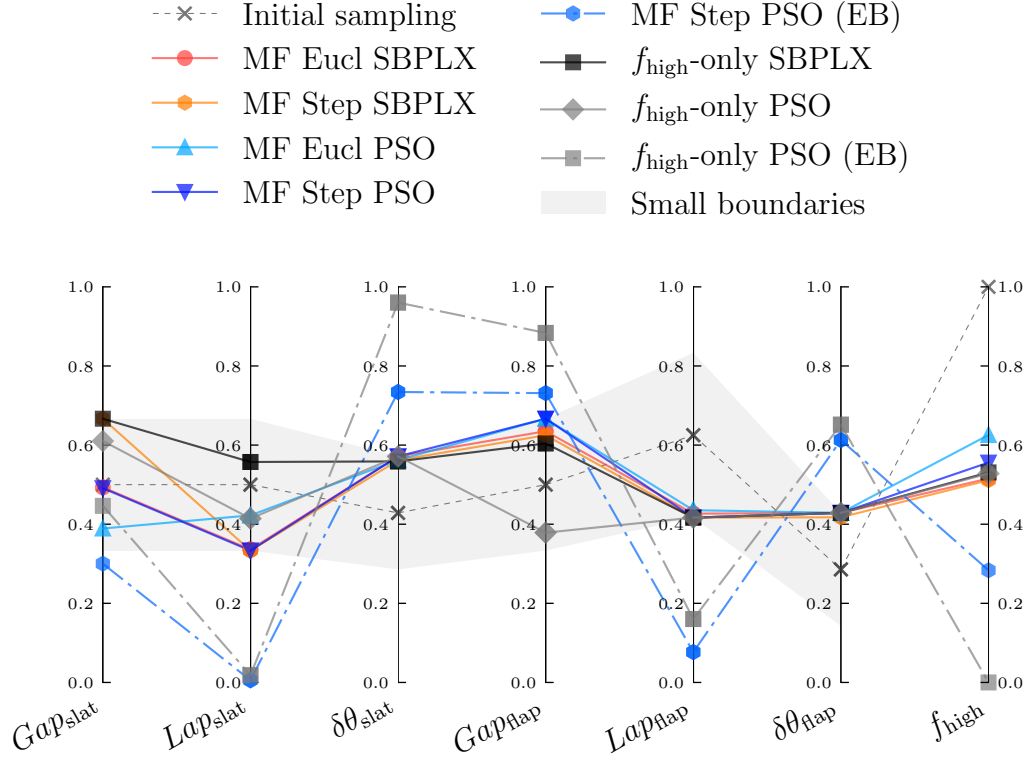


Figure 5.21: Parallel coordinate displaying the optima solutions from multifidelity and high-fidelity-only optimisations for the small and extended (EB) boundaries cases. The ranges are the extended one and the small boundaries are shown shaded.

PSO can be attributed to the lower flap deflection and lower slat gap. Despite the high maximum c_l obtained from the multifidelity Subplex optimisation (probably thanks to a slightly larger flap deflection), their slat gap value is close to the starting point; it pushes us to question this variable sensitivity with respect to the maximum lift coefficient.

5.6.7 Results: Extended Boundaries

From figure 5.21 for SB, it is clear that the optimisation boundaries are preventing the optimisers from achieving better performances. The bounds are therefore increased according to table 5.11, and are as well shown in figure 5.15.

Running a similar optimisation on the extended bounds leads to a premature convergence: the high-fidelity-only optimisation and the multifidelity with a step trust

Table 5.11: Optimisation normalised starting and boundaries values for the extended boundaries case.

	Slat			Flap		
	Gap	Lap	Defl.	Gap	Lap	Defl.
Min. [†]	-1.0	-1.0	-1.0	-1.0	-1.0	-0.5
Max. [†]	2.0	2.0	2.5	2.0	1.4	3.0
Start [*]	0.5	0.5	0.43	0.5	0.63	0.3

[†] Relative to initial bounds^{*} Relative to extended bounds

region using both PSO are run and their convergences are shown in Figure 5.20 with the convergence entitled (*EB*). With those bounds, the high-fidelity only optimisation reaches a significantly better optimum than the multifidelity method. However the multifidelity optimisation reaches its optimum in a smaller number of CFD calls and a shorter time. Figure 5.21 shows as well the two optimum configurations. Despite having the same directions, and for most variables a very similar positioning, the slat deflection and flap gap are lower for the multifidelity. The flap deflections are similar, but the lower slat deflection prevents an effective flow preparation for such high flap deflection, which leads to higher maximum lift generated from the optimum high-fidelity-only configuration. It is hard to conclude that multiple minima exist, but we can conclude that the multifidelity optimisation leads to a different solution, despite having very similar trends: the function we are trying to optimisation is highly complex.

5.7 Conclusion

From the first aerodynamic test case, we have proven that the method provides a large decrease in time to reach the optimum, albeit a decrease in the exploration is observed. The step-based trust-region convergence was identical to the work found in the literature: a large decrease in the required high-fidelity function evaluation but a larger variability in convergence than the high-fidelity-only. We showed that the two trust region definitions were not equivalent: the Euclidean one finds a better optimum, is insensitive to its initial size in terms of speed but is sensitive in terms of optimum whereas the step-based trust region convergence time is sensitive to its initial trust-region value but its optimum value is not affect by the initial trust region.

The multielement academic test case showed that this kind of problem is much harder and depends on the low-fidelity model used. A faster convergence is not always seen and the variance in convergence can be substantial, especially for the step-based trust-region. A large decrease in accuracy is as well found when using the step-based trust-region. An increase in the low-fidelity accuracy provides as well a better and faster convergence.

Finally the last test case, using tools from the aerospace industry, shows different results (with a different optimiser): large time-improvements are seen when the boundaries are kept small, but the multifidelity method loses quickly the subproblem optimiser global exploration capabilities. Because the boundaries are increased, the flow is more likely to be detached; the aerodynamic tools therefore works in a area where they were not designed to be accurate (see the wrong stalling characteristics predicted by HiLi).

There is therefore a consistent decrease in accuracy of the optima found between the multifidelity and the high-fidelity-only optimisations, that seems to depend on the low-fidelity accuracy. Four sources of inaccuracies can be distinguished. First, the captured physics are different between the tools and we saw that the low-fidelity tool was providing wrong directions in certain cases. For example, close to maximum lift, HiLi does not capture the stall characteristics seen by the high-fidelity tool. Although the correction is expected to reduce the issue, its quality may not be sufficient, and the additive correction may create *artificial* local minima that will disappear only with an extensive sampling of the design space. A more thorough analysis on the effect of the low-fidelity captured physics is done in chapter 8. Secondly, and related to the previous point, the suboptimiser is expected to find the global optimum, possible only with an extensive search. But this cannot be guaranteed with metaheuristic optimisers since their purpose is to efficiently explore the design space. The function to optimise is more complex due to the introduction of the correction, increasing the non-linearities (when the low- and high-fidelity models greatly differ). Hence the multifidelity may get trapped in a local minimum, and to solve this issue, a more aggressive globalisation strategy should be implemented. Thirdly, the suboptimiser rely on the low-fidelity value to find the subproblem solution. However it may happen, particularly seen with MSES, that the low-fidelity tool converges only for certain configurations that depends on the mesh and flow physics (large separation, wake interaction). If the low-fidelity does not converge, the region is ignored and will never be considered as a candidate solution for the

trust-region optimiser. It is therefore important to ensure that the low-fidelity tool robustness matches, at least, the high-fidelity one in order not to restrict the search space. Fourthly and finally the correction model is constructed on points from the optimisation path. The points uniformity is not guaranteed and clusters may appear which translates into instabilities in the model construction. They are limited with the use of filtering as described in the chapter but still remain present. Those instabilities reduce the accuracy of the corrected function. This is already seen on the two-dimensional function presented in this chapter when comparing the single- and multifidelity optimum values.

The next chapter focuses on the method extension to multiobjective optimisation problems and improvements on the subproblem correction.

Multifidelity Multiobjective Optimisation Methodology

The previous chapter introduced the methodology, based on the existing literature. The application to the maximum lift prediction for a multielement aerofoil appears to be challenging. However, the picture showed is only a partial solution to the original thesis aim. Chapter 2 introduced the multiple objectives to consider when treating such problems, including both lift and drag. To address such problems, specific methods are used that can provide the full Pareto set of solutions. This chapter aims at extending the multifidelity trust-region based method to implement the concept of Pareto-optimality within the suboptimisation and trust-region algorithm and assess the strengths and weaknesses of such approach on the high-lift device design. Section 6.1 introduces the modification to the algorithm and then is applied to the two previous test cases: the academic (section 6.2) and industrial (section 6.3) problems. The performances are not achieving an acceptable level, explained in section 6.4 and discussed in section 6.5; the following chapter proposes improvements to the method.

6.1 Method Extension

Increasing the number of objectives imposes an extension of the trust-region method: first, the solution of the suboptimisation is no longer a single point but rather a set of points forming a Pareto front. Second, each point has at least two objective values that need to be considered when computing the ratio of improvement.

In this chapter, we now consider that the high- and low-fidelity functions have more than one output: $\mathbf{f}_{\text{low}} : \mathbb{R}^n \rightarrow \mathbb{R}^l$, $\mathbf{f}_{\text{high}} : \mathbb{R}^n \rightarrow \mathbb{R}^l$ with $l \geq 1$. The same notation as in chapter 5 is used and the problem considered is still defined by (5.1).

6.1.1 Main Process

Figure 6.1 shows the main process framework that is very similar at first to the single-objective one, presented in the previous chapter (figure 5.1): the first step is still the evaluation of the starting point with both the low- and the high-fidelity functions.

Similarly, the first iteration uses a constant correction across the design space, and then a surrogate model comes to model the error between the various models. The main difference arises during the suboptimisation: the suboptimiser must encompass the notion of Pareto-dominance in order to count the number of the non-dominated points needed by the step-based trust-region formulation. The optimiser must as well return a set of Pareto equivalent points instead of a single optimum point.

Multiobjective optimisation provides a set of solutions that minimises all objectives. Usually, this set is spread over the design space, and not constricted to a single region. The standard trust region, as presented in section 3.2.3, converges when the trust region tends to zero, in one point. It is clear that this cannot hold for a multi-objective optimisation as the solution is a region in the design space, different from the empty set. Thus this chapter is focusing on the step-based trust region, even though the Euclidean one will as well be considered.

Continuing the process from figure 6.1, the ratio of improvement needs to be computed for each point. The process is described in the following section (section 6.1.2), followed by the trust region management in section 6.1.3, and the new starting point selection in section 6.1.4.

6.1.2 Ratio of Improvement Computation

The framework is extended to parallelise the high-fidelity function computation in order to soften the wall-clock time overhead introduced by the higher number of points given by the suboptimiser. This is used to compute, for each point and for each objective, the ratio of improvement $\rho_k = [\rho_k^{(1)}, \rho_k^{(2)}, \dots, \rho_k^{(l)}]$ defined by

$$\rho_k^{(i)} = \frac{f_{\text{high}}^{(i)}(\mathbf{x}_k) - f_{\text{high}}^{(i)}(\mathbf{x}_s)}{f_{\text{corr}_k}^{(i)}(\mathbf{x}_k) - f_{\text{corr}_k}^{(i)}(\mathbf{x}_s)}, \quad i \in [1, \dots, l]. \quad (6.1)$$

Depending on the problem considered but also on the optimiser type (population-based, diffusion-based, ...), the candidate Pareto front will have different numbers of points. Because each point evaluation has a high time cost, and because high-fidelity points are used to correct the function via a surrogate model, the number of points to evaluate must be limited. This is achieved by filtering the candidate Pareto front based on the Euclidean distance between each point. All unsuccessful points (configurations that result in a failed high- or low-fidelity function evaluation) are discarded. The remaining ones are retained to be compared with the actual Pareto

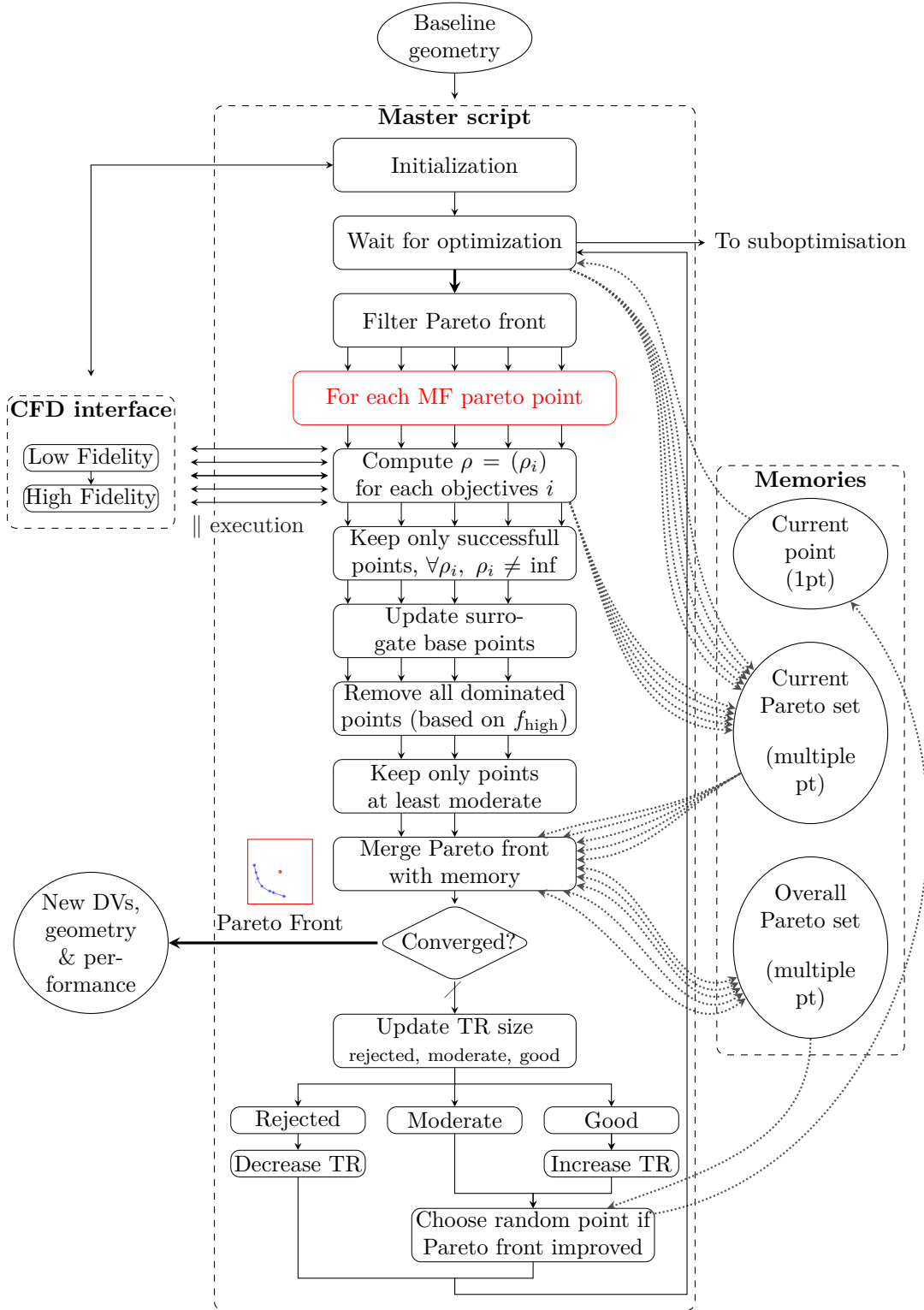


Figure 6.1: Multiobjective multifidelity optimisation framework main process.

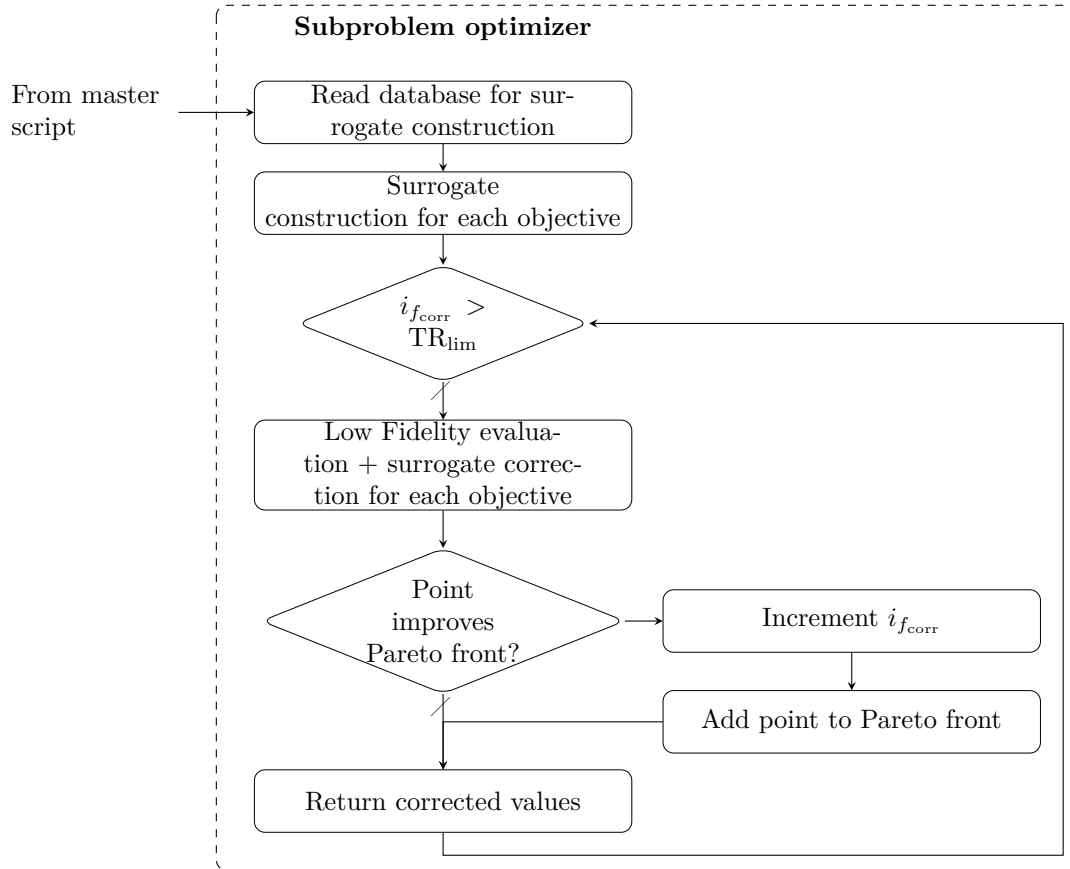


Figure 6.2: Multiobjective multifidelity optimisation framework suboptimisation part.

front and to be used at a later stage to construct the surrogate models. The author would like to highlight the increased number of high-fidelity points computed at each suboptimisation iteration compared to its single-objective counterpart: this leads to a faster correction of the design space for a similar running time, increasing the accuracy of the correction.

6.1.3 Trust Region Management

In order to handle the increased number of points and the concept of Pareto equivalence, *memories* are implemented: one coming from the suboptimiser (called *current Pareto set*), one for the overall optimisation (called *overall Pareto set*), and the current point that is the suboptimisation starting point, trust region centre; this point changes only after successful iterations and is therefore a Pareto-optimal point.

Since each point of the current Pareto front is evaluated with the high-fidelity function, the front is updated accordingly: some points that would be Pareto equivalent using $\mathbf{f}_{\text{corr}}$ won't have the same status once compared with $\mathbf{f}_{\text{high}}$. Figure 6.3 shows, with star markers, the Pareto front obtained from the suboptimiser using values from the corrected function $\mathbf{f}_{\text{corr}}$ in the objective space. In the subiteration presented, no points are filtered out. Each candidate point is then updated with its high-fidelity value depicted in figure 6.3 with arrows joining $\mathbf{f}_{\text{corr}}$ and $\mathbf{f}_{\text{high}}$. Red markers represent the Pareto-optimal points using the high-fidelity function; only those points are kept for the iteration status determination.

The ratio of improvement definition is extended according to equations (6.1) in order to be used with multiple objectives as follow:

- for each candidate point $\mathbf{p}_k = [f_{\text{high}}^{(1)}(\mathbf{x}_k), \dots, f_{\text{high}}^{(l)}(\mathbf{x}_k)]$ in the objective space,
 - if the number of objectives having their ratio of improvement greater than ρ_{bad} is higher or equal to the trigger n_f , the point $\mathbf{p}_k$ is marked *moderate*;
 - if the number of objectives having their ratio of improvement strictly greater than ρ_{good} is higher or equal to the trigger n_f , the point $\mathbf{p}_k$ is marked *good*;
 - otherwise, the point is marked *bad*.
- If the number of points $\mathbf{p}_k$ marked:
 - *moderate* is greater than the trigger n_p , the overall iteration is considered *moderate*,

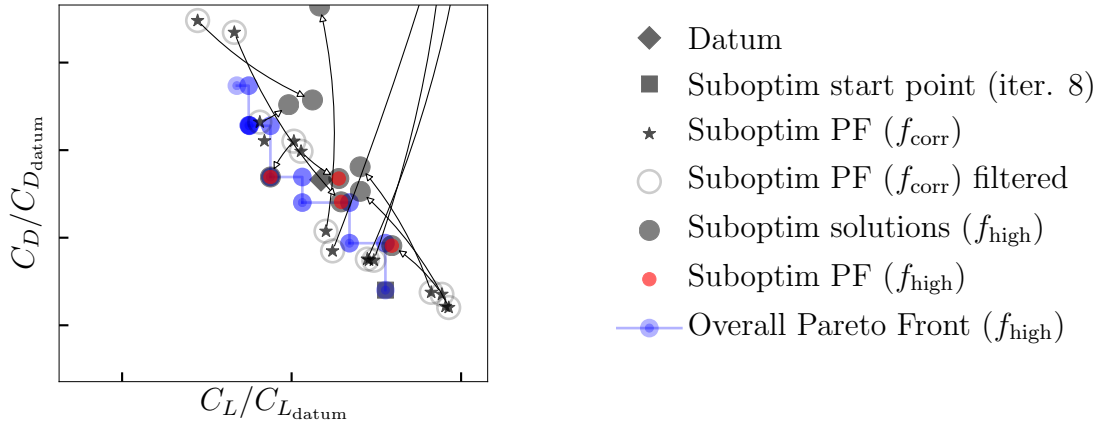


Figure 6.3: Multifidelity suboptimisation results with ESDU model and step trust region, at iteration 8. Datum is the overall optimisation starting point. Points coming from the suboptimiser are marked with a star and use their corrected function values; points kept after filtering are circled. High-fidelity values are plotted with gray circles; an arrow links the corrected and high-fidelity functions points. High-fidelity points that are Pareto equivalent are marked with red circles. Finally, the overall Pareto front at the end of the iteration is shown in blue. *PF* stands for Pareto Front.

- *good* is greater than the trigger n_p , the overall iteration is considered *good*,
- otherwise the iteration is considered *bad*.

The trust region is expended or shrunk according to the current iteration status of the according to equations (6.2):

$$\Delta_{k+1} = \begin{cases} \Delta_k - \gamma_s & |\{\mathbf{p}_{\text{moderate}}\}| < n_p & (\text{bad prediction}) \\ \Delta_k & |\{\mathbf{p}_{\text{moderate}}\}| \geq n_p & (\text{moderate prediction}) \\ \Delta_k + \gamma_e & |\{\mathbf{p}_{\text{good}}\}| \geq n_p & (\text{good prediction}). \end{cases} \quad (6.2)$$

6.1.4 New Point Selection

The current Pareto front is then merged with the overall one, noted $\mathcal{P}$, and the number of points improving the Pareto front is counted. If at least one point improves the Pareto front, a new point $\mathbf{x}_s$ is randomly chosen from the overall Pareto front, otherwise the suboptimisation is restarted from the same point with a more accurate

corrected function:

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_s & \text{if } |\{\mathbf{p}_{\text{good}}\}| \geq n_p \\ & \text{and } \exists \mathbf{p} \in \{\mathbf{p}_{\text{good}}\}, \forall \mathbf{p}_0 \in \mathcal{P}_k \exists i \in [1, \dots, l], f_{\text{high}}^{(i)}(\mathbf{x}_p) < f_{\text{high}}^{(i)}(\mathbf{x}_{p_0}) \\ \mathbf{x}_k & \text{else,} \end{cases} \quad (6.3)$$

with $\mathbf{x}_p$ and $\mathbf{x}_{p_0}$ being the coordinates in the design space of the respective points $\mathbf{p}$ and $\mathbf{p}_0$, and $\mathcal{P}_k$ being the overall Pareto front at iteration k .

The random selection is biased towards low-density part of the Pareto front in order to allow its entire exploration. To do so, the Euclidean distances d_i between each consecutive point from the Pareto front $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_f\}$, containing f points, are computed according to equation (6.4):

$$\forall i \in [1, \dots, f-1], d_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|, \quad (6.4)$$

and then used as weights for the random selection.

6.1.5 Hypervolume Indicator

The multiobjective multifidelity evaluation requires many Pareto fronts to be benchmarked after termination (convergence or allowed wall clock time exceeded) but also while the optimisation is running. In the following analysis, the Pareto front will be plotted, but due to the number of different suboptimisers, a single indicator is conjointly used to show the Pareto front evolution compared to each other.

According to Cao, Smucker, and Robinson (2015), the hypervolume parameter can help providing this information, and will be used in the rest of the document. This indicator represents the objective space volume (surface for 2 objectives) that is weakly dominated by the Pareto front, bounded by a reference point given by the user. Depending on this reference point position, the indicator is giving more importance to the number of points forming the Pareto front (diversity) and its Pareto-optimality, or to the extreme points of the Pareto front. Figure 6.4 shows the definition of such indicator. The Pareto front, which is normalised, is made of the red points that are all Pareto equivalent. The reference point is chosen slightly less optimal than the nadir point (point having as coordinates the objective values upper bounds): by doing so, the indicator uses information from the most extreme points along with the diversity inside the Pareto front. The volume is then computed using the area coloured in figure 6.4.

A potential choice for the reference point $\mathbf{r}_p = (r_1, r_2)$ is presented in Cao, Smucker, and Robinson (2015) and is summarised in (6.5).

$$\begin{aligned} |r_1| = |r_2| > 2 & \quad \text{extremes points favoured} \\ |r_1| = |r_2| \in \left[1 + \frac{\sqrt{2}}{m_p-1}, 1 + \sqrt{2}\right] & \quad \text{diversity favoured} \end{aligned} \quad (6.5)$$

The paper also introduces a *contribution rate* that helps comparing multiple Pareto fronts via the hypervolume indicator. Let's consider that n_p Pareto fronts ($PF_1, \dots, PF_{n_p}$) need to be compared; the procedure is as follow:

1. Combine all Pareto fronts into a single one, keeping only the non-dominated points. This will give the best Pareto front achievable (if the true Pareto front is not known). This Pareto front will be called the *surrogate Pareto Front*, PF_s .
2. For each Pareto front, keep only the intersection between PF_i and PF_s : $PF'_i = PF_i \cap PF_s$.
3. Normalise the surrogate Pareto front and the PF'_i .
4. Compute the hypervolume indicator for each PF'_i : $I_h = I_h(PF'_i, \mathbf{r}_p)$ and for PF_s : $I_h(PF_s, \mathbf{r}_p) = I_{hs}$, relative to the same reference point $\mathbf{r}_p$.
5. Compute the contribution rate c_r for each PF'_i according to (6.6):

$$c_r = \frac{I_h(PF'_i)}{I_{hs}}. \quad (6.6)$$

The closer to 1 is the ratio, the closer to the surrogate is the Pareto front.

6.1.6 Implementation

The same philosophy as in the single-objective framework is used: the program core is implemented in the *MultiFidelity_MO* class. More information are shown in appendix B.4. In the core MultiFidelity is implemented: the whole multifidelity algorithm, the files management, the restart strategy and the specific implementations for low- and high-fidelity models depending on the use-case. The core class also contains the *cached* version of the various fidelity model methods that keeps track of all high- and low-fidelity function calls and return a saved value if a point is reevaluated. In this implementation, all the results are stored in a single XML file, which is used at a later stage to produce the analysis plots.

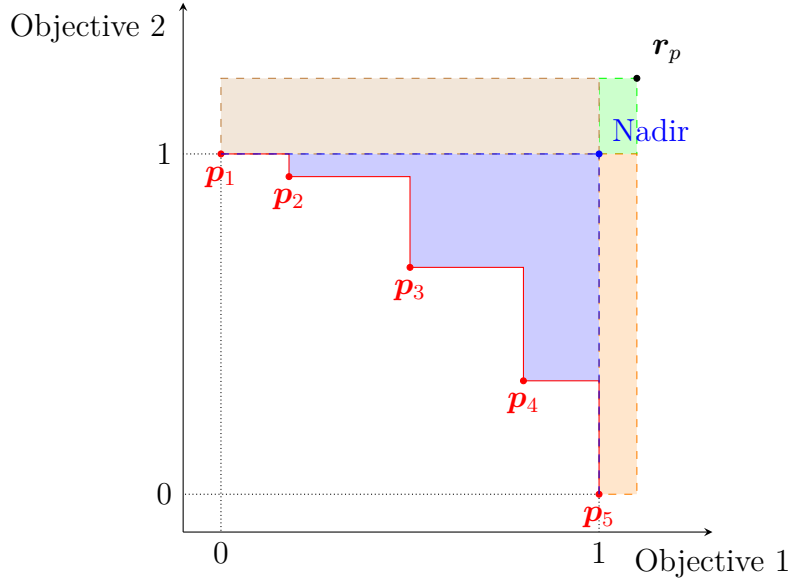


Figure 6.4: Hypervolume definition from a normalised Pareto Front $\{p_i, i \in [0, \dots, m_p]\}$ and a reference point r_p .

Because the suboptimiser returns a Pareto front rather than a single point, each point is evaluated in parallel. The master script, handling the workflow, and the suboptimisation can submit on the HPC queue system multiple high-fidelity function jobs to be run in parallel in order to reduce the wall-clock time.

Each new case requires the creation of a new class inheriting from the core MultiFidelity_MO. Appendix B.4 presents multiple UML diagrams of the trust-region-based multifidelity multiobjective framework implementation, done in Python (see figures B.6, B.8 and B.7). The biggest change is coming from the suboptimisation method that implements the surrogate creation as well as the suboptimisation.

6.2 Academic Test Case

The test case presented in the previous chapter in section 5.5 is used as is, with the same conditions. Two objectives (6.7) and (6.8) are now used in the problem formulation:

$$f_{\text{mo}}^{(1)} = -\frac{c_l}{c_{l_{\text{datum}}}}, \quad (6.7)$$

$$f_{\text{mo}}^{(2)} = \frac{c_d}{c_{d_{\text{datum}}}}. \quad (6.8)$$

Table 6.1: Multifidelity parameters for the Garteau multiobjective test case.

Parameter	Min/shrink	Initial/Value	Max/grow
Step trust region size	2	6	inf.
Step TR ratio	2		2
Number of objectives required	1		
Number of points required	2		
Number of DVs		6	
Correction surrogate		Kriging	
Suboptimiser		Tabu Search	
Low-fidelity		MSES / ESDU	
High-fidelity		ICEM & Fluent	

New parameters for the multifidelity optimisation are needed compared to the single-objective one. They are reminded in table 6.1.

The following section focuses on the suboptimisation inner workings by looking at a particular iteration. Then section 6.2.2 presents the final Pareto front along with the convergence behaviour. Section 6.2.3 shows the results obtained when the optimisation is performed only on the low-fidelity model. Finally, section 6.2.4 compares three solutions from the Pareto fronts.

6.2.1 Suboptimisation

Figure 6.3 presents a result from the 8th suboptimisation: MOTS returns a Pareto front that contains points that should improve both objectives when considering the corrected function (shown with stars). After evaluation with the high-fidelity function (with grey points, and arrows pointing from the corrected function to the high-fidelity one), those points move on the objective space: only 4 points are Pareto equivalent; the others are discarded but still used to improve the corrected function for the following iterations. Out of those 4 points, 3 do not improve the overall Pareto front, depicted in blue in figure 6.3, but because the top left point improves it, this iteration is considered successful and a new point will randomly be selected from the set of solutions for the next iteration according to section 6.1.4.

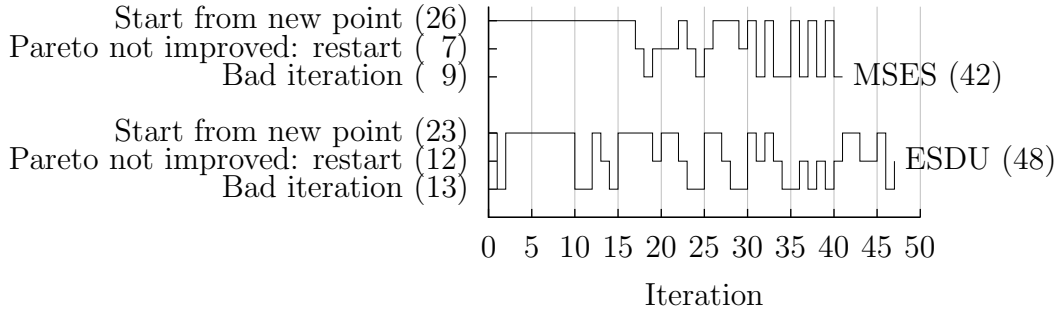


Figure 6.5: Iteration statuses distribution during the ESDU and MSES multifidelity optimisations. Numbers in bracket on the left hand side are the exact number of iteration for each category and the total number on the right hand side.

6.2.2 Convergence and Final Pareto Front

The optimisation is run using either ESDU or MSES. Figure 6.5 shows the sub-optimisation status distribution with respect to the iteration for both multifidelity optimisations: depending on conditions presented in sections 6.1.3 and 6.1.4, the iteration can be considered as *bad*, meaning the corrected function did not provide a good approximation; *restart*, if despite having a good correction, the Pareto front is not improved; or *new point* if the Pareto front is improved and a new starting point should be selected. Both multifidelity optimisations start similarly with a high number of *new point* statuses, endorsing the values used for the initial trust region size and its growth. Both then obtain balanced behaviour between *bad*, *restart* and *new point*: once the exploration is achieved, thanks to the low-fidelity model, more exploitation needs to be done with less chance of finding improvements.

The method validation is accomplished by comparing the Pareto front obtained by the f_{high} -only optimisation to those obtained with the multifidelity framework. This is shown in figure 6.6 after the same¹ number of high-fidelity function calls (CFD calls, as opposed to function evaluations —*eval*— that contains also database recalls). Although the multifidelity optimisations do not reach the same accuracy as the high-fidelity optimisation (similar outcome as for the single objective optimisation (Demange, A. Mark Savill, and Timoleon Kipouros 2016b)), the Pareto front

¹The number of CFD calls is slightly different since one iteration requires the evaluation of multiple high-fidelity points

is more spread out providing the upper and lower parts of the solution more quickly, dominating the high-fidelity Pareto front. Moreover, the multifidelity optimisation using the MSES model dominates both the high-fidelity-only and ESDU multifidelity optimisations in the high- c_l region. In the low-drag region, both low-fidelity models provide the same Pareto front.

The optimisation convergence is first analysed by plotting in figure 6.7 the Pareto fronts at 6 stages of the optimisations. Results observed in Demange, A. Mark Savill, and Timoleon Kipouros (2016b) cannot be extended for the convergence of this optimisation. From the beginning, the multifidelity optimisations are more extended but are dominated by the high-fidelity-only optimisation in the central part. Thanks to the higher accuracy of MSES close to the starting point compared to ESDU, the former obtains a dominating Pareto front up until 100 CFD calls. Then both multifidelity optimisations perform similarly despite a small dominance with MSES in the high-lift part. It is only after 500 CFD that Tabu Search using only the high-fidelity function extends its Pareto front, but 1000 CFD calls are not enough to reach the same level of exploration than the multifidelity optimisations after only 500 CFD calls. It is as well important to note that the high-fidelity-only optimisation has a denser Pareto Front. This multifidelity optimisation framework inherently keeps the Pareto front density low: two points very close on the Pareto Front have a high probability of being close on the design space. The use of multifidelity optimisation is directed to identified the most promising spots on the design space, and therefore there is no need in having many points in a single location; this helps reducing the optimisation cost.

Figure 6.8 shows a second method to compare the convergence behaviour of many optimisation methods. It is based on the hypervolume indicator introduced previously in section 6.1.5. The left plot shows the same convergence as the previous plot: only the number of high-fidelity function calls is taken into account. The *surrogate Pareto Front* represents the Pareto front from all optimisation methods and is an approximation of the ideal Pareto Front that could be achieved. The figure confirms that Tabu Search with high-fidelity function only starts quicker but gets dominated by the multifidelity runs from iteration 40. The wall-clock time on the right of figure 6.8 shows a completely different picture: the high-fidelity-only optimisation is much faster and shows a more advanced convergence (convergence achieved after 1000 CFD) at large elapsed time. We clearly reduced the number of high-fidelity

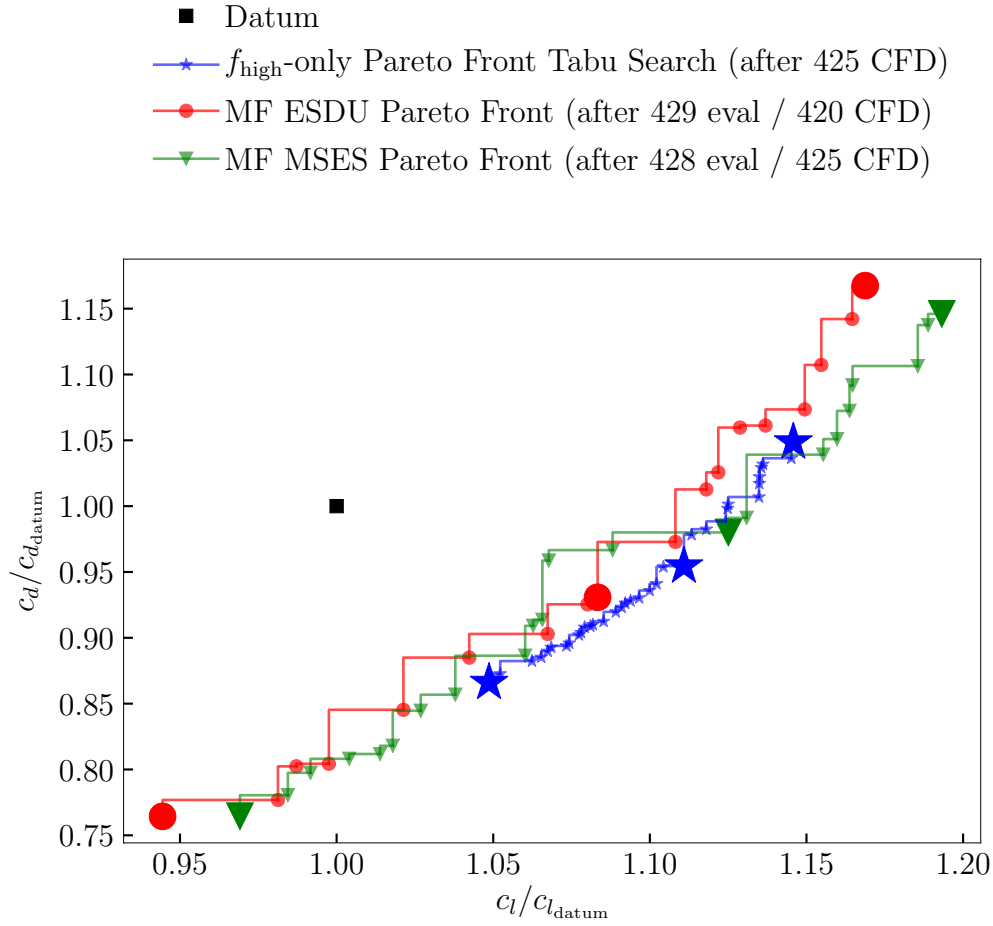


Figure 6.6: Pareto front after the same number of high-fidelity calls for the f_{high} -only and both multifidelity optimisations (ESDU and MSES). The bigger markers are the configurations used for the deeper analysis.

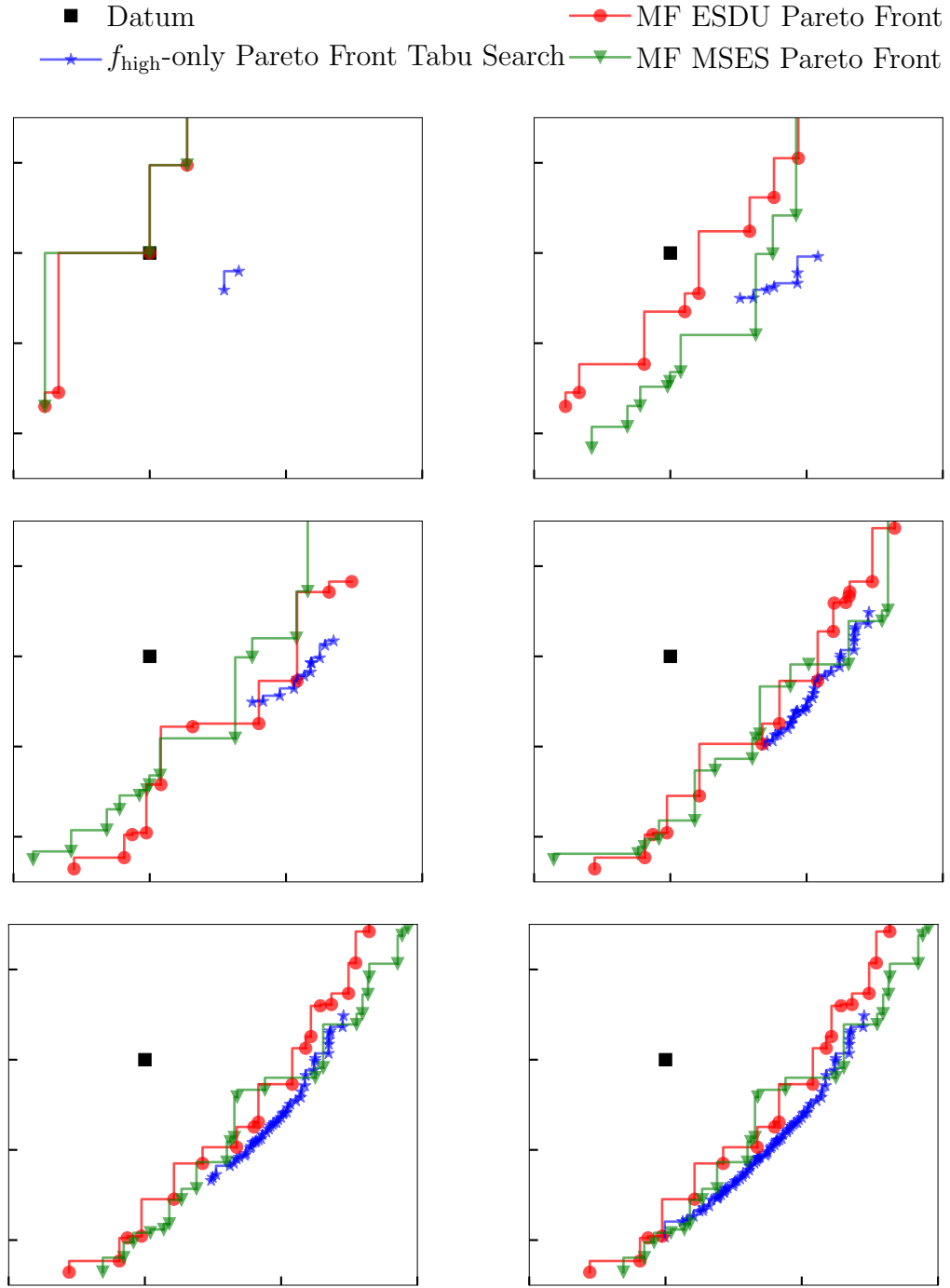


Figure 6.7: From top left to bottom right: Pareto front after 10, 50, 100, 250, 500 and 1000 CFD calls for both the high- and multifidelity optimisations.

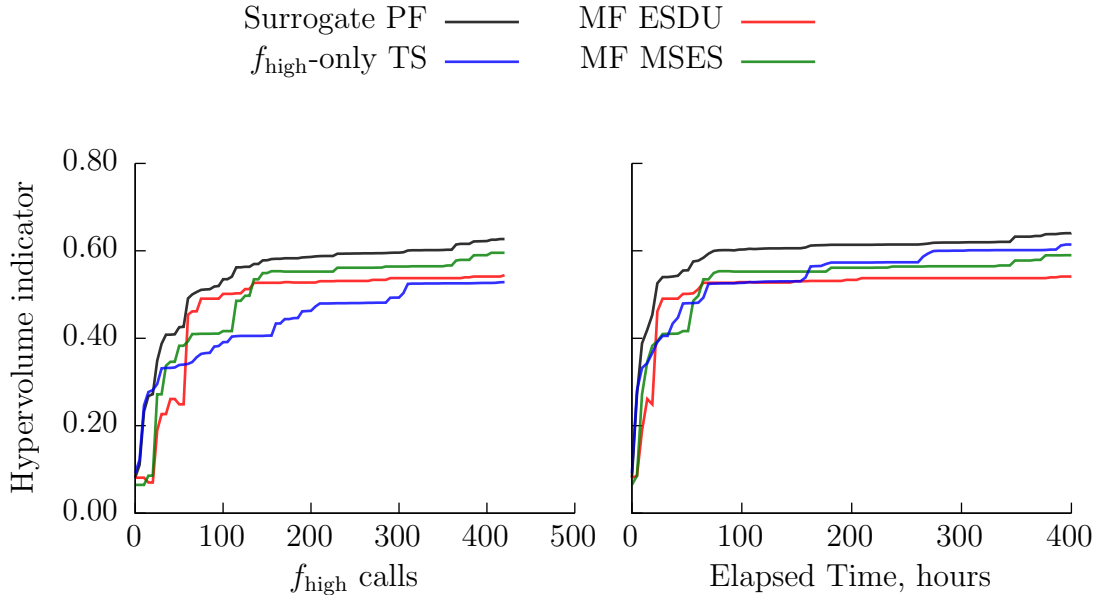


Figure 6.8: Hypervolume indicator convergence for the Garteur multiobjective optimisation with respect to the number of high-fidelity function calls (left) and to the wall-clock time (right).

function calls at the expense of a consequent time spent in the suboptimisation which penalise the multifidelity wall-clock time.

Despite having reduced the number of CFD calls, the multifidelity optimisation is slowed down by the suboptimisation process that requires the creation of the Kriging model and multiple low-fidelity model evaluations. Figure 6.9 shows the multifidelity time distributions between CFD and suboptimisation for each iteration. The first iterations (up to iteration 15) show a very small amount of time spent in suboptimisation compared to the time cost of the high-fidelity function; this can be compared with figure 6.5 which shows a good success rate in the optimisation early stage. Afterwards, the Kriging requires more time to train (since more points are used giving bigger matrices) and more low-fidelity functions evaluation is needed to obtain a sufficient improvement as shown by the green line in figure 6.9. This large increase in the number of low-fidelity function calls is the main reason for the difference between the convergences considering the number of high-fidelity function calls and the wall-clock time.

Figure 6.10 presents, for the three optimisations, the Pareto points with a parallel-coordinates plot: each vertical axis represents a variable (the first six are design

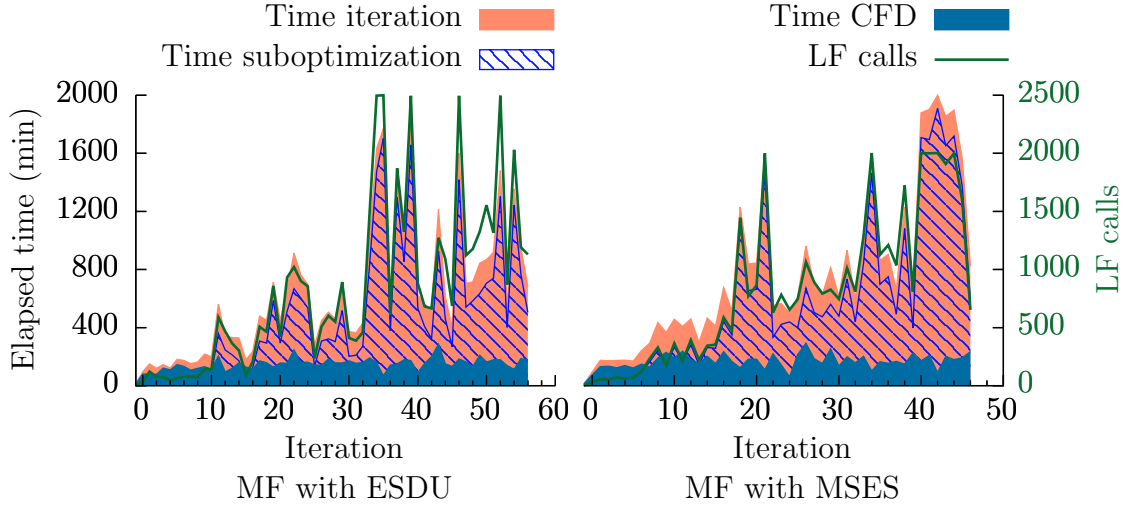


Figure 6.9: Wall-clock time distributions of the multifidelity optimisations between CFD and suboptimisation. On the left is the multifidelity optimisation performed with ESDU and on the right with MSES. The green line represents the number of low-fidelity calls for each suboptimisation

variables and the last two are the objective variables), and each coloured line represents a point in the 8-dimensions space. All variables are located in the same part of the design space. The lack of extent of the high-fidelity optimisation can primarily be explained by the small flap deflection compared to the multifidelity optimisations. Tabu Search is a discrete optimisation method relying on variable step size: this is visualised on the parallel coordinates plot with a discrete set of position for each variable. This behaviour is corrected with the multifidelity optimisation and the positioning of the slat is more continuous. Despite having a similar Pareto front between multifidelity optimisations, one can observe that the ESDU one uses a different flap positioning: both x and y being lower. This translates on the geometry as a flap with a bigger gap (distance between main trailing edge to closest flap point) and higher overlap: this configuration allows more flow re-energisation over the flap while benefiting more from the main element wake. Nonetheless, the slat position is consistent between all methods, with the multifidelity ones having a larger extent to accommodate the larger flap deflections.

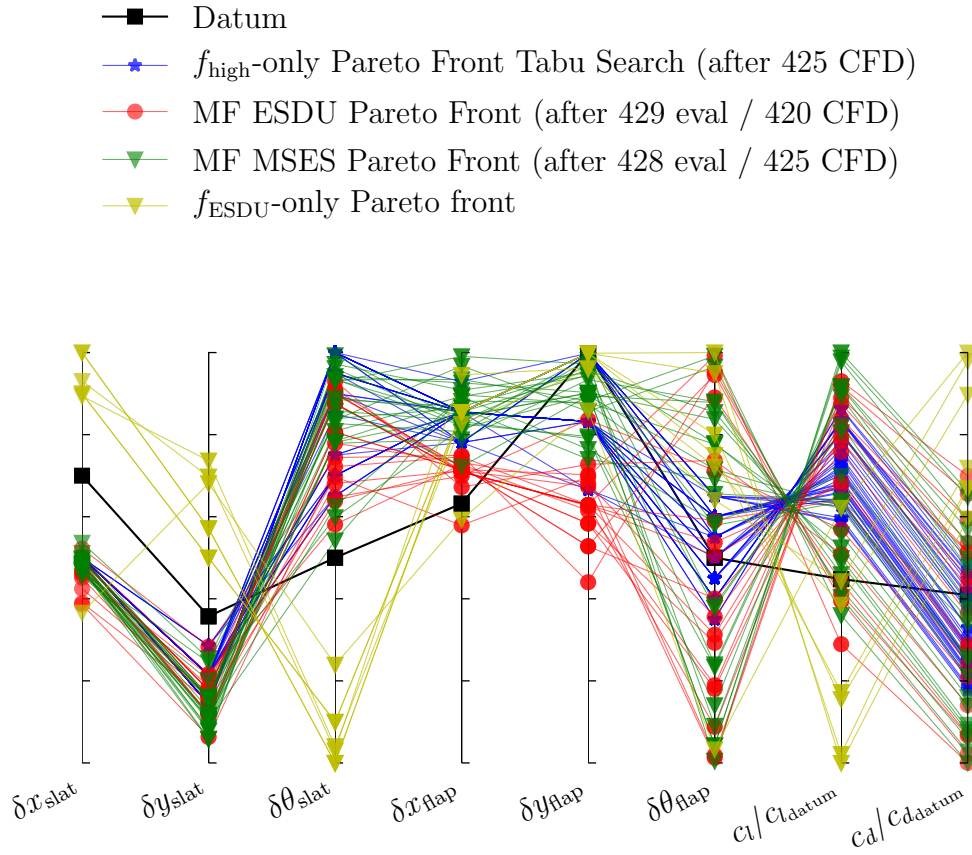


Figure 6.10: Parallel coordinates plot for the Pareto fronts with the f_{high} -only (blue), f_{low} -only (yellow), multifidelity ESDU (red) and multifidelity MSES (green) optimisations including all design variables and the two high-fidelity objectives. All objectives values are from the high-fidelity function.

6.2.3 Low-Fidelity Effect

To understand the low-fidelity effect on the final multifidelity solutions, figure 6.10 presents as well the Pareto front and design variables associated with the optimisation performed directly on the ESDU model². Firstly, none of the Pareto equivalent points improve the datum configuration (visible on the parallel coordinates by the inverse relation between c_l and c_d compared to all the other runs): the low-fidelity does not give reliable information in the high- c_l and low- c_d part of the objective space. Secondly, we can focus on the design space regions found with the low-fidelity-only optimisation. The flap positions are well estimated since ESDU tends to push the flap downstream, behind the main element, similarly to the high-fidelity position and the multifidelity ones. However, the slat positioning is opposite which explains the lower maximum lift and high drag due to the flow not being prepared for the high-energy extraction done by the flap. Thus, ESDU method is relatively accurate for positioning the flap, which accounts for most of the lift gain, but fails to position the slat in the required position to prepare the flow for the aggressive flap position; this results in a flow that cannot stay attached, and therefore results in a loss of lift. This illustrates the complicated objective space resulting from the highly non-linear behaviour of the Navier-Stokes equations. The multifidelity framework is successful at correcting the low-fidelity estimation, especially seen on the slat position, but does not sufficiently utilise this information, particularly on the x -axis for the flap.

6.2.4 Three Configurations Analysis

To further understand these discrepancies, the flow around the optimised geometry is compared for three positions shown with bigger markers in figure 6.6. They are selected to represent respectively: the maximum lift, the minimum drag and the *compromise* solution of the high-fidelity-only and multifidelity optimisations. The two extremes are the anchor points approximation and define the solution set bounds. Those two points also define the *utopia* point (Marler and Arora 2004) $\mathbf{P}_u$, which is the point having as objective space coordinates the corresponding anchor's coordinates:

$$\mathbf{P}_u = \{P_u^{(i)}, i \in [1, \dots, l]\} \text{ with } P_u^{(i)} = \min_{\mathbf{x} \in \mathcal{D}} f_{\text{high}}^{(i)}(\mathbf{x}). \quad (6.9)$$

²The final Pareto points are evaluated with the high-fidelity model.

This point is approximated with the Pareto front obtained: $P_u^{(i)} \approx P_{u_{\text{approx}}}^{(i)} = \min_{p \in \mathcal{P}} p^{(i)}$ with $\mathcal{P}$ the Pareto front. Similarly, the *nadir* P_n point can be approximated with $P_{n_{\text{approx}}}^{(i)} = \max_{p \in \mathcal{P}} p^{(i)}$.

The *compromise* point P_c is then estimated by using the closest Pareto-front point in the sense of the Euclidean distance after normalising each point with the anchors:

$$P_c = \min_{p \in \mathcal{P}} \|\tilde{p} - P_u\| \quad (6.10)$$

with $\mathcal{P}$ being the Pareto front and $\tilde{p} = [\tilde{p}^{(1)}, \dots, \tilde{p}^{(l)}]$:

$$\tilde{p}^{(i)} = \frac{p^{(i)} - P_{u_{\text{approx}}}^{(i)}}{P_{n_{\text{approx}}}^{(i)} - P_{u_{\text{approx}}}^{(i)}}. \quad (6.11)$$

The approximated anchors and compromise points are shown with a parallel-coordinates plot in figure 6.11 for the three optimisations with the corresponding geometries below. Each optimised configuration presents a similar slat placement with the x and y positions being very consistent; they are as well consistent with the datum that represents an already optimised configuration for a different angle of attack. The slat deflection varies more albeit it is higher for all configurations. Oddly, the highest deflection for the slat are used for the compromise and low-drag points rather than the high-lift configuration which requires a higher flap deflection. The high-fidelity-only optimisation however, appears to use this highest deflection for the high-lift. As pointed out earlier, the latter does not have the same large flap deflection since its exploration is not yet finished.

Figure 6.12 shows the high-fidelity and multifidelity optimisations' lift and drag breakdown for each element, with the values normalised with the baseline configuration. Globally, the optimised solutions tend to unload the slat, but load more the main and flap elements regardless of the selected point (high-lift, low-drag or compromise). In more detail, the slat loading is reduced for all methods and for all selected points compared to the datum, but its drag is higher for the high-lift configuration obtained by the high-fidelity-only and the multifidelity with MSES optimisations. The main element is more loaded by all methods and for all points with a constant loading for the high-fidelity-only configurations, whereas the multifidelity configurations tends to load more the main element to get more total lift, although the drag is reduced for all configurations. The main differences between the points arise from the flap drag due to the higher deflection angle from the multifidelity optimisations.

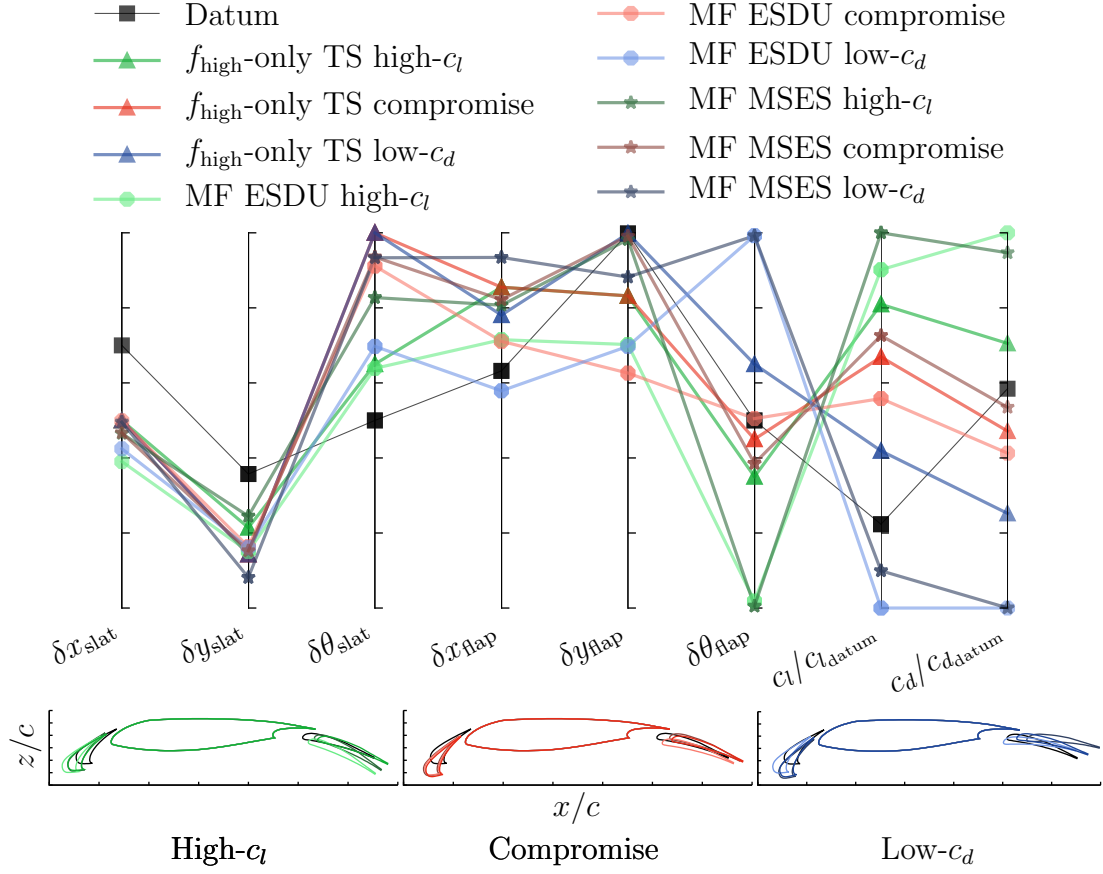


Figure 6.11: On top: normalised parallel-coordinates plot of the selected configurations representing the low-drag, compromise and high-lift configurations for the high- and both multifidelity optimisations; on bottom: selected configurations geometries.

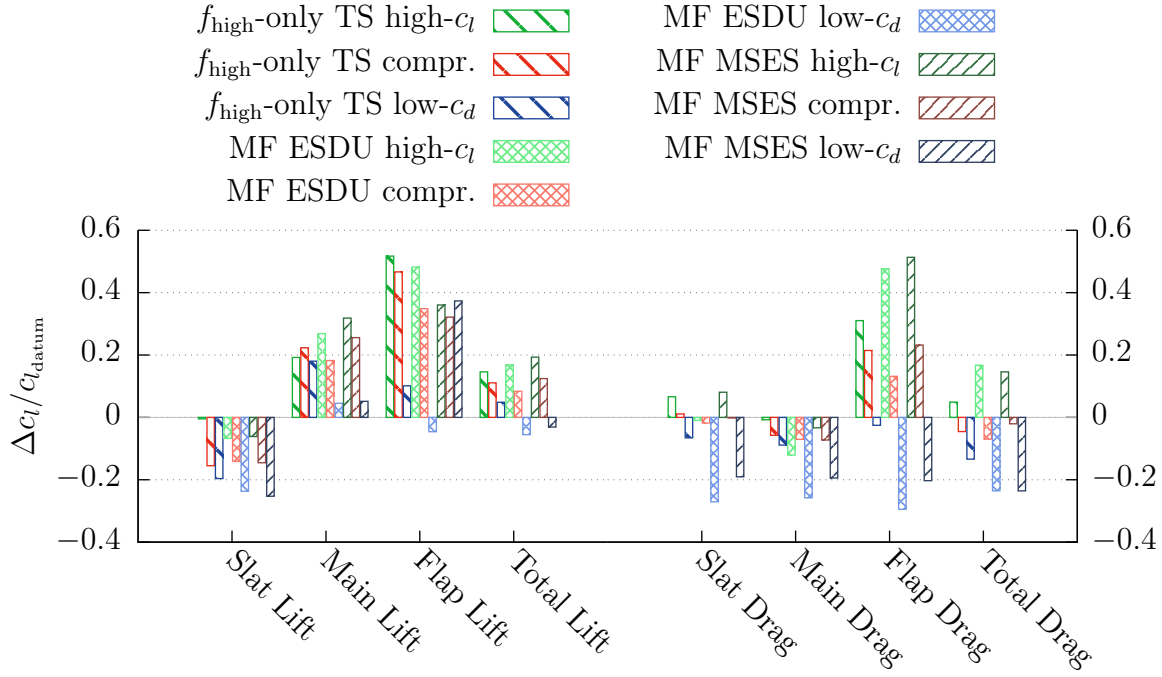


Figure 6.12: Lift and drag coefficients breakdown for both high- and multifidelity optimisations, for the selected configuration (high-lift, compromise — *compr.* — and low-drag). Values are normalised by and relative to the baseline configuration.

The high-fidelity-only configurations load the flap more than the multifidelity ones. Strangely, configurations resulting from the MSES low-fidelity model have a constant flap loading, even for the low-drag configuration, which results in a large loading increase compared to the datum. Despite the difference in flap drag, all methods follow the same pattern: a large drag increase for the high-lift configuration, a moderate one for the compromise, and a drag reduction for the low-drag ones.

To conclude this section, all methods tend to unload the slat and increase the main and flap elements loading while restricting the drag increase from the flap, thanks to more wake interactions between the slat and the main elements. Also, although the low-fidelity model gives opposite trends in some locations, the correction allows a more reasonable placement of the slat and flap. However, some trend differences depending on the low-fidelity model used still exist.

6.3 Industrial Test Case

Similarly to the single-objective chapter, the same method is applied on the second problem, defined in the industrial environment, using a different geometry, suboptimiser and different aerodynamic tools. The same geometry, conditions and parameters as in section 5.6 are used. Another objective is added to transform the problem to a multiobjective one; the two objectives (minimisation) are shown in equations (6.12) and (6.13), justified in chapter 2:

$$f_{\text{mo}}^{(1)} = -\frac{c_{l_{\text{max}}}}{c_{l_{\text{max, datum}}}}, \quad (6.12)$$

$$f_{\text{mo}}^{(2)} = -\frac{L/D}{L/D_{\text{datum}}}. \quad (6.13)$$

6.3.1 Suboptimisation

The method implemented so far is independent of the suboptimiser as long as it provides a Pareto front for the multiobjective problem. Hence, this section compares three optimisers:

- the commercial optimiser from Datadvance, Macros representing the state of the art SBO optimisers,
- the Multiobjective Particle Swarm Optimiser developed as part of this work and presented in section 3.2,
- and the well-proven NSGA-II algorithm from the DEAP library³ (Fortin et al. 2012).

All the multifidelity corrections are done with the Kriging model.

6.3.2 Convergence and Final Pareto Front

The optimisation process has not changed since the last section and there is therefore no need to dive into the suboptimisation inner workings. The final Pareto front after approximatively the same number of CFD calls is shown in figure 6.13. The CFD limit is set according to Macros that stopped once it achieved its exploration for a minimal number of CFD evaluations.

³Itself included in the Airbus optimisation toolbox openDACE.

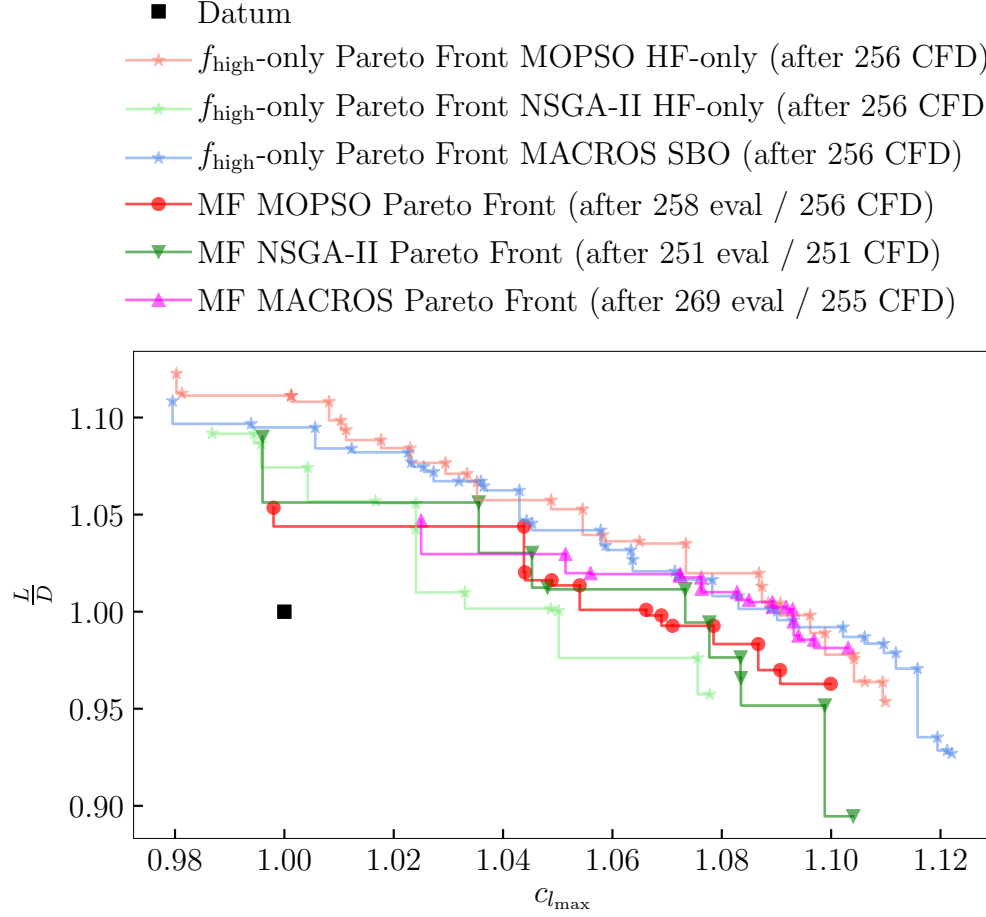


Figure 6.13: Pareto front after approximatively 250 CFD calls for the industrial test case.

Focusing first on the traditional single-fidelity optimisation (with stars in figure 6.13) it is clear that the genetic algorithm NSGA-II is not suitable for such expensive simulation: due to the random initialisation, and the slow random progression, the Pareto front points, including the anchors, are dominated by the other methods. Oppositely, the MultiObjective Particle Swarm Optimisation handles very efficiently the problem considered. It has a similar accuracy compared to Macros SBO. Only the anchor point towards the high- c_l is not as good. Not shown on this graph, is the tendency of MOPSO to prematurely converge in some runs.

Considering the multifidelity optimisation, different behaviours are observed. When used with MOPSO, as previously, the exploration is reduced and both the Pareto front extent and optimality are reduced. Since the optimisation design space is restricted by the trust region, the MOPSO parameters (population size, speed, ...) should be adapted for best performances. The MOPSO parameters study within a multifidelity trust region algorithm should be the subject of future works. On the contrary, NSGA-II is improved when used within the multifidelity framework. Despite a very limited number of points forming the Pareto front, the multifidelity solution dominates its single fidelity counterpart. The multifidelity trust region restricts the exploration and forces the genetic algorithm to explore more thoroughly the regions close to the optima. Two population-based optimisers having different performances when used in a single-fidelity optimisation perform however similarly within the trust-region. It is important to keep in mind that both optimisers are relying on randomness which is being smoothed by the multifidelity due to the numerous suboptimisation starts. Finally using Macros as-is in the multifidelity process, leads to a high concentration of points in the high- $c_{l_{\max}}$ region. The trust-region method requires a small number of points to be evaluated during the subproblem optimisation that must lead to an improvement of the corrected function. Macros SBO mainly performs a DoE around the starting points and therefore cannot explore the regions far from it, leading to restrictive exploration.

In order to have more insight in the multifidelity process, the iteration statuses for each multifidelity optimisations are presented in figure 6.14. As mentioned previously the SBO approach within the trust-region multifidelity leads to many failed iteration due to the lack of sampling points available for each suboptimisation. The two population-based optimisers have few iterations leading to a Pareto front improvement: most iterations provide points improving the suboptimisation starting point but not sufficiently to provide Pareto-dominating points.

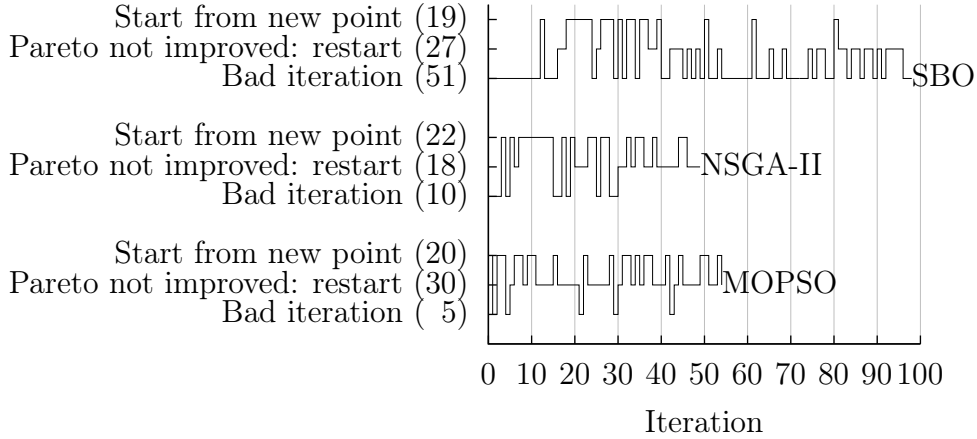


Figure 6.14: Multifidelity iteration statuses for the three optimisations.

More globally, considering SBO and MOSPO single-fidelity as reference, none of the multifidelity run managed to match their accuracy and their extent. The largest discrepancy appears on the upper left corner, in the high- L/D region.

6.4 Problem Identification

Because differences in the results between the academic and the industrial work are seen, the shortfall of the correction needs to be understood. To do so, one surrogate for the high- and one for the low-fidelity functions are created in order to:

1. have a fast model so that its value can be known anywhere and the optimisation can be left until convergence,
2. keep both functions as close as possible to real CFD,
3. and keep the low- and high-fidelity functions behaviour.

Both models are constructed and validated with the commercial software MACROS from Datadvance (see table 6.2). The High Dimensional Approximation (HDA) method is selected by the decision tree due to the high number of training data, coming from all the previous optimisation gathered points. The method consists of a decomposition of the approximation with different basis functions, similar to a neural network using sigmoid activation functions (Belyaev and E. Burnaev 2013).

Table 6.2: Surrogate information constructed by Macros to replace CFD expensive functions for test purposes. HDA stands for High Dimensional Approximation.

	f_{high}	f_{low}
Sample size	1371	19613
Technique	HDA	HDA
RMS (obj. 1 / obj. 2)	0.0132 / 0.464	0.0232 / 5.108

The test sample is automatically split into data for training and data for validation. The resulting Root Mean Squared Error (MSE) is shown in the table 6.2: only the low-fidelity model RMS error for the lift-over-drag ratio is high, but considering the testing purpose of the model, this value is considered satisfactory.

To find an explanation for the lower performances induced by the multifidelity, the same 3 optimisers are used: the MOPSO, the NSGA-II optimiser and the commercial tool Macros SBO, all introduced previously. All function evaluations, still labelled as high- and low-fidelity are only done via the surrogate model.

Figure 6.15 shows the evolution of the hypervolume indicators during convergence; the surrogate Pareto front indicator defined in section 6.1.5 is also represented. The figure presents a quick overview of the Pareto fronts quality and their convergence at the cost of not differentiating between Pareto-optimality and extremes location. First, all the single-fidelity (high-fidelity) Pareto fronts present the best indicators, with Macros SBO and MOPSO being very similar although MOPSO seems faster to converge at first, but does not reach the same final value; NSGA-II is slower but finally converges to the same indicator value, still after 1000 “CFD calls”⁴. In this case MOPSO performs particularly well at first, thanks to the random global initialisation and the smoothness added to the objective functions from the HDA surrogate that works as a regression and hence reduces the modelled function noise and fluctuations.

Second, the multifidelity optimisations, as found on the real industrial case, do not reach the same level of convergence: the final indicator values are far lower than their high fidelity counterparts (from 16% for MOPSO up to 40% for NSGA-II).

More details can be obtained from figure 6.16 that presents the Pareto front obtained after approximatively 700 CFD calls (except Macros SBO that stops after 238

⁴“CFD calls” refers here to the number of function calls needed by the optimiser, the function being for testing purposes a surrogate of the CFD.

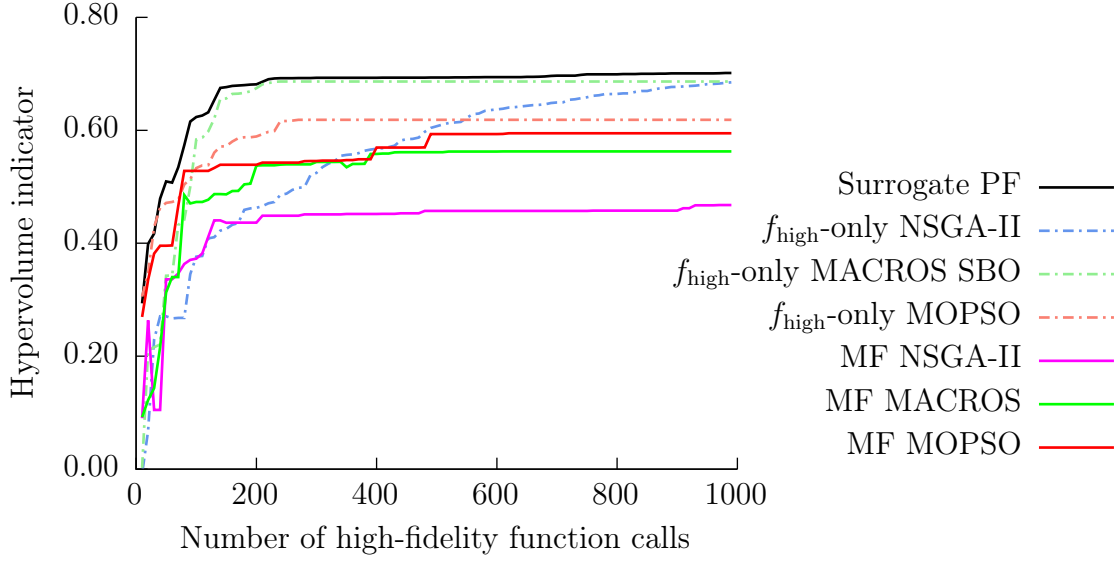


Figure 6.15: Hypervolume indicator evolution during convergence of all optimisers shown in figure 6.16. The surrogate PF is made from all surrogates obtained, keeping only non-dominated points.

CFD calls, when its convergence criterion is reached). Indeed, the high-fidelity-only Pareto fronts are more optimal than the multifidelity ones. MOPSO has many more points than Macros SBO mostly due to its internal working and extra CFD points evaluation. More differences between those two can be seen on the Pareto front than with the hypervolume convergence: Macros SBO has less points in the extremes part but is more optimal in the high maximum-lift region. This reflects its inner working: finding first the anchors, and then splitting the design space accordingly to perform a limited number of SBO local searches.

As in the real case, the multifidelity Macros has the tendency of clustering in medium-high maximum-lift region but misses completely the medium-high lift-over-drag one. This medium lift-over-drag region seems also to be missed by the other multifidelity runs confirming that this portion is the most challenging Pareto front part. Multifidelity using MOPSO however managed to get a relatively good point in the high lift-over-drag ratio part, not present in the previous real case due to the randomness involved in the swarm behaviour. Finally, the multifidelity using NSGA-II gives the worst results with only one point at the extreme high maximum-lift, and few points in the mid part; all the other points are dominated by the other methods.

We can draw better conclusions on the effect of each variable with a more thorough Pareto front analysis with the parallel coordinates shown in the bottom part of the figure 6.16:

Slat gap The very high maximum-lift part of the Pareto front is made only of point with very high slat gap whereas the points with medium to low value occupies the rest.

Slat overlap All the points are at proximity of the slat overlap lower bound. However, some methods such as NSGA-II (high-fidelity-only and multifidelity) and few multifidelity Macros points miss the exact boundary which cost them the Pareto-optimality. This highlights the design variables high sensitivity on the Pareto front.

Slat deflection Similar comments can be made on the slat deflection since most methods use a high deflection except NSGA-II that presents slightly lower deflection, but still in the same region.

Flap gap Two regions can be identified: the lower half that is occupied only by the high-fidelity-only MOPSO and some high-fidelity-only Macros SBO, and the upper half, occupied by all other methods. If the focus is made on high-maximum lift (high flap deflection), it can be seen that the position philosophy used by MOPSO gives a much lower lift-over-drag for the same maximum lift performances.

Flap overlap The best Pareto points are obtained for very low flap overlap values, attained by the Macros SBO (both in high-fidelity-only and in multifidelity), and few MOPSO high-fidelity-only and multifidelity points. More approximate methods however, such as NSGA-II do not reach the boundary and find themselves far away from the optimal Pareto front.

Flap deflection A perfect corresponding between high-deflection and high maximum-lift is found as well as low-deflection corresponding to high- L/D regions. This behaviour is expected since the functions output are the most sensitive relative to this region.

To conclude this section, it seems that methods that do not provide accurate optima such as the evolutionary algorithms cannot reach the same Pareto front optimality since the design variables do not reach the boundaries. Moreover, we can

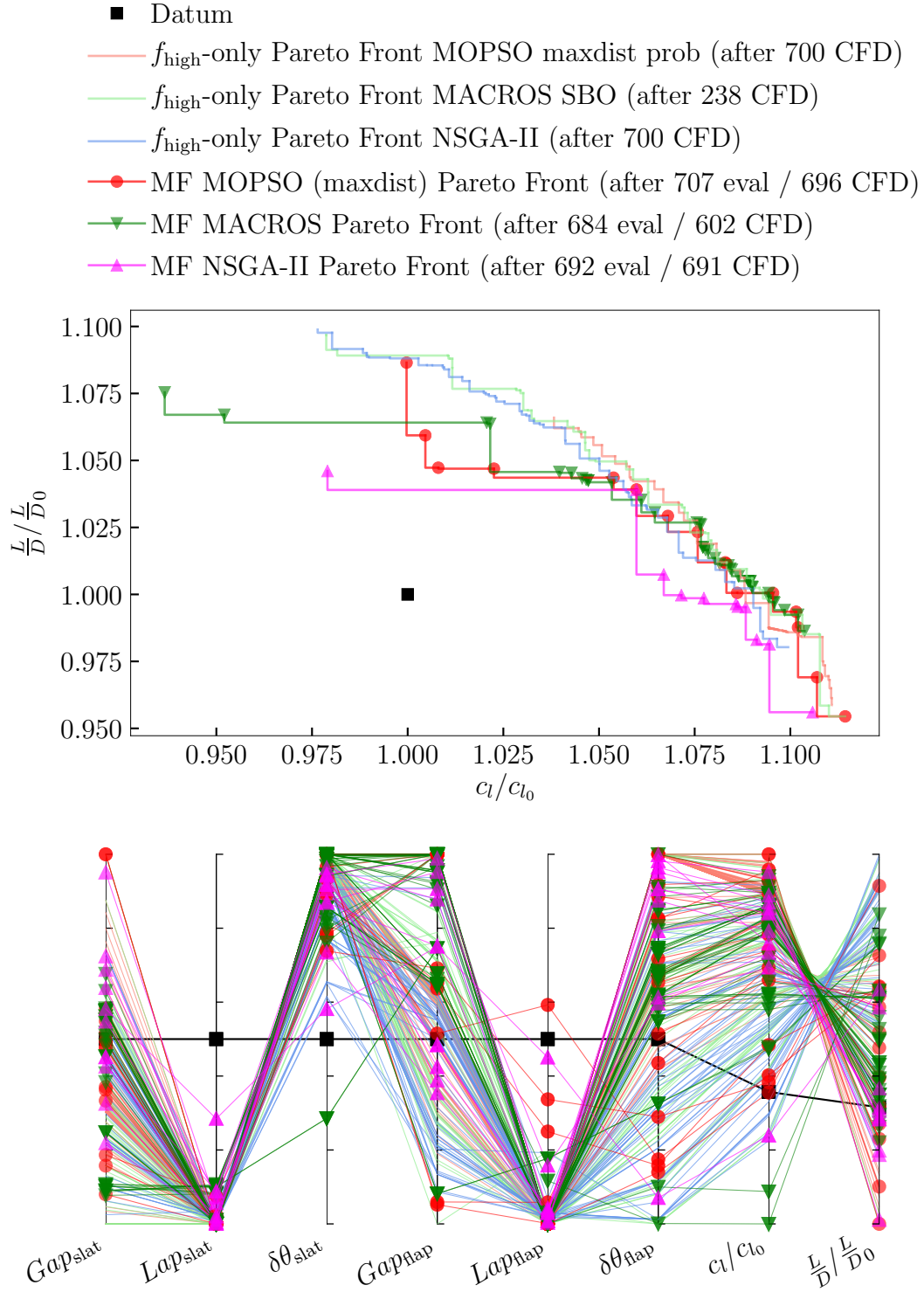


Figure 6.16: Pareto front (top) and parallel coordinates (bottom) obtained by each optimiser with and without multifidelity. The optimisers are left running for 700 “CFD calls” approximatively (“CFD calls” refers to function calls).

conclude that depending on the design space portion found, the Pareto front shape can differ. Regarding the multifidelity method, the multifidelity NSGA-II does not reach the same Pareto front due to the extra number of function calls required by the method during the suboptimisation compared to the other optimisers. The multifidelity Macros got clustered in a specific Pareto front region. Last but not least, the multifidelity runs are struggling in the high lift-over-drag ratio region.

To identify the problem, and since the surrogate high- and low-fidelity models are very cheap to evaluate, each corrected objective can be plotted against its high-fidelity counterpart. The resulting plot is shown in figure 6.17. The left hand side shows the first objective (the maximum lift) with a dotted black line showing the perfect match. The circles represent prediction made within the sampled area whereas the crosses represent the prediction made outside, that is when Kriging works in extrapolation. Apart from few predictions, most of the corrected values are well matched against the high-fidelity ones. On the right hand side, the picture is completely different: the correction range is far greater than the high-fidelity one. The match is bad and explains the issues encountered in the high lift-over-drag ratio region, seen previously. Finally, from this figure, it is important to highlight that the prediction accuracy does not change whether the Kriging interpolates or extrapolates.

6.5 Discussion

This chapter is dedicated to the application of the multifidelity trust-region based optimisation to multiobjective problems. First the extension to the single-objective was presented and tested on the academic test case, then on the industrial one. The outcome of the first application is that the multiobjective multifidelity optimisation converged to a similar Pareto front (in terms of shape), more extended than the high-fidelity-only optimisation albeit dominated. Secondly, the low-fidelity wrong trends were corrected by the multifidelity approach but leading to various slat and flap placements. The application to the second test case provided a different answer because the optimisers, the low- and high-fidelity tools were changed. First of all the optimiser is switched from Tabu Search, a method starting locally followed by a Pareto front extension, to the MOSPO, a method starting randomly and working more globally from the beginning. Consequently the Pareto front extent is not improved by the multifidelity method. But as previously, the Pareto front does not reach the same optimality and is dominated by the high-fidelity only method. Despite

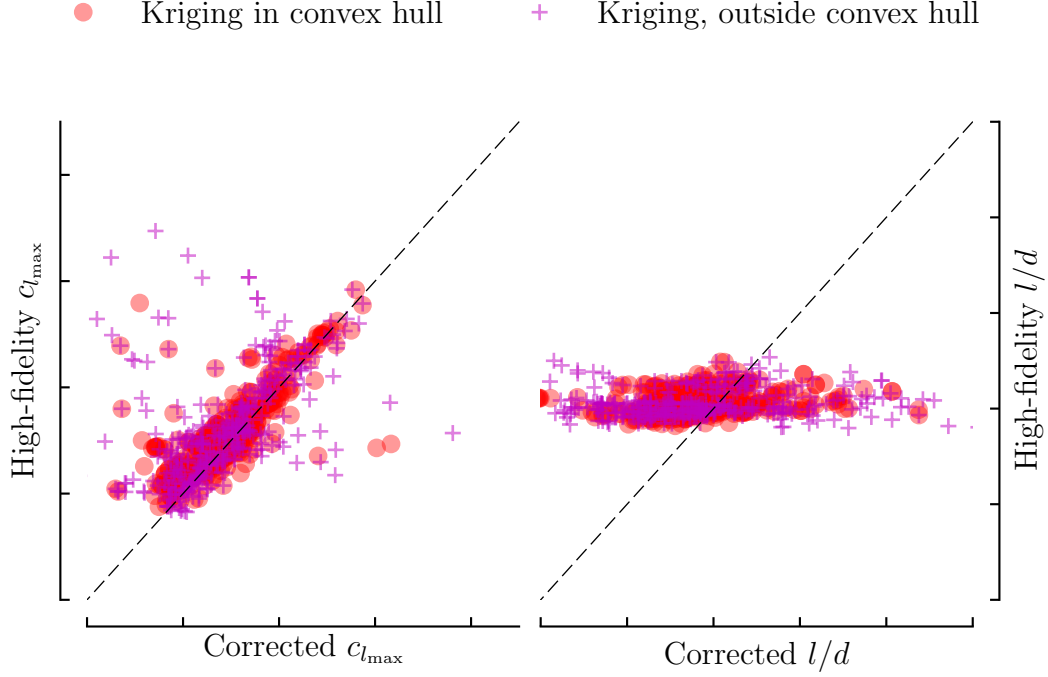


Figure 6.17: Comparison between corrected and high-fidelity objectives (left is maximum lift, right is lift-over-drag ratio).

this consistency between the use of TS and MOPSO for the multifidelity subproblem, the conclusion is not extended to the NSGA-II that is improved by the multifidelity method, this method being the slowest one to converge, even in single-fidelity formulation.

We have found that the multiobjective multifidelity optimisation is deteriorating the convergence of most of the optimisers tested. The additive correction reaches its limit due to the low-fidelity model not always positively correlated to the high-fidelity ones. The noise present in both CFD methods is also combined when calculating the difference between low- and high-fidelity models leading to a noisy response of the RBF and Kriging correction models.

The next chapter presents the use of a more advanced surrogate model to provide the correction.

Improvements with a co-Kriging Model

The previous chapter showed that the correction provided by the Kriging model was not enough to correct the inaccuracies from the low-fidelity model. A surrogate model combining both fidelities without using an additive correction is used in this chapter. The modifications in the method are shown in section 7.1 and the results with the two test cases are provided in sections 7.3 and 7.4.

7.1 Method Extension

A co-Kriging model as presented in section 3.3.5 is implemented in the framework to provide a better correction. As pointed out by Forrester, Sóbester, and Andy J. Keane (2008), a co-Kriging is more suitable than previous additive corrections to model complex relationship between low- and high-fidelity functions. It provides the interpolation of a high-fidelity function by using as well information from the low-fidelity one. Therefore equation (5.3) does not hold anymore, and we have instead:

$$\mathbf{f}_{\text{corr}_k}(\mathbf{x}) = \hat{\mathbf{y}}_{\text{cokri}}(\mathbf{x}). \quad (7.1)$$

Previously the correction construction was requiring points where both the high- and the low-fidelity functions were known. The co-Kriging interpolation however can also use the low-fidelity where the high-fidelity function is not known. During a multifidelity optimisation, as the low-fidelity is cheap, a high number of points is available although the co-Kriging performances decrease when used with more than a few thousand points (E. V. Burnaev and Zaytsev 2015) due to the matrix inversions. In the framework, only the thousand closest points from the suboptimisation starting point are used. However, due to the use of a co-Kriging inside the suboptimisation, the low-fidelity model does not need to be evaluated; the low-fidelity model needs to be evaluated at a different step. To alleviate this problem without increasing the wall-clock time, for the academic case, every time the high-fidelity function is evaluated through the queue system, the master script, that is only waiting for the

high-fidelity to finish running, is performing a design space sampling with the low-fidelity function. Despite increasing the total CPU time, the elapsed time one is still defined by the high-fidelity cost.

Finally, since the co-Kriging requires more parameters to be tuned, its minimum number of training points is higher than for the RBF which prevents the use of co-Kriging from scratch, when no DoE is provided. To overcome this restriction, the RBF additive correction is used until the number of points is sufficient to use a co-Kriging; the limit is set to $2 \cdot (n + 1)$ with n the design space dimension. Hence, the good exploration capability that relies on the direct use of low-fidelity is kept but once a sufficient number of points is available, the more accurate metamodel (but not using the low-fidelity tool during the suboptimisation) provides the corrected function.

7.2 Application on Surrogate Optimisation

The test surrogate models for the low- and high-fidelity functions introduced in chapter 6 are again used for validations. Only the best-performing optimiser is kept: the MOPSO. The only difference being the single-fidelity Kriging correction model replaced by the multifidelity co-Kriging. Macros SBO, performed only on the high-fidelity function, is used for comparison.

Figures 7.1 and 7.2 show respectively the convergence with the hypervolume indicators and the resulting Pareto front. Comparing the multifidelity convergences to the one showed in figure 6.15 (chapter 6), it is clear that the co-Kriging allows a better convergence, of the same order than the SBO: the multifidelity run performs as well as Macros SBO until 170 CFD calls. However, the convergence performance is, at first, still lower than the MOPSO on the high-fidelity-function only. This is most likely due to the suboptimisation that needs to restart the swarm evolution at each iteration, when the standalone MOPSO uses the whole swarm past experience. To finish with the hypervolume indicators convergence, it is worth noting that the multifidelity performs well in the very first iterations (up to iteration 10), since the low-fidelity function provides direction to the optimiser whereas both Macros SBO and the high-fidelity-only MOPSO are still sampling the design space.

Focusing on the Pareto front, it is hard to consider a method better than the other, as seen with the final hypervolume indicators. The multifidelity optimisation is more consistent all over the front whereas the high-fidelity-only MOPSO and Macros SBO

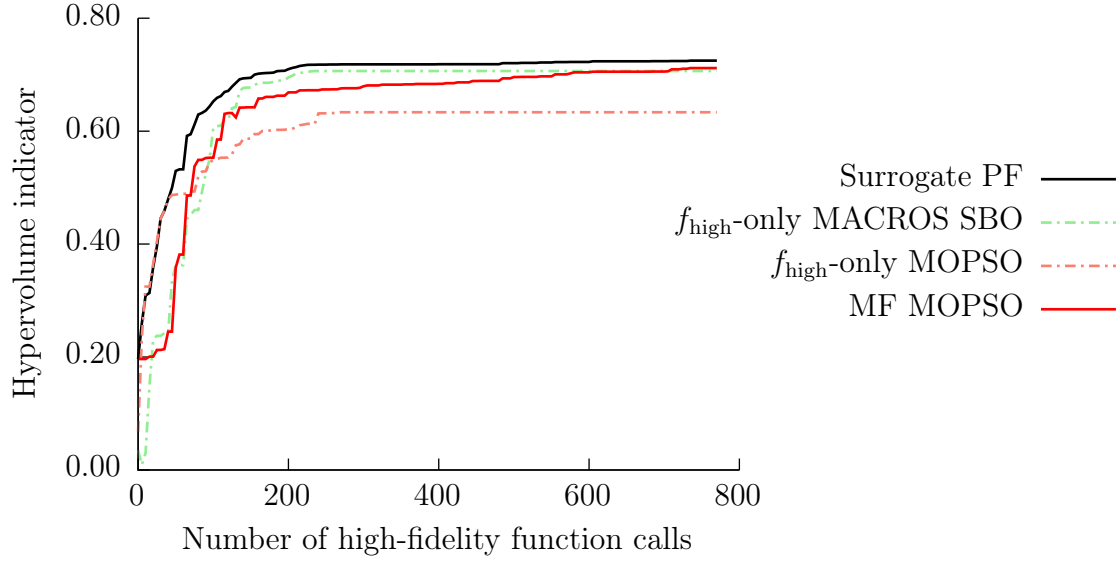


Figure 7.1: Hypervolume convergence for the optimisation on the CFD surrogate with the co-Kriging correction.

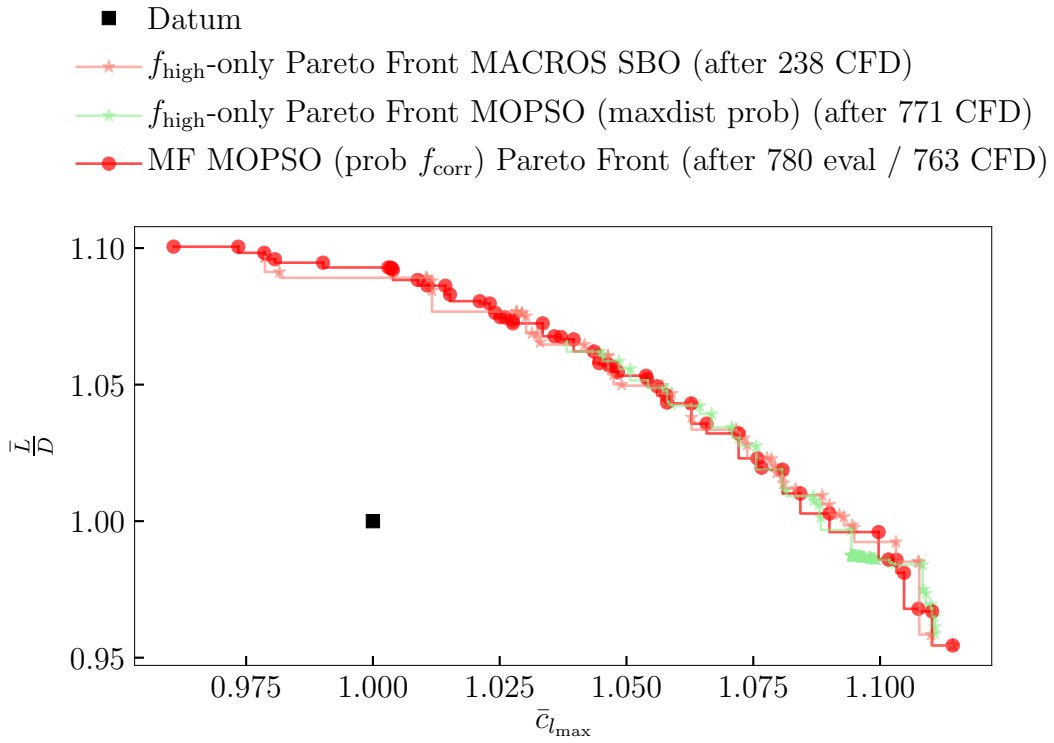


Figure 7.2: Pareto front after the optimisation on the CFD surrogate with the co-Kriging correction.

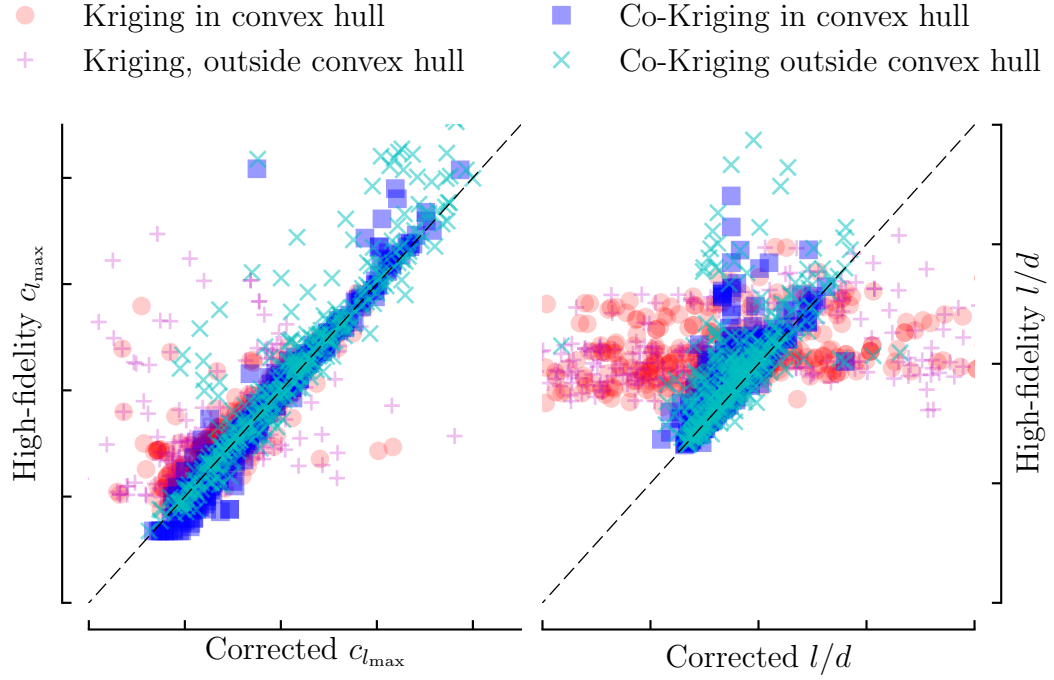


Figure 7.3: Comparison between corrected and high-fidelity objectives with co-Kriging (left is maximum lift, right is lift-over-drag ratio).

present some concessions close to the anchor points: MOPSO with high-fidelity-only is dominated by the other methods in the high-lift region but Macros SBO is dominated in the high lift-over-drag region.

Figure 6.17 that showed the prediction problem on the second objective is updated in figure 7.3. The correlation between low- and high-fidelity is more exploited with the co-Kriging resulting in a better prediction for both objectives. The maximum lift objective is closer to the perfect match line and only the prediction made outside the convex hull are deviating from this line. The biggest benefit is visible on the second objective: the lift-over-drag ratio. Most of the points are now close to the line representing a perfect match, albeit a small region tends to under-predict the high-fidelity value, no matter if the surrogate works in or outside the convex hull. Therefore, the co-Kriging is sufficient to alleviate the prediction issue encountered on the lift-over-drag ratio.

To conclude this preliminary optimisation, it is clear that the use of the co-Kriging correction fixes the problems encountered with the additive correction found on the high- L/D , at least on this test function.

The next section confirms the finding on the real CFD functions to ensure that the multifidelity optimisation performs better than its high-fidelity counterpart both in terms of accuracy and of time: the co-Kriging model training has a time cost compared to the Kriging model.

7.3 Results on the Academic Test Case

Chapter 5 showed that the Euclidean trust region was providing better results than the step-based one. Those results were not confirmed on the industrial test case. Moreover, it was shown that the Euclidean trust region was slowing down the convergence process on the single-objective industrial test case due to the high number of low-fidelity function evaluations required to solve the subproblem. Adding an objective increases in any optimisation problem the number of function evaluations to obtain a Pareto front; it can be extended to the subproblem optimisation as well.

The introduction of a multifidelity surrogate model, the co-Kriging model, alleviates the issue since the model evaluation requires a much shorter time. An implementation of the Euclidean trust region is added to the framework and this section compares the convergence and the final Pareto front obtained with the Euclidean and the step trust regions with the co-Kriging model providing the corrected function.

7.3.1 Convergence and Final Pareto Fronts

First the hypervolume indicators are plotted with respect to the number of CFD calls (left) and to the elapsed time (right) in figure 7.4. Focusing on the number of CFD calls, the multifidelity optimisations seem to require less CFD calls to reach the same dominated volume. Only the Euclidean-based multifidelity with MSES seems to struggle between 100 and 350 CFD calls. The step method, independently of the low-fidelity tool used, explores very quickly the Pareto front leading very quickly to a high hypervolume indicator. This behaviour does not seem to affect the convergence after 200 CFD calls. In terms of domination, the Euclidean-based multifidelity using ESDU has the same hypervolume indicator as the “ideal” one (the surrogate PF) meaning that it dominates almost all the other Pareto fronts (if there were merged). However, when ESDU is used with the step approach, the final hypervolume indicator value is lower, and the method with MSES as low-fidelity model provides a better hypervolume indicator. As explained in previous sections, the indicator higher value

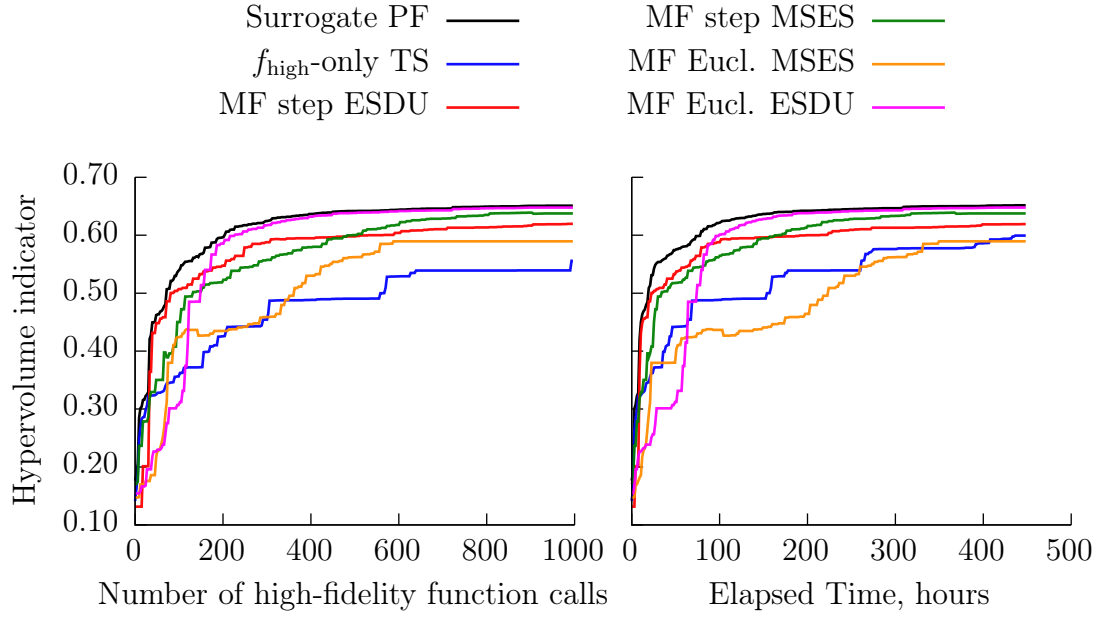


Figure 7.4: Convergence indicators against the number of CFD calls (left) and against the elapsed time (right) for the academic test case using a co-Kriging correction.

can be explained by a dominating Pareto front or a more extended one; the next paragraph is using the final Pareto fronts to understand where those differences are coming from. Focusing on the right hand part of figure 7.4, similar conclusions are seen, although the Tabu Search on the high-fidelity-function-only does not have the surrogate model construction time overhead and the hypervolume indicator difference between high-fidelity-only and multifidelity is smaller. Also, the behaviour differences between step-based and Euclidean trust-region is clearly seen: the Euclidean-based one is much slower than the step-based one due to the longer time spent to solve the subproblem. Finally, table 7.1 quantifies the savings provided by the use of multifidelity optimisation: an average of 71% CFD calls and an average of 50% of time reduction can be achieved.

Figure 7.5 presents the statuses at each iteration for the four multifidelity optimisations (defined in section 6.1). The Euclidean trust region with MSES, from iteration 23 (136 CFD calls) to 35 (257 CFD calls), has a high number of bad iterations. They are alternating between bad ratios of improvement (the corrected function not accurate) and Pareto front not improved (good prediction from the corrected function but not improving the global Pareto front). The suboptimisation starts from a single

Table 7.1: Savings provided by multifidelity to achieve an hypervolume indicator of 0.55 for the number of CFD calls and 0.60 for the time.

	Method	CFD saved	Time saved
Step	ESDU	78%	63%
	MSES	70%	63%
Eucl	ESDU	83%	77%
	MSES	53%	0%

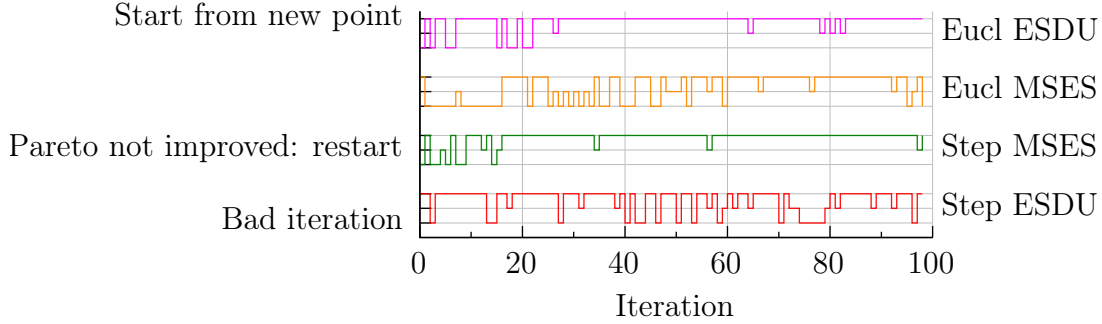


Figure 7.5: Iteration statuses for all multifidelity optimisation convergence (academic test case with co-Kriging corrected model). Status levels: top is *start from new point*, middle is *Pareto not improved: restart* and bottom is *bad iteration*.

point and the ratios of improvement are calculated between the candidate points and this starting point. Obtaining a ratio of improvement greater than 0 for at least one objective guarantees that the points improve the starting point but does not guarantee Pareto optimality with respect to the other global Pareto front points. This is what happens to the Euclidean MSES optimisation between 130 and 250 CFD calls. Focusing on the other optimisations, the Euclidean ESDU and step-based MSES has the same status convergence behaviour: the first 20 iterations alternates between success and failure, corresponding to the rough correction of the low-fidelity model, followed by a very high number of successful iterations (the suboptimisation returns points improving the global Pareto front); the correction being accurate enough to represent the high-fidelity function. Finally, from those statuses, no similarities are seen between the step-based and the Euclidean trust regions.

To extend the multifidelity optimisation time convergence analysis, the figure 7.6 shows the time breakdown for the multifidelity with MSES, including both trust re-

gion definitions. The top plot shows the time breakdown between suboptimisation (model training and optimisation) and CFD for each iteration for the step-based trust region on the left hand side and for the Euclidean trust region on the right hand side. First, until iteration 30, the suboptimisation is cheaper than a CFD call, but after, the increased number of points used for the surrogate model training increases, leading, towards the end of the optimisation, to a higher training cost by a factor of 2 to 3. Clearly, using a more advanced and efficient co-Kriging training strategy would decrease this loss of efficiency. Let's note also that because the low-fidelity model is not used during the suboptimisation, infill points are generated during the high-fidelity sample. This clearly leads to a quick increase in low-fidelity points which creates a burden for the co-Kriging training. Tests have been done without performing the low-fidelity sample other than at high-fidelity sampling points and it was shown that no significant reduction in accuracy appears but a reduction in training time occurs (figure C.4). Secondly on figure 7.6 lower part, the trust region size evolution and the number of corrected function calls are plotted: for the step-based trust region, the increase is almost linear until reaching the maximum size; for the Euclidean trust region however, the size is relatively small until the suboptimisation returns only improving points, leading to a quick increase of its size (until the maximum). Consequently the number of corrected function calls increases as well greatly with no noticeable consequences on the suboptimisation time. But the cost associated with the low-fidelity function calls is making this approach impossible when an additive correction is used (as in section 6.2). The same behaviour is observed with the ESDU low-fidelity tool, as shown in figure C.3.

The hypervolume indicators only present a partial view; the final Pareto fronts after the same number of CFD calls are shown in figure 7.7 to understand whether the difference comes from a dominated Pareto front or from a less extended front. The Pareto fronts for each method are showed in figure 7.7 with the anchor and compromise points highlighted. The Tabu Search with high-fidelity-function-only hypervolume low-score is explained by the small Pareto front portion discovered compared to any multifidelity method; the reason being the increased number of function evaluations needed to get the same extent. The lower Pareto front region is however dominating the multifidelity optimisation solutions, and the upper part is dominated. The multifidelity using an Euclidean trust region with ESDU dominates the other solutions. The low hypervolume indicator for the Euclidean MSES multifidelity optimisation is explained by the low extent and the domination close

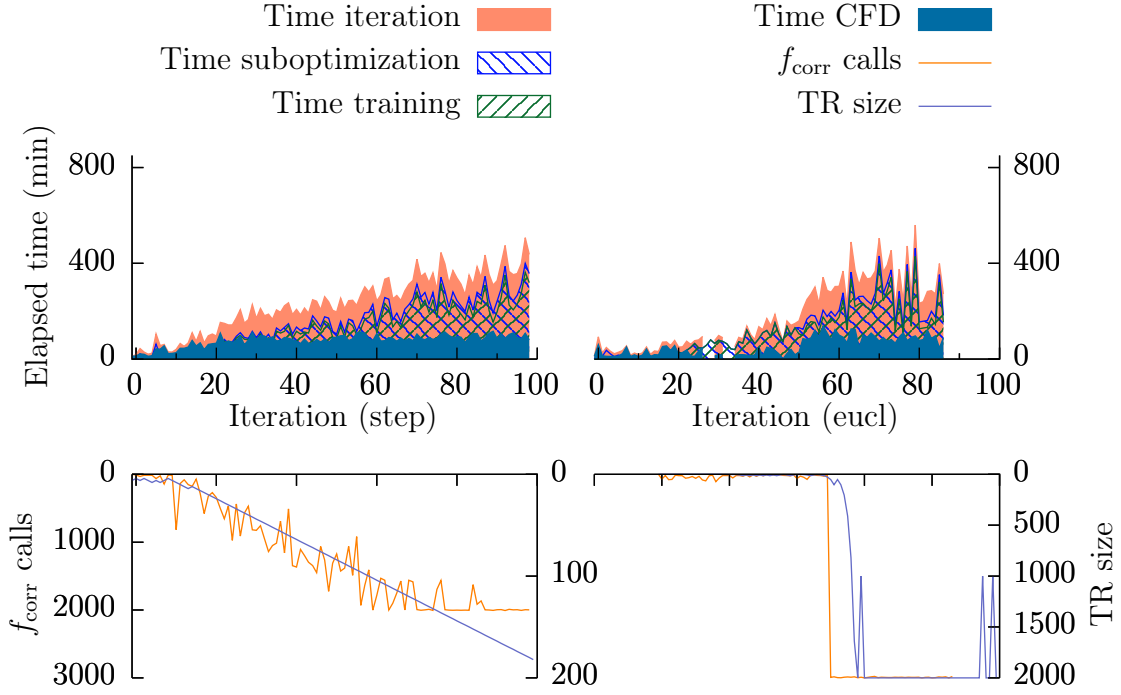


Figure 7.6: Time repartition for the multifidelity optimisation when MSES is used as low-fidelity tool (top) and trust region size along with the number of corrected function calls.

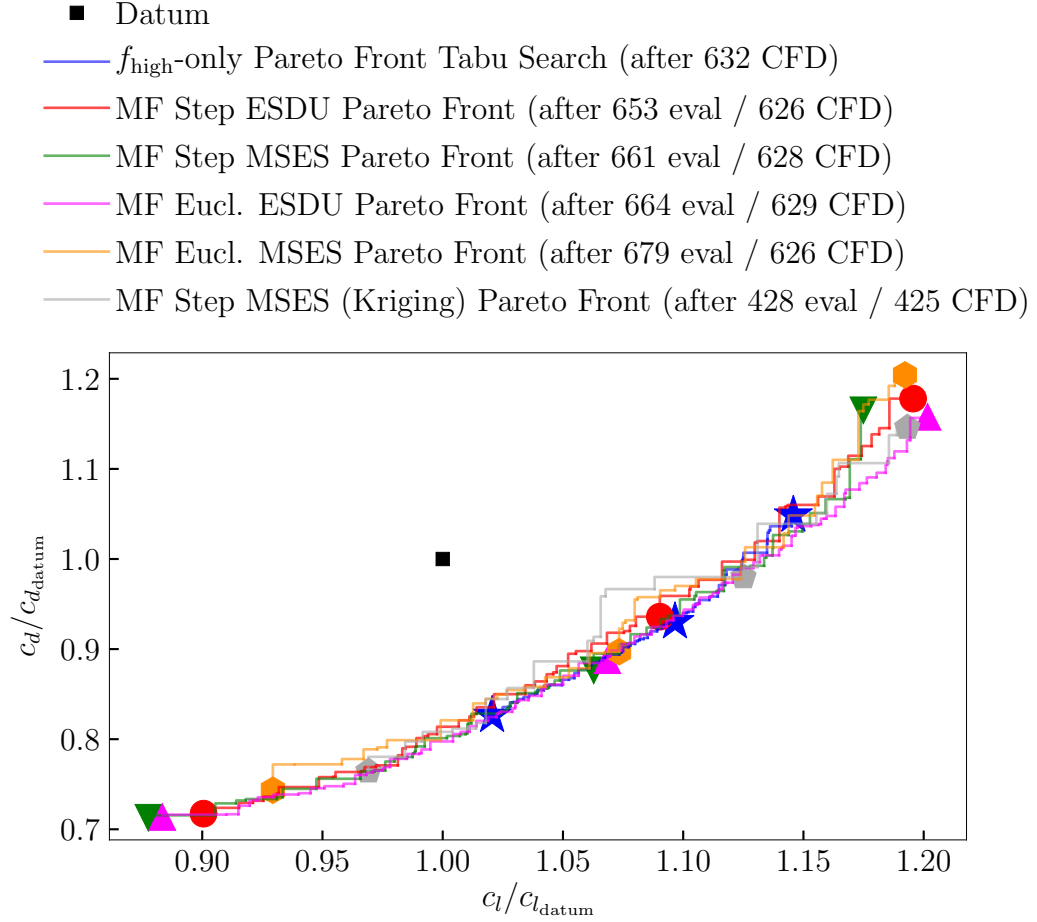


Figure 7.7: Pareto fronts after approximately 700 CFD calls for multiobjective optimisations using co-Kriging as correction.

to the anchor points. The multifidelity anchor points are Pareto equivalent in the low- c_d part but their extent varies whereas the extent is similar in the high- c_l part but ESDU multifidelity dominates its MSES counterparts. Finally, the Pareto front shape and position are very similar and consistent between each multifidelity and between the single- and multifidelity solutions. Hence, the multifidelity optimiser reduces the time (and in a bigger extent the number of CFD calls) to discover the Pareto front, no matter the trust region definition or the low-fidelity tool used. Finally, the Pareto front from the previous Chapter is also shown in the figure. Despite having less iterations, the Pareto front is less defined (less points) and the low- c_d and mid regions are clearly dominated by the new method using a co-Kriging. The co-Kriging model definitely improved the method.

7.3.2 Three Configurations Analysis

In order to get a better insight on the solutions, the anchor and compromise points are selected (shown with markers on figure 7.7) and their design variables and objective values are shown in a parallel coordinates plot in figure 7.8; the geometries are also represented. Reading from left to right, the slat position consistency is first seen: no matter the method nor the objective (compromise, low- c_d or high- c_l) the slat is always positioned more forward (δ_x more negative), lower and with a higher deflection. The deflection is the slat parameter having the highest variance, due first to the point considered: the lowest deflection is for the high- c_l points but also due to the low-fidelity tool used: the MSES solver seems to provide a lower slat deflection compared to ESDU or to the high-fidelity only. Second, the flap position has differences depending on the method used. For example, the low- c_d x position is consistent between all multifidelity (closest to the main element) but not with the high-fidelity only: this is explained by the lack of exploration performed by the high-fidelity only optimisation at this stage. Indeed this positioning associated with the lowest deflection enables to achieve the lowest drag. The flap deflection being the most sensitive variables for lift and drag, the optimum values are consistent between the methods, except the high-fidelity-only that has not explored the whole deflection space yet.

In this section a co-Kriging model is used to provide the corrected function instead of an additive correction. The problem highlighted in chapter 6 is no longer seen: despite an increase in training time for the co-Kriging model, especially towards the end of the optimisation (when lots of low- and high-fidelity points are used for the training), the time needed to find the full Pareto front is reduced by 50% in average.

7.4 Results on the Industrial Test Case

Similarly, the results from section 6.3 need to be updated with the co-Kriging correction; but before, the improvements are compared on the single-objective optimisation from section 5.6.7, with the extended bounds.

7.4.1 Single-Objective Optimisation

Section 5.6.7 concluded that the increase in the optimisation design space size prevented the multifidelity optimisation to reach the same optimum as the high-fidelity

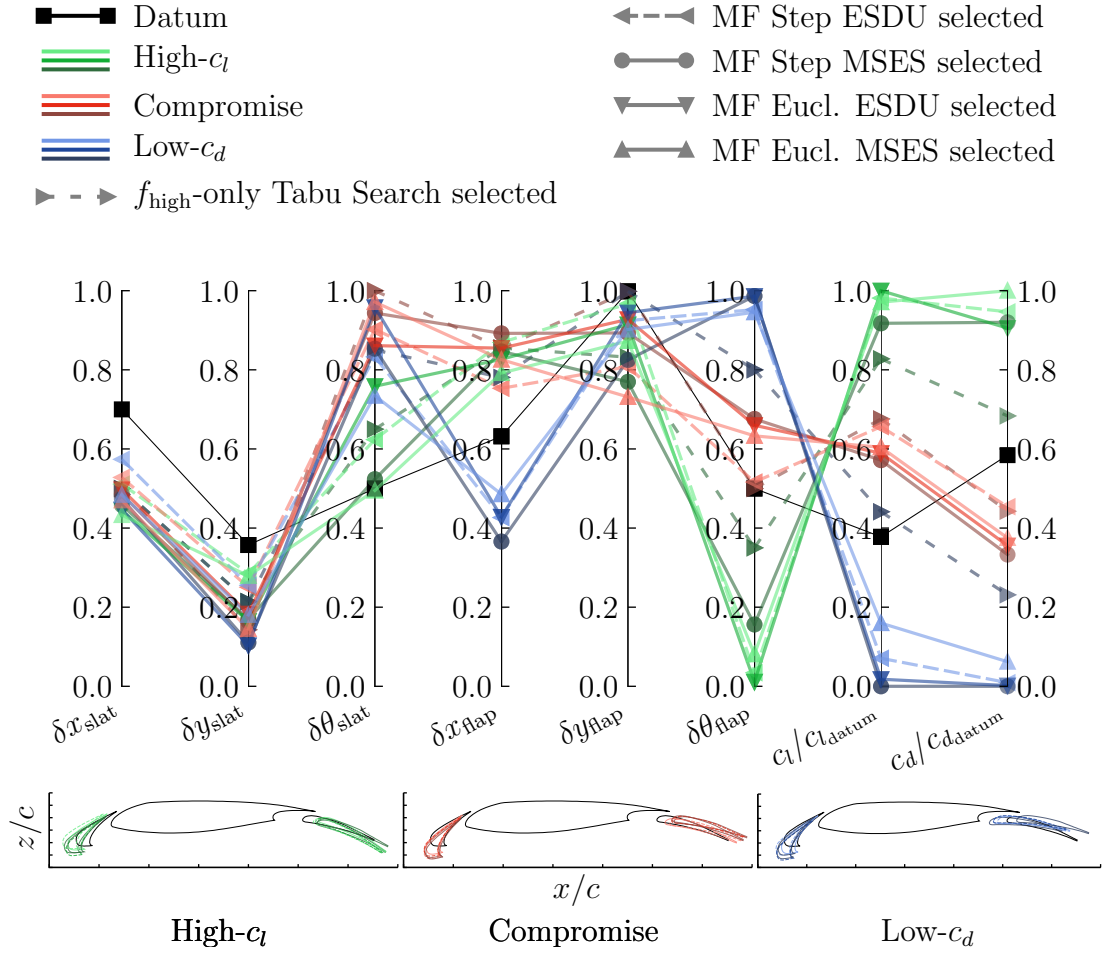


Figure 7.8: Comparison of high- c_l , compromise and low- c_d points for all methods for the multiobjective optimisation using a co-Kriging correction with a parallel coordinates plot on top and with the final geometries on bottom.

Table 7.2: Trust region parameters for the single-objective extended bounds multifidelity optimisation.

Number of objectives	1
Number of objectives required	1
Number of points required	1
Initial trust region size	8
Trust region grow/shrink ratio	2.0

only optimisers. Hence the Kriging correction is replaced with the co-Kriging model constructed from both the low- and the high-fidelity models. The same multiobjective implementation is used with a single-objective and with the parameters described in table 7.2.

The multifidelity convergences are compared with 2 single-fidelity optimisations: the SBPLX, and the ALPSO optimisers. The multifidelity suboptimisation is also done with a Particle Swarm Optimiser, but with the multiobjective implementation presented in section 3.2.4. In addition to the step-based and Euclidean trust regions multifidelity optimisations, an expected improvement multifidelity method is implemented: instead of using the co-Kriging for the corrected function, the expected improvement defined in section 4.4 (*SBO*) is used to drive the subproblem optimiser. The co-Kriging function estimate is then used to compute the ratio of improvements. The $E[I]$ implementation also differs from the other implementations for the initial sampling. In order to rely from start on the co-Kriging model, an initial sampling of 10 points is done. The standard multifidelity co-Kriging runs start with a single sampled point, and until a sufficient number of points is obtained relies on the low-fidelity model corrected by the Kriging model.

Only the extended bounds convergence is shown in figure 7.9; for reference, the small boundaries convergence is shown in the previous chapter, figure 5.20. The previous Kriging implementation is also plotted and is the only one not converging to the optimum value, as seen in the previous chapter. Only the $E[I]$ and high-fidelity only optimisations do not start from the same value due to the initial random sampling; the others starts from the datum configuration. Apart from the Kriging run, the other optimisations reach the same optimum in terms of objective function value, representing an improvement of 24%. The two single-fidelity optimisations are slower to converge both in terms of number of high-fidelity calls and in terms of elapsed time. The SBPLX run takes longer at first but quickly finds a relatively promis-

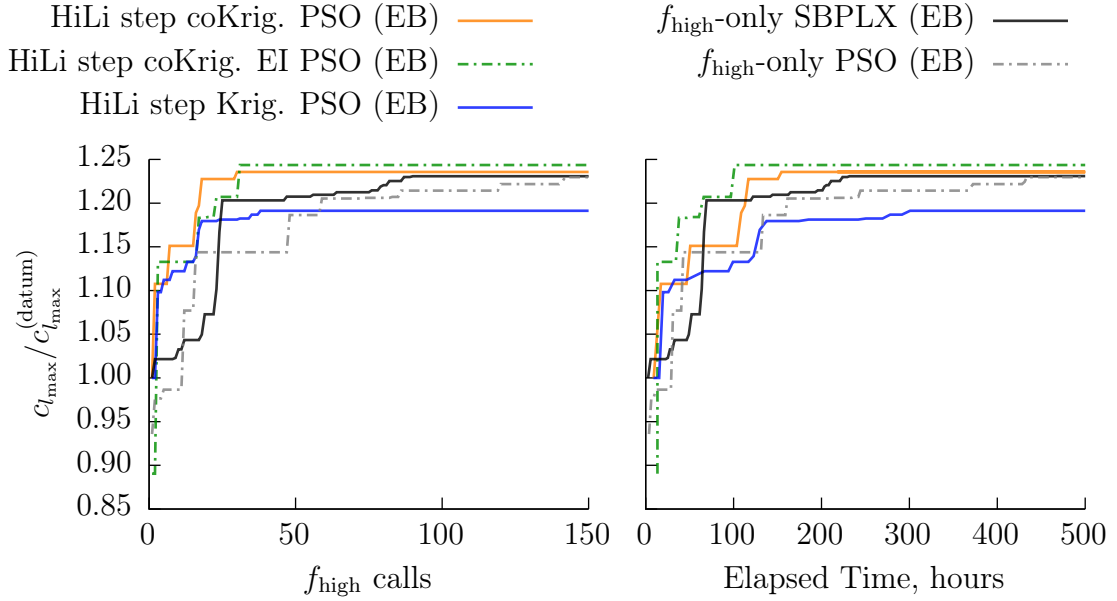


Figure 7.9: Convergence of high- and multifidelity EB boundaries optimisations with respect to the number of high-fidelity function calls (left) and in terms of elapsed time (right).

ing region, whereas the single-fidelity only PSO converges at a more steady pace due to a balance more focused on exploration than exploitation. Regarding the co-Kriging multifidelity, their convergences are similar with respect to the number of high-fidelity function calls, both reducing by 80% the number of fidelity simulations required. However, the $E[I]$ implementation is faster with respect to the elapsed time since the standard implementation spends a lot of time escaping the first region found. The standard definition reduces by 70% the required time to reach the 24% improvement compared to the single-fidelity optimisation whereas the $E[I]$ implementation saves 80% of elapsed time. The time increase due to the co-Kriging is seen in the first 50 iterations but does not impact the final objective found.

To conclude the single-objective comparison, the optima parallel coordinates are plotted in figure 7.10, with the initial small bounds shaded to highlight that most of the variables take advantage of this design space size increase. Despite its less optimal objective value, the Kriging implementation optimal positions are not different from the other configurations; the biggest difference being the lower flap deflection, reducing the produced lift. Also, the SBPLX converges to a very different final solution: the slat gap is much higher and its deflection lower. This modified arrangement

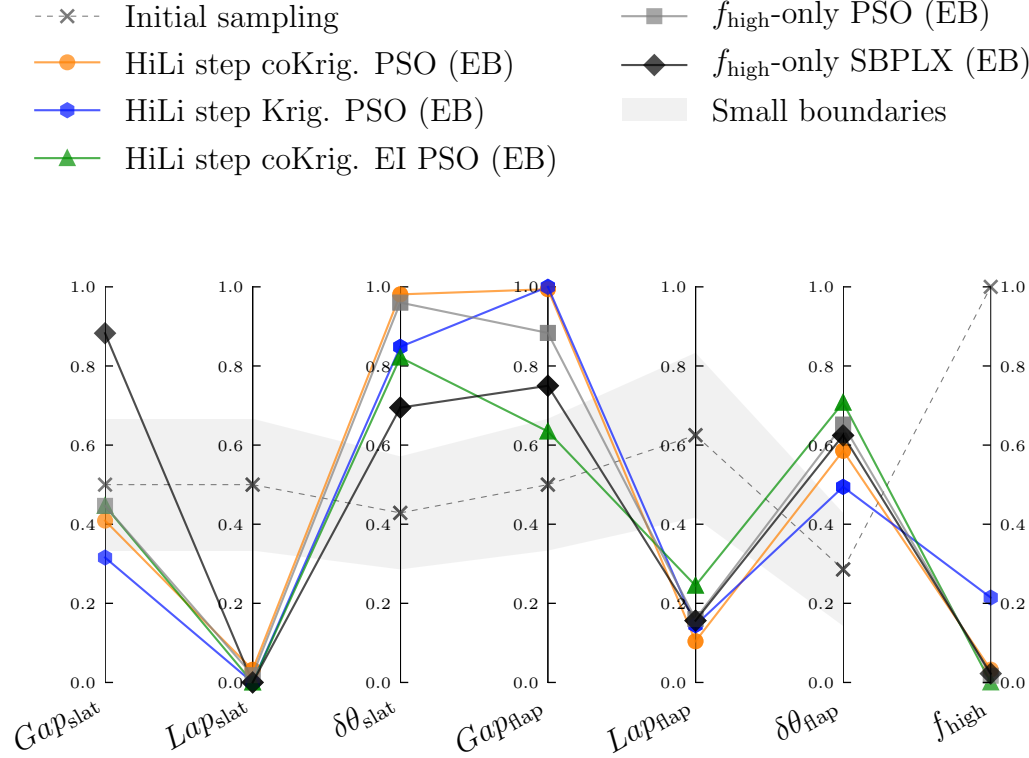


Figure 7.10: Parallel coordinate displaying the optima solutions from multifidelity and high-fidelity-only optimisations for the extended (EB) boundaries cases. The ranges are the extended one and the small boundaries are shown shaded.

still allows the same flap deflection as the other configurations, leading to a similar lift-over-drag ratio.

7.4.2 Multiobjective Optimisation

Moving forward, the multiobjective problem is now optimised with the improved implementation. The parameters defining the trust region size behaviour are shown in table 7.3.

Similarly to the single-objective EB optimisations, an expected improvement formulation is also compared to the standard implementation. The multifidelity multiobjective subproblem (5.2) minimisation is undertaken by the MOPSO algorithm for

Table 7.3: Trust region parameters for the multiobjective multifidelity optimisation.

Number of objectives required	1
Number of points required	2
Initial trust region size	6
Trust region grow/shrink ratio	2.0

all methods. The optimiser is also run only with the high-fidelity function to show the savings; finally the Macros SBO is used as the reference SBO method.

Convergence and final Pareto fronts The figure 7.11 presents the hypervolume indicator for each method, at all the multiobjective optimisations stages. The expected improvement method balances exploration and exploitation, the former one is directly beneficial to the initial Pareto front discovery: the best start is provided by the multifidelity expected improvement method up to 100 CFD calls. Macros SBO takes the lead from 150 CFD calls until its own convergence, at 250 CFD calls, explained by the specific behaviour of SBO: an initial design of experiments slows down the initial Pareto front exploitation, but then once the model provides accurate directions, the exploitation allows to discover the whole Pareto front; however, the SBO may only approximate the Pareto front. All multifidelity optimisations then reach the same coverage after 300 CFD calls. When the same optimiser (MOSPO) is used but only with the high-fidelity function, a significant time is initially spent to explore the design space while the particles learn. But once they find promising regions, the convergence indicator recovers to reach the same coverage than the multifidelity one between 330 to 500 CFD calls. Finally, it is worth noting that the surrogate Pareto front (the Pareto front made from all the non-dominated optimisations points) is never equal to the one from a single optimisation, meaning that all methods are complementary.

Each optimisation requires a significant time for the full convergence due to the significant time spent in CFD computations. In a large scale cluster, transfer and maintenance problems have made impossible to run the optimisation in a single run; optimisation restarts were therefore used but the elapsed time could not be measured accurately. Figure 7.11 shows therefore the complete CPU time for each optimisation. For the multifidelity runs, it takes into account the CFD (run on 8 cores) and the suboptimisation, including co-Kriging training and suboptimisation

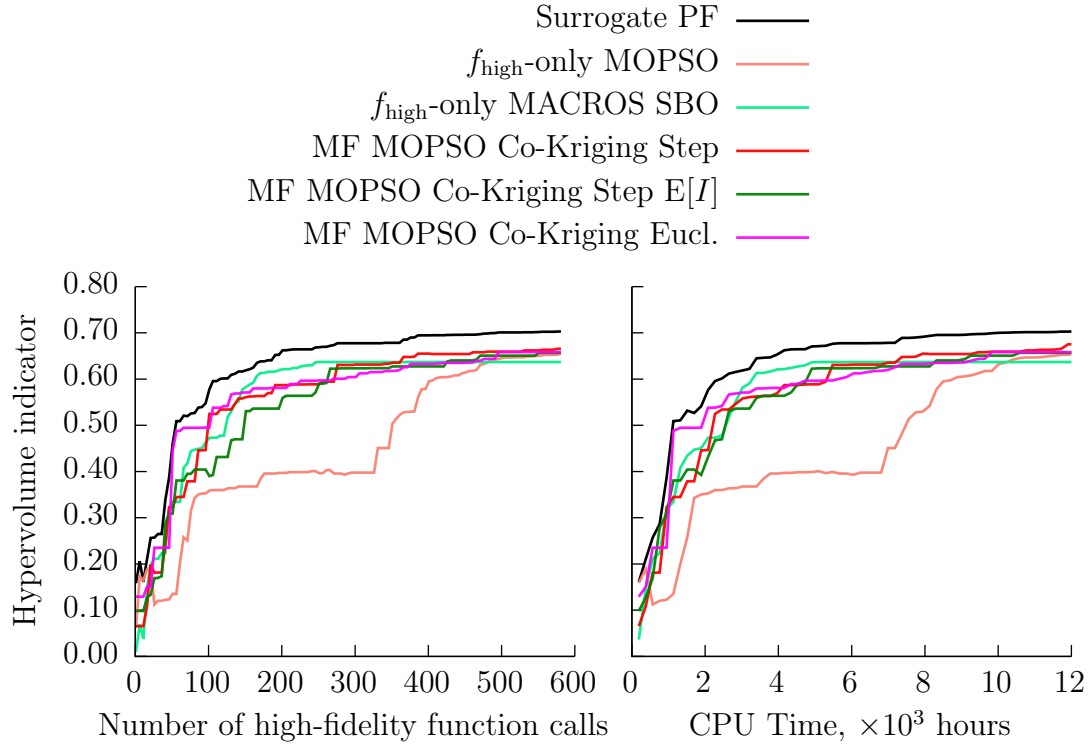


Figure 7.11: Hypervolume indicator convergence for the multiobjective multifidelity optimisation against the number of high-fidelity calls on the right and against the elapsed time on the left.

wall-clock times. Only the CPU time for each CFD calls is taken into account for the high-fidelity-only and Macros SBO. One aspect not shown with the CPU time is the capability to run in parallel CFD calls (that reduces the wall-clock time), but all the methods tested, except Macros SBO, can be set up to run in parallel. The hypervolume indicators convergence is the same when considering the CPU time: MOPSO calling only high-fidelity is much slower to reach the converged indicator than all other methods. Table 7.4 shows the CPU time savings to reach two different hypervolume indicators, when compared to the high-fidelity-only optimisation. The first indicator is achieved by all methods, and the commercial tool Macros SBO provides the best savings (50%) whereas the multifidelity methods provide savings between 20 to 30%. A second indicator is chosen for comparison, but not achieved by Macros SBO (having converged prematurely). Similar savings are observed by the multifidelity: between 15 to 25%.

Table 7.4: Savings provided by each methods compared to high-fidelity-only to achieve an hypervolume indicator of 0.64 and 0.66 for the number for the CPU time (industrial case).

Method		$h = 0.64$	$h = 0.66$
Macros	SBO	51.9%	-
MF	Direct	29.7%	24.2%
	E[I]	19.2%	15.2%
	Eucl.	19.2%	21.6%

To analyse more in depth the time spent in the suboptimisation versus the time required for evaluating the CFD, figure 7.12 shows for both multifidelity trust region definitions, the total time for each iteration with a breakdown between CFD and suboptimisation. Unlike the previous case, the CFD requires more time since a full polar is needed. The time spent in the suboptimisation is now small compared to the CFD time, and hence, despite a linear increase in the suboptimisation time (for model training and subproblem resolution), the total time remains constant, largely dominated by the CFD. On the utmost right hand side of figure 7.12 is shown the CPU time against the number of CFD calls for both the step-based multifidelity and high-fidelity-only optimisation. The linear curves show that although the multifidelity method needs to solve the subproblem, the time increment is negligible, and the CPU time required to perform 550 CFD calls is the same between the multifidelity and the high-fidelity only methods.

The number of CFD calls savings is greater than in terms of elapsed time for the academic case, whereas the industrial test case presents the same convergence and savings. First, the CFD calls in the industrial case are much larger and they represent the major part of the CPU time. Second, the figure 7.13 shows the suboptimisation elapsed time in function of the number of high-fidelity function points used for training. Clearly, for the academic test case, an increase in high and low-fidelity points has a linear effect on the training time whereas for the industrial case, this increase is closer to a logarithmic increase. The industrial test case is only using low-fidelity points evaluated in the first stages of the optimisation (when co-Kriging is not yet available) and during the high-fidelity function evaluation. Hence the number of low-fidelity points is lower in the industrial case which affects the matrices inversion both in the Kriging training for the low-fidelity estimator (part of the co-Kriging model) and the co-Kriging training.

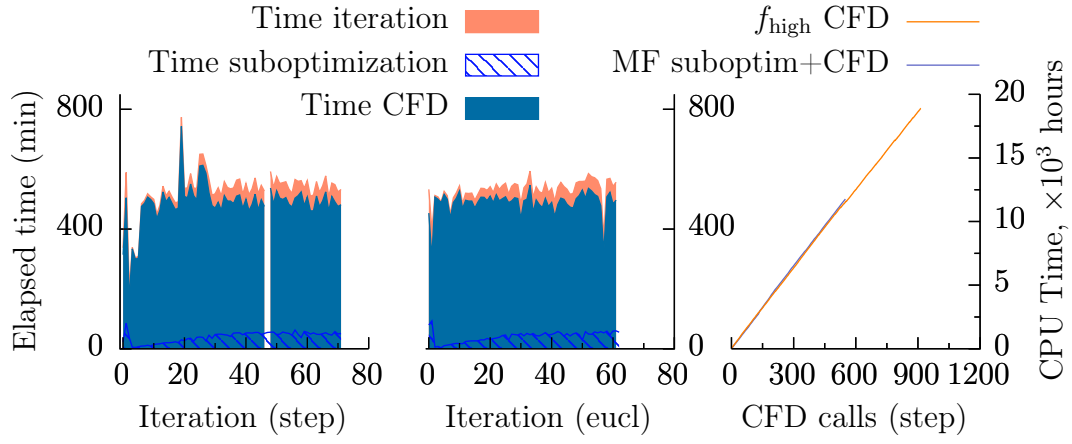


Figure 7.12: Time distribution for the multifidelity optimisation (left and middle) on the industrial case and comparison between CPU time and number of CFD calls for multifidelity (MF) and high-fidelity-only (right).

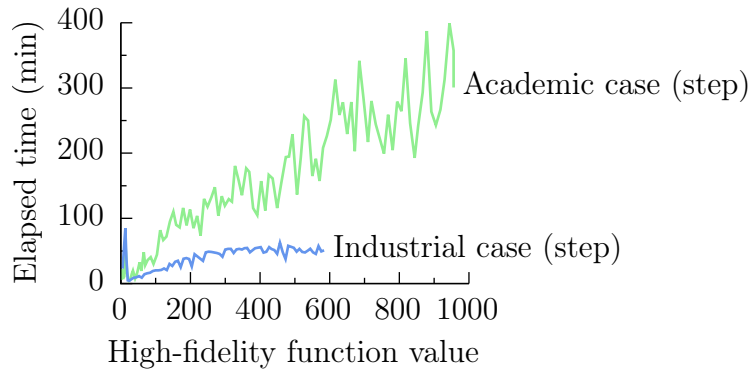


Figure 7.13: Suboptimisation wall-clock time comparison between academic and industrial cases.

The final Pareto front obtained after approximately 580 CFD calls is shown in figure 7.14. Macros did not reach this number of function calls as its convergence criteria was triggered after 250 CFD calls. All methods converge to a similar front, all improving the datum configuration. The largest discrepancy appears at the high- $c_{l_{\max}}$ region where the step-based direct¹ multifidelity finds the best maximum lift for the highest lift-over-drag ratio whereas Macros SBO finds the same maximum-lift but for a lower lift-over-drag ratio. The Pareto fronts discrepancies are also high in the rest of this region; but no method seems to be globally better, explaining the very similar hypervolume indicators. Overall, the direct multifidelity has the most advanced front albeit being dominated in the high- L/D region, by all methods.

Three Configurations Analysis Although all the optimisations have very similar anchor and compromise points, they are compared between each other in figure 7.15 (top) to confirm that the same flow physics is present. The parallel coordinates plot (Inselberg and Dimsdale 1991) shows the normalised values for each variables on the first 6 vertical axes, and the normalised objective values on the last two vertical axes. Each configuration is represented by a coloured line associated to an unique marker.

As expected, the $c_{l_{\max}}$ and L/D are inversely correlated and it exists a clear correlation between the flap deflection angle and the achieved maximum lift and lift-over-drag ratio. Similarly consistent is the slat deflection which is lower for the low-drag configurations and on the boundaries for the others, requiring more flap deflection. Only the multifidelity with standard error for the compromise point has a lower value despite having similar lift and lift-over-drag ratio. This lower slat deflection is alleviated by the slat gap which is higher than the other compromise points. Apart from this point, all the others methods have consistent slat gap with large gaps for large flap deflection, to get a fresher flow, and smaller gaps for the low flap deflection, to get more wake interactions. Another particularity to point out is the overlap for all configurations: no matter the lift or drag demanded, the overlap will always remain on (or close to) the lower boundary; this value is still positive and therefore the flap leading edge is still in front of the main element trailing edge. As for the single-objective optimisation performed on the small bounds, they should be increased to larger values to reach a better optimum although, it will have an impact on the kinematics size and complexity.

¹direct: use the co-Kriging estimator directly, without using expected improvements.

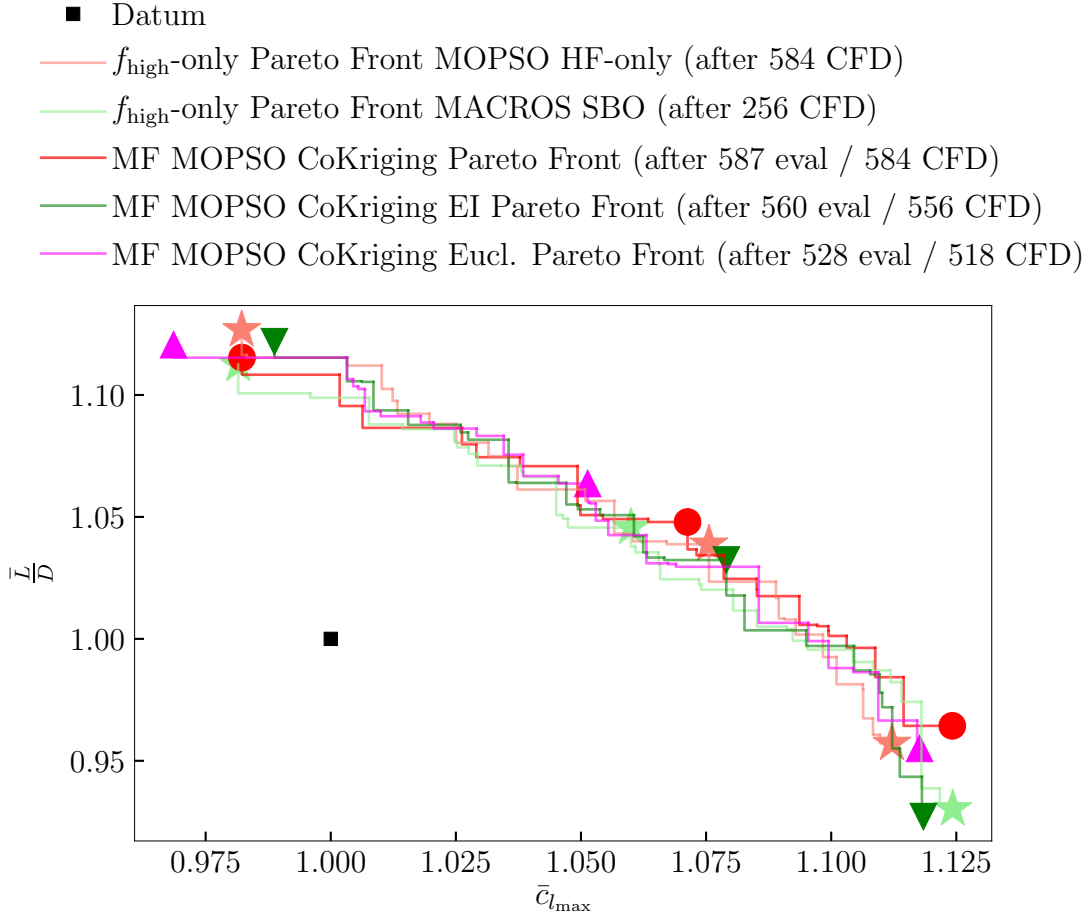


Figure 7.14: Multiobjective optimisation Pareto front comparison. *Eval* stands for number of high-fidelity calls, including points already run and taken from the database; *CFD* stands for number of CFD calls (expensive). Large markers are anchors and compromise points.

On the bottom of figure 7.15 is shown the deployed geometries for the different methods at the anchor and compromise points. All the positions are very consistent, and as in the parallel coordinates plot, only the gap varies between methods. Since the overlap is very consistent, the main variability is seen on the vertical positioning, but compared to the academic case, this variability is much smaller. Finally, the two test cases show the same trends for the high- c_l and compromise configurations: a small increase in the slat deflection to prepare the flow for the larger flap deflection, higher for the high-lift and a small increase for the compromise configuration. Both cases and configurations lead to the flap aft movement, much larger for the academic case. The flap low-drag configuration is not consistent: in the academic case, most of the solutions have a forward flap movement (increase of negative overlap) whereas in the industrial one, the movement is aft. This reflects the different second objective: the academic case was used to minimise the drag where the industrial case is used to minimise the lift-over-drag. Since a reduction in drag affects both objectives, the decrease in flap deflection is seen in both cases, but the lift requirement on the second case leads to a higher acceleration for a higher lift.

As a verification, the high- $c_{l_{\max}}$ configurations (in green in figure 7.15) can be compared to the single objective with small bounds optimisations, showed in figure 5.21 from section 5.6. The low overlap values are indeed consistent along with the slat and flap deflections which are the most sensitive variables for maximum lift. However the gaps positioning are different especially for the flap: for the multiobjective optimisation, they are spread over the design space whereas they are more concentrated in the upper values for the single-objective SB optimisations. A comparison between the $c_{l_{\max}}$ optima for the single-objective with small bounds and for the multiobjective optimisation is shown in figure 7.16. Since the optimisers had more iterations during the multiobjective optimisations hence more chance to explore the design space, the best maximum lift coefficient found by the multiobjective case is better than using the single-objective formulation; only MOPSO without multifidelity finds an objective slightly lower than the multifidelity single-objective with the Subplex. Let's now analyse the plot in the context of maximum lift prediction: as seen throughout the two test cases, the maximum lift estimation is a difficult problem for RANS. The accuracy reached is enough to predict the maximum lift coefficient with an accuracy of roughly 0.5. But the various optima shown in figure 7.16 are all within this value. Despite the small differences in numerical values, it is not expected to see differences in real conditions. All the solution can therefore be considered equivalent.

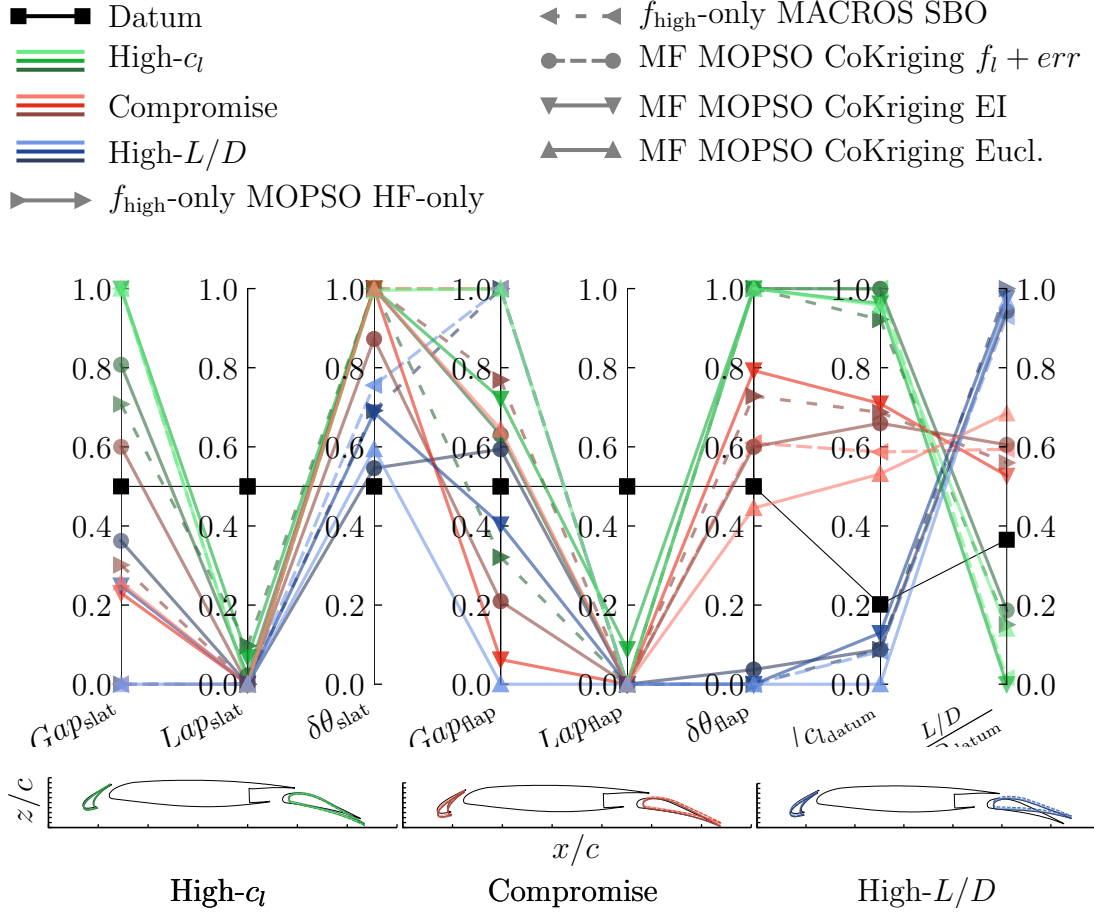


Figure 7.15: Parallel coordinates plot (top) and geometries (bottom) for the anchors and compromise points for each method.

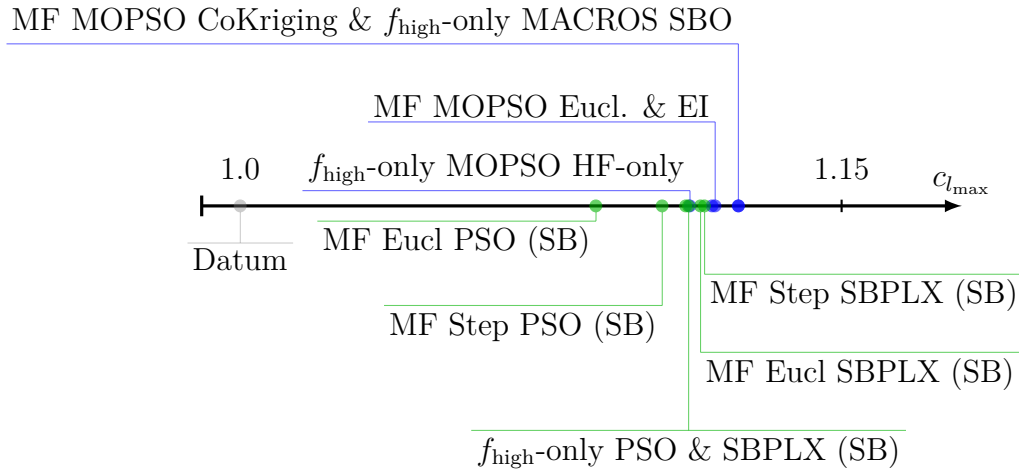


Figure 7.16: Normalised maximum lift coefficients comparison between multiobjective high- c_l (blue) and single-objective with small bounds (green).

7.5 Discussion

In this chapter we first showed that the use of a co-Kriging metamodel can improve the corrected function estimation when a sufficient number of points is available.

Secondly, the chapter is dedicated to the results update with the multifidelity co-Kriging method. The Pareto front obtained on the academic case is greatly improved, providing a more extended Pareto front with the same Pareto dominance than the high-fidelity-only optimisation. Despite a linear increase in training time (observed with respect to the number of high-fidelity points available), the multifidelity approach provides 50% time savings to reach the same dominated area than with the high-fidelity-only optimisation. Finally, the first test case successfully used the Euclidean trust region definition in a multiobjective context by using the co-Kriging model since its evaluation cost is low. The second test case provided a confirmation of this behaviour: the use of the co-Kriging metamodel gives between 15 to 30% CPU time savings compared to its high-fidelity only counterpart. The commercial Macros SBO method shows the same convergence speed, but the final Pareto front obtained is dominated by the multifidelity approach. As highlighted in chapter 4, SBO methods are multifidelity methods, but using a single source of information. Using a second model allows a better Pareto front.

Also, the conclusions from the single-objective chapter (chapter 5) are updated with the use of the co-Kriging correction. This multifidelity model within the trust-

region framework provides between 70% to 80% savings both in terms of number of high-fidelity calls and in terms of total elapsed time.

Finally, the two different cases optimisation results show the same trends in the devices placement but a change in the low-fidelity tool affects greatly the optima configurations between methods as seen in the chapter 5. However, the small differences in accuracy between the methods will not have any effect in real flight conditions due to the inaccuracies introduced by RANS at maximum lift conditions.

Influence of the Low-Fidelity Tool

Accuracy on Convergence

Multifidelity surrogate models such as co-Kriging assume that the low- and the high-fidelity functions are correlated in some way (M. C. Kennedy and O'Hagan 2001). Similarly, the trust-region based multifidelity optimisation strategy requires the low-fidelity function to be a sufficient approximation of the higher one in order for the surrogate model to correct the low-fidelity sensitivities to the high-fidelity ones (March and Willcox 2012b).

First, a visualisation method is introduced in section 8.1 to allow the function trends difference visualisation in a high dimension space. Then, the low-fidelity model used in the industrial test case is modified, first by changing the low-fidelity variables effect on its response (section 8.2). Secondly, various noise levels are added to the low-fidelity model in section 8.3. Those low-fidelity alterations will be used to quantify the low-fidelity model accuracy on the trust-region based multifidelity single objective method.

8.1 Trend Agreement Visualisation

Alexandrov, Lewis, et al. (2001) compare the low- and the high-fidelity functions (and constraints) shapes. Despite having a 15-dimension problem, the comparison is performed only on 2 variables, using two contour plots: one for the high-fidelity and one for the low-fidelity. The trends are very similar and are easily seen on the two plots. However, considering now a more complicated shape function, this comparison is no longer possible, especially if all dimensions need to be considered. Alexandrov, Lewis, et al. (2001) highlight as well that the absolute values (and a fortiori the optimum values) are not similar; only the shape is similar.

To combine two function plots in a single contour one, a visualisation technique needs to be found in order to show trends difference without considering the absolute values. First we consider multidimensional function contour plots and then we present a method to visualise the trends.

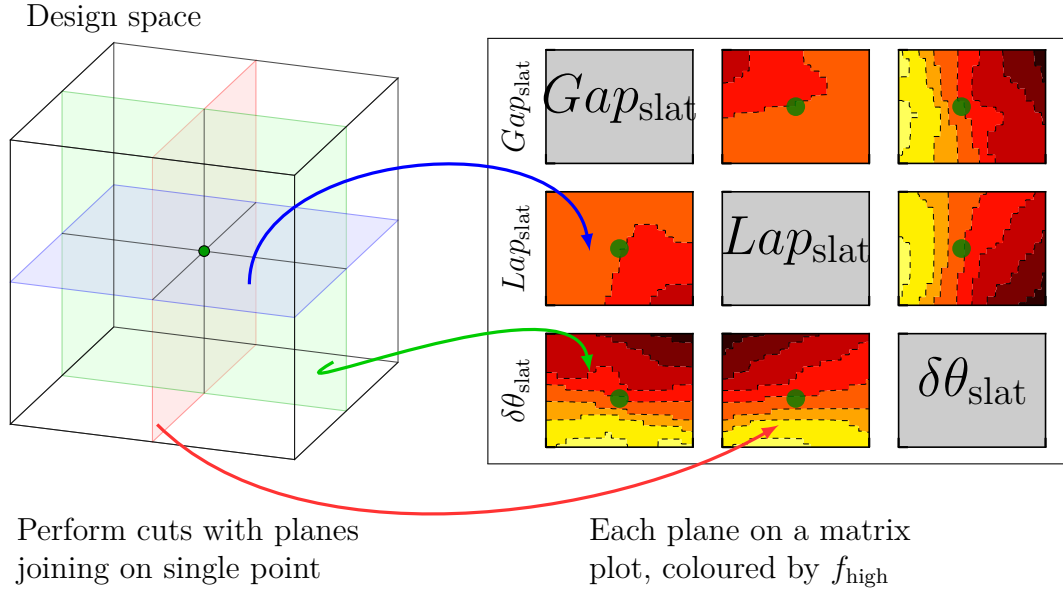


Figure 8.1: Three dimensional function with cuts organised in a contour matrix plot, coloured by high-fidelity function value.

8.1.1 Matrix Contour Plot

A contour plot is a representation of a three dimensional shape on a two-dimensional plane. It is usually used to represent a function response with respect to two inputs. An example of such plot has already been show in figure 5.19.

If more dimensions need to be represented, parallel coordinates plot is traditionally used, but they can only show the points in this high-dimensional space, without the visualisation of the function trends; correlations can be identified but the actual variations are hard to grasp. Another solution is through the use of two-dimensional cuts. A drawing of such solution is shown in figure 8.1: on the left is a three dimension space (only input variables are represented, the function value would use another dimension, hard to represent) which has been cut by three planes intersecting at a single point; on the right is represented each plane, with a contour plot of the function value. In this example, each subplot considers the variation of two inputs while the third one is fixed to the green point.

If we now consider the case studied, the high-fidelity function is expensive to evaluate, and therefore cannot be evaluated on a grid for each subplot. To overcome this problem, a surrogate model of the high-fidelity function is created and used to generate the contour plots. The only drawback of this method is the restriction of the

visualisation on the planes intersecting at the selected point, but this is inherent to high-dimension visualisation; only an interactive tool would alleviate this restriction.

This kind of plot, in addition of giving a representation of the function, also gives a quick overview of each variable sensitivity with respect to the output. Indeed, the normalised colour bars share the same range between all plots and therefore, when colours are varying greatly, the output is very sensitive to the inputs, and alternatively, when the colours are uniform, the inputs have no influence on the output; this is true only for the selected point however, and can vary at different cut locations. The contour matrix plot of the industrial high-fidelity function, namely the maximum c_l from TAU presented in section 5.6.1, is shown in figure 8.2. The six design variables are used and the colours are representing the maximum lift coefficient. The intersecting point is the datum configuration. According to the plot, the flap gap, slat lap and gap have minor influence on the maximum lift coefficient whereas the flap lap and deflection angle explain most of the function response variance.

Finally, it is important to remember that the contour plots are only as good as the surrogate model, and it becomes therefore important to validate the model to build confidence in the contour matrix plot.

8.1.2 Multidimensional Trend Visualisation

The contour matrix plots only give information regarding the output shape with respect to the inputs for a single function. If one wishes to see two functions, as for the application to multifidelity visualisation, two contour plots could be superimposed but this will quickly lead to esoteric plots with lines crossing anarchically and most of the visual information will be lost. This section focuses on the new visualisation technique to alleviate this problem by plotting the trends difference instead of the two function values.

Starting from two one-dimensional functions, on the left of figure 8.3 is showed the different situations that can be encountered when considering two functions trends and absolute values: 1. if the functions are similar, both the values and the trends (gradients) agree, 2. if the functions are different, both the values and trends can be expected to be different, 3. but similar values while different trends could also be encountered.

A multifidelity optimisation requires the low-fidelity function to give hints on the optimum location, or at least on the direction of the optimum region. Absolute values are not required to be similar, since the correction model easily correct this

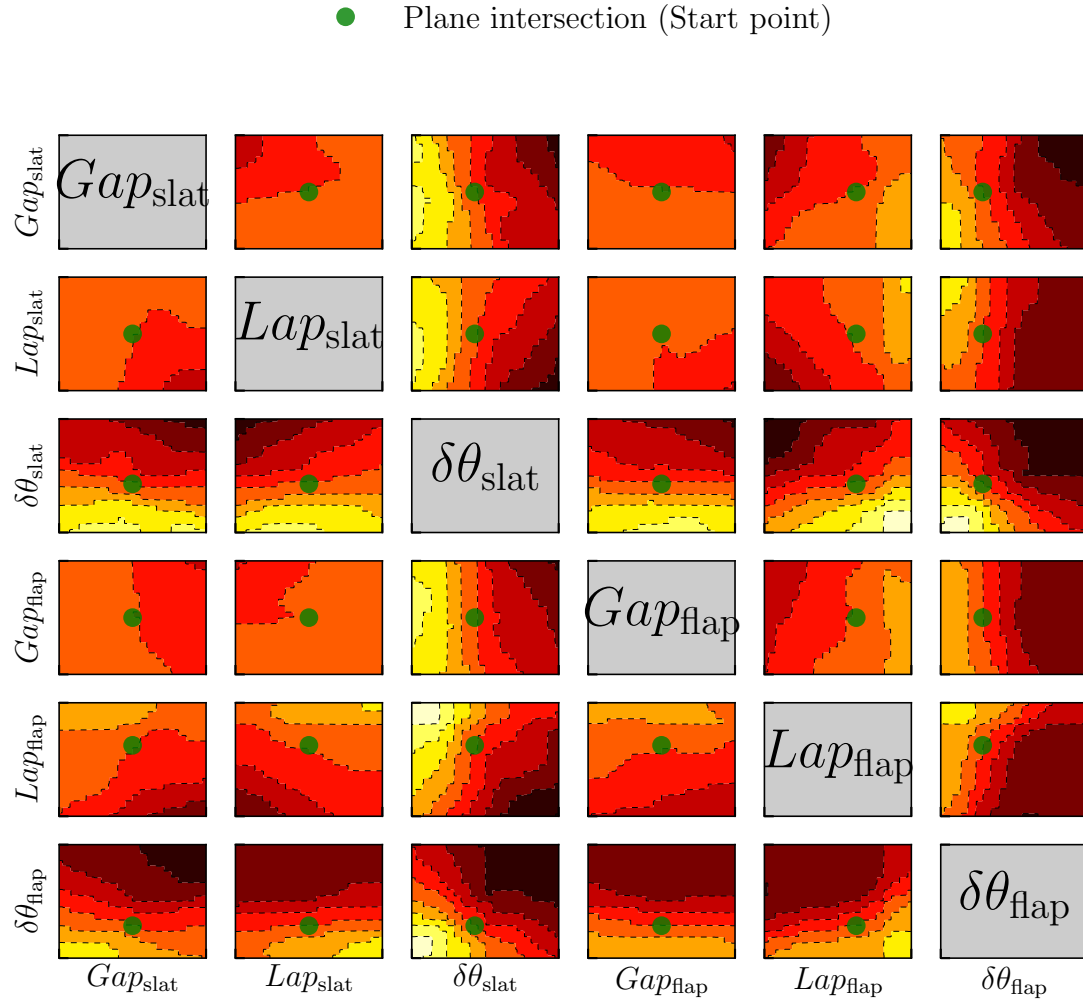


Figure 8.2: Contour matrix plot of the high-fidelity function (TAU) at the datum point.

constant error. Hence, to help visualising the difference between the high- and the low-fidelity functions, the tool should distinguish the trend differences, no matter the absolute value discrepancies. This is done by using the gradient sign at each discretised location in the contour plot: for each direction (since the contour plots have two input variables, two directions need to be considered) a marker is plotted if the gradient signs are equal. More formally, if the first variable is noted x and the second noted y in each subplot, a marker is plotted in the direction x (respectively y) if $\delta_x(x, y) = 1$ ($\delta_y(x, y) = 1$), otherwise no marker is plotted. δ_x (δ_y) is defined in equation (8.1) (equation (8.2)):

$$\delta_x(x, y) = \begin{cases} 1, & \text{if } \text{sign}\left(\frac{\partial f_{\text{high}}}{\partial x}(x, y)\right) = \text{sign}\left(\frac{\partial f_{\text{low}}}{\partial x}(x, y)\right), \\ 0, & \text{otherwise} \end{cases}, \quad (8.1)$$

$$\delta_y(x, y) = \begin{cases} 1, & \text{if } \text{sign}\left(\frac{\partial f_{\text{high}}}{\partial y}(x, y)\right) = \text{sign}\left(\frac{\partial f_{\text{low}}}{\partial y}(x, y)\right), \\ 0, & \text{otherwise} \end{cases}. \quad (8.2)$$

On the right of figure 8.3 is shown an example of such plot for a three-dimensional function (only considering the slat position). The contour plots represent the high-fidelity function while the markers represent the difference in trends between the low- and the high-fidelity functions. Areas where no markers are present show the regions where the low-fidelity function has opposite trends in both direction, areas with “+” markers show the regions where the low-fidelity function has the same trends as the high-fidelity one, regions with “-” show where the low-fidelity function has similar trends on the abscissa but opposite on the ordinate, and regions with “|” where the low-fidelity function trends are similar on the ordinate but opposite on the abscissa.

The full multifidelity contour plot for the functions presented in section 5.6.1 is shown in figure 8.4. The contour plots show the high-fidelity $c_{l_{\text{max}}}$ from TAU and the trends are compared with the low-fidelity prediction from HiLi. The gradients are obtained directly from the surrogate model (analytically differentiated) created with the commercial toolbox MACROS (see table 6.2 for model information). First, focusing on the slat, the graphs show that HiLi provides good trends with respect to the overlap through the whole design space, good trends except in the centre region with respect to the slat deflection but relatively bad trends when considering the gap, despite being a sensitive variable (with respect to TAU $c_{l_{\text{max}}}$ prediction). Clearly, the flap deflection is the most sensitive variable: it provides the greatest change in $c_{l_{\text{max}}}$ value. HiLi trends are as well the most accurate for this variable, especially when

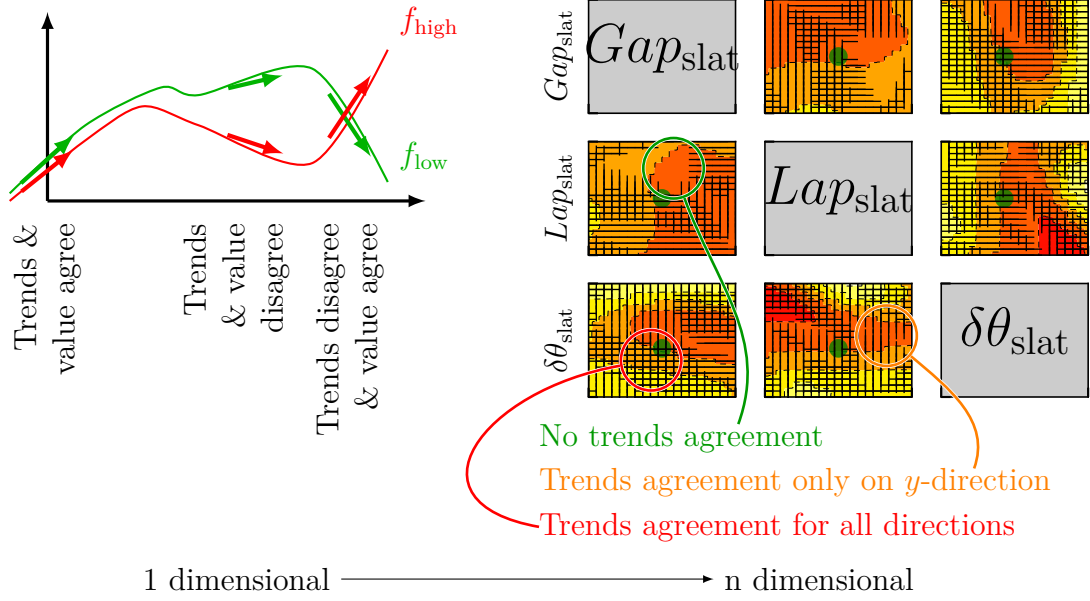


Figure 8.3: Multifidelity trends visualisation in high-dimension space and analogy with one-dimension function.

considering flap gap and overlap plots. The gap overlap, as for the slat, is not well predicted by HiLi, explained by the very small sensitivity. To conclude, HiLi seems to predict accurately the most sensitive variables.

8.2 Sensitivity Removal on the Low-Fidelity Model

To quantify the low-fidelity tool inaccuracies effect on the multifidelity method convergence, two modifications are tested in this chapter: inverting trends for some design variables and adding a smooth random noise in the low-fidelity response; this section deals with the first modification. As we saw in the previous section, close to the starting point, the low-fidelity tool has trends that differ from the high-fidelity one. Those differences are difficult to isolate but some variables seem more affected than others.

The following section presents the modification done to the low-fidelity model to simulate different sensitivities for different groups of input variables. Then the single-objective multifidelity optimisation is performed with the modified low-fidelity function. The case studied is the industrial one, presented in section 5.6.

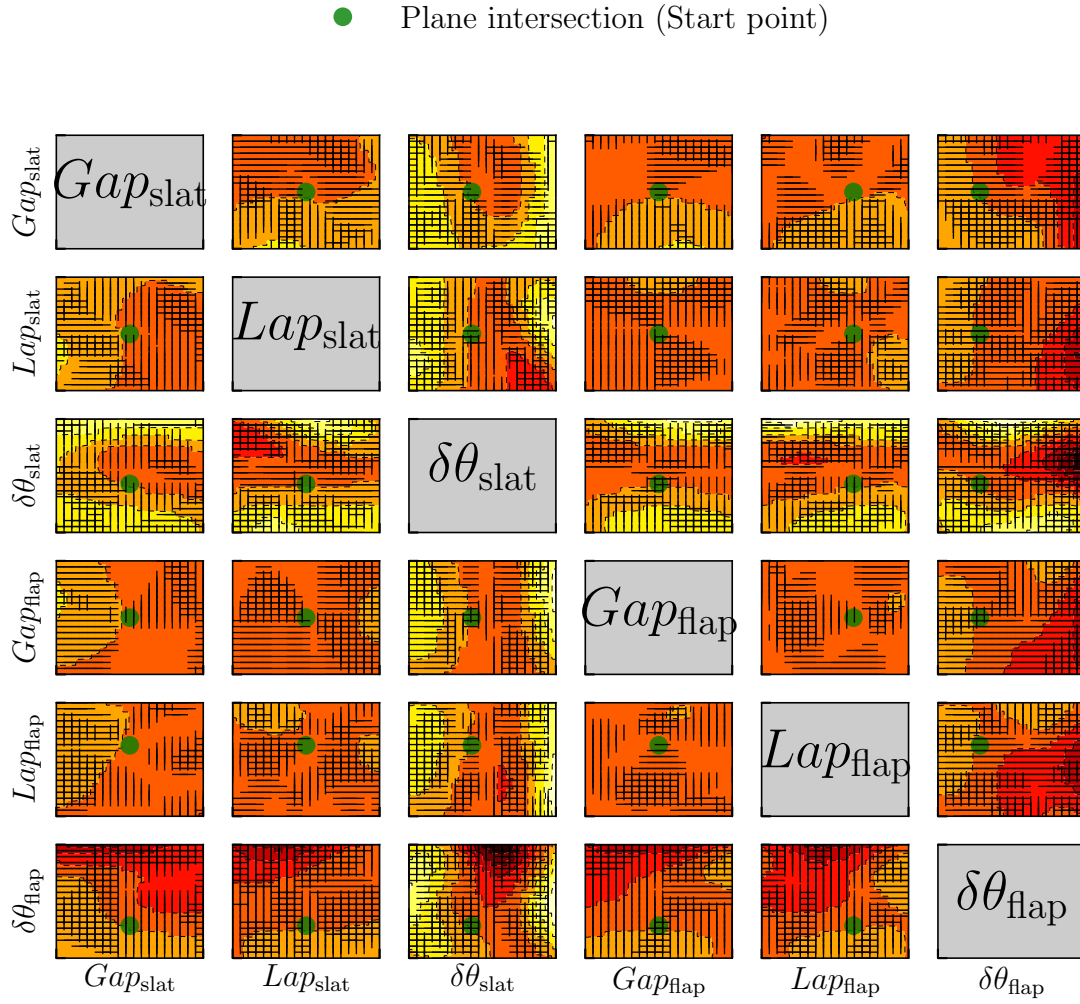


Figure 8.4: Multifidelity matrix contour plot for the maximum c_l predicted by TAU (high fidelity) and by HiLi (low-fidelity).

8.2.1 Modification on the Low-Fidelity Model

To simulate differences in the trends, the low-fidelity design variables are modified in two different ways:

Lack of sensitivity: the assumptions to the Navier-Stokes equations made in the lower fidelity models mean that some flow features cannot be resolved. It is thus sound to assume that some design variables will have less effect or no effect at all on the aerodynamic objective due to this simplification. To quantify this effect, the low-fidelity model is modified according to equation (8.3) for the slat and equation (8.4) for the flap:

$$\tilde{f}_{\text{low, ns}}(g_s, l_s, \theta_s, g_f, l_f, \theta_f) = f_{\text{low}}(g_{s_0}, l_{s_0}, \theta_{s_0}, g_f, l_f, \theta_f), \quad (8.3)$$

$$\tilde{f}_{\text{low, nf}}(g_s, l_s, \theta_s, g_f, l_f, \theta_f) = f_{\text{low}}(g_s, l_s, \theta_s, g_{f_0}, l_{f_0}, \theta_{f_0}); \quad (8.4)$$

with g the gap, l the overlap and θ the deflection angle; subscript s is relative to the slat, subscript f to the flap and subscript s_0 describes the slat initial variable value, and f_0 the initial flap position.

Reverse sensitivity: similarly, it can be found that some variables have an opposite action on the objective value between the low- and the high-fidelity functions. Thus, to quantify this effect, the low-fidelity function is modified according to equation (8.5) for the slat and equation (8.6) for the flap:

$$\tilde{f}_{\text{low, rs}}(g_s, l_s, \theta_s, g_f, l_f, \theta_f) = f_{\text{low}}(-g_s, -l_s, -\theta_s, g_f, l_f, \theta_f), \quad (8.5)$$

$$\tilde{f}_{\text{low, rf}}(g_s, l_s, \theta_s, g_f, l_f, \theta_f) = f_{\text{low}}(g_s, l_s, \theta_s, -g_f, -l_f, -\theta_f). \quad (8.6)$$

8.2.2 Visualisation

The multifidelity matrix contour plots for the modified low-fidelity tool are in appendix C.3. Interestingly, removing the slat effect (figure C.5) from the low-fidelity shows a better trend match on the matrix contour plots, especially in the low-sensitivity regions. It is important to remember however that those plots are highly dependent on the surrogate model quality: this may explain the difference of trends between the flap variables when the slat is fixed.

Reversing slat trends on the low-fidelity model (figure C.7) increases the trends discrepancies with the high-fidelity model, once again not only for the slat variables.

Figure C.6 shows the effect of removing the flap sensitivity. More regions with missing trends between the low- and the high-fidelity models are present but, due to the surrogate, some regions still have concordant sensitivity for the flap. A similar observation is done on figure C.8.

To conclude, altering the low-fidelity model is seen through the multifidelity matrix contour plots but due to the surrogate modelling, some of the changes are not perceived in those plots. However, the fidelity is decreased and an effect on the multifidelity optimisation should be seen.

8.2.3 Multifidelity Optimisation Results

The multifidelity is first run with a non-modified low-fidelity model. The optimisation is as well performed with the high-fidelity function only to provide a reference. The multifidelity optimisation is then run with the modification presented in section 8.2: the figure 8.5 shows the convergences. The problem boundaries are quite large, and the low-fidelity model must be used in configuration for which it was not intended to. This is reflected in the convergence of the normal multifidelity that is, at first faster than the high-fidelity-only optimisation but then, slows down.

Focusing on the no- and reverse-slat movement effects, the convergence history clearly shows that reversing the slat has a negative effect whereas, not using the slat information from the low-fidelity is beneficial in terms of convergence speed and optimum value. To explain this behaviour, we need to look at the suboptimisation behaviour. At the first iteration, when only one point is sampled, the Subplex is performed only on the low-fidelity. In the absence of information on the slat movement, the suboptimisation only varies the flap position, which has the greatest effect on the objective function ($c_{l_{\max}}$). However, when the slat effect is reversed, the Subplex focuses on optimizing the slat, without changing the flap position. This behaviour is repeated throughout the following iteration up to iteration 6 when a sufficient number of points is available to reverse the corrected function trends thanks to the correction. The following iterations deflect the flap giving a large high-fidelity function decrease, conform to the slat deflection found previously. However, the deflection is so high that the flow is mostly separated leading to a loss of maximum lift compared to the other optimisation cases: figure 8.6 presents this optimal configuration in a parallel-coordinates plot. Alternatively, the lack of sensitivity on the slat pushed

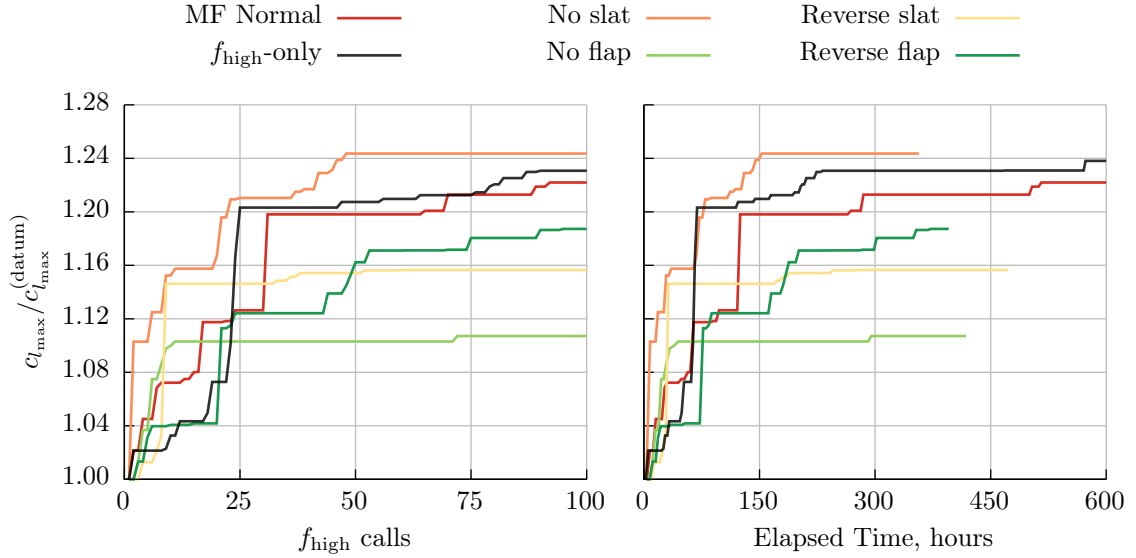


Figure 8.5: Normalised convergence with respect to the number of high-fidelity function calls (left) and to the wall-clock time (right) for the sensitivity modifications on the low-fidelity model.

the trust-region methods to perform as a traditional Surrogate Based Optimisation (SBO), progressively moving the slat according to the enhanced surrogate model prediction.

Modifying the flap sensitivity however, shows a different behaviour. Since the flap, and more specifically, the flap deflection is the most sensitive variable with respect to the maximum lift coefficient, altering the low-fidelity model is detrimental to the convergence in both cases. Nonetheless, removing the flap movement sensitivity improves the initial multifidelity optimisation convergence mainly because the problem dimension is reduced (the optimiser will not vary a variable that has no effect) and the best slat position for the initial flap deflection is found quickly. Due to the nature of the Subplex that performs multiple simplexes on variables subset and due to the inner working of the step-based trust region optimiser, the flap position during the *Normal* multifidelity optimisation is rarely varied until iteration 28. Only the flap gap is varied, giving a slight advantage to the *normal* run compared the *no flap* one. However, not having any prior sampling and only relying on the optimisation path for the correction model training, especially when the low-fidelity lack sensitivity for some variables, shows its limitation: the *no flap* flap lap and deflection variables

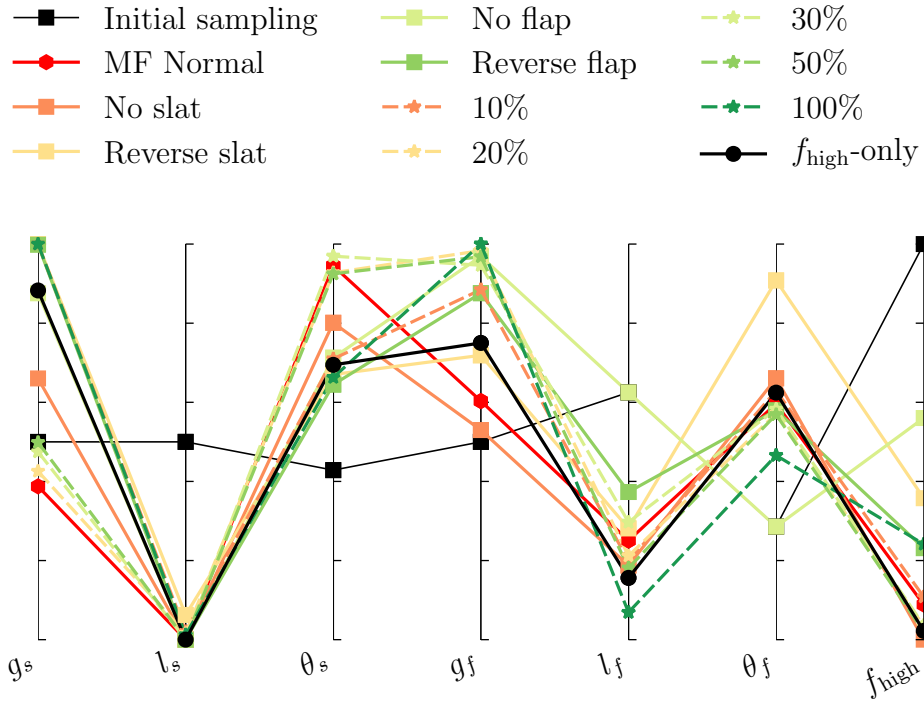


Figure 8.6: Parallel-coordinates plot showing the normalised optimum for each optimisation.

are not varied throughout the optimisation (see figure 8.6). Using a more advanced technique such as the expected improvement (Jones, Schonlau, and William J. Welch 1998) would be beneficial here as it would help to sample the underexplored regions. If the low-fidelity model has incorrect sensitivities, the convergence is slower since the optimiser must “*make mistakes*” in order to reverse the sensitivity. Despite the convergence not being complete, the convergence history shows that the multifidelity with reversed flap converges to better values than the reverse slat, probably due to the higher sensitivity of the flap leading to a better exploration of the flap movement than of the slat one.

To summarise, no matter if the low-fidelity has opposite or no trends at all, its effect on the optimisation depends on the variable considered. The most sensitive variables must be well captured by the low-fidelity: a lack of sensitivity will lead to an early convergence whereas a wrong sensitivity will result in a slower convergence. If the less sensitive variables are not correctly captured, it seems that no sensitivity at all is beneficial since the correction will act as a surrogate model.

8.3 Noise Addition on the Low-Fidelity Model

MSES for example requires a full convergence to have a meaningful result. Similarly, Euler, or RANS solutions require a reasonable convergence to have an accurate result. If it is not achieved, the solution is parasitised by numerical error; despite not being random since the computer code is deterministic, it could be considered as a noise when the results are used within a surrogate and/or an optimisation process (Giunta et al. 1994). Therefore, to quantify the *noise* effect on the convergence of the multifidelity optimisation, an artificial noise is added to the low-fidelity objective space according to the following process.

8.3.1 Modification on the Low-Fidelity Model

In order to get a smooth noise, a set of $\tilde{n}_{cp}$ control points $\{\mathbf{c}_i, i \in [1, \dots, \tilde{n}_{cp}]\}$ are selected using a Latin Hypercube Sampling (LHS) over the design space (Lian, Oyama, and Liou 2010). For each control point, a random number r_i is generated between $-\zeta$ and ζ from a uniform distribution as in equation (8.7):

$$r_i = -2\zeta\mathcal{U}(\mathbf{c}_i) - \zeta, i \in [1, \dots, \tilde{n}_{cp}]. \quad (8.7)$$

Then, to have a continuous noise function $\eta(\mathbf{x})$, a RBF with a linear kernel is fitted on the control points. The same method as in section 3.3.3 is used with the kernel function $\phi(\|\mathbf{x}\|)$ being defined in equation (8.8):

$$\phi(\|\mathbf{x}\|) = \|\mathbf{x}\|. \quad (8.8)$$

Finally, the modified low-fidelity function $\tilde{f}_{\text{low},\zeta}$ is defined by equation (8.9):

$$\tilde{f}_{\text{low},\zeta}(g_s, l_s, \theta_s, g_f, l_f, \theta_f) = [1 + \eta(g_s, l_s, \theta_s, g_f, l_f, \theta_f)] f_{\text{low}}(g_s, l_s, \theta_s, g_f, l_f, \theta_f), \quad (8.9)$$

giving a modified low-fidelity function within the following bounds:

$$\tilde{f}_{\text{low},\zeta} \in [f_{\text{low}} - \zeta f_{\text{low}}, \dots, f_{\text{low}} + \zeta f_{\text{low}}]. \quad (8.10)$$

The noise generated being random and affecting the whole design space, the multifidelity contour matrix plots are not used for this modification.

8.3.2 Multifidelity Optimisation Results

The second study consists in adding a random noise on the low-fidelity function as to simulate computational noise arising from badly converged simulations or discretization error. Figure 8.7 shows the convergence for different levels of noise as defined in section 8.3. The same *normal* and f_{high} -only convergence histories as in the previous section are shown. All noisy low-fidelity runs perform better than the *normal* multifidelity run, except for the highest level of noise, representing noise level as high as the low-fidelity response. When the noise levels are relevant (lower than the response amplitude, less than 50%), the low-fidelity function pushes the trust region optimisation to explore more the design space, which allows a better optimum and a quicker exploration.

8.4 Discussion

A summary of the results are presented in figure 8.8. To quantify as well the convergence speed differences, the area between the reference multifidelity run (*normal*) and the other runs are computed. It is clearly visible that the noise is beneficial to the optimisation until a level of 50%, both in terms of convergence speed and

optimum accuracy. However the trends in the effect from the sensitivity change is harder to highlight: we can only encourage the trends to be as close as possible to the high-fidelity one to help the convergence.

Trust-region multifidelity methods are dependent on the accuracy of the low-fidelity model: the final optimum and the convergence speed is affected by its accuracy relative to the high-fidelity one. This chapter aimed at quantifying the effect of inaccuracies from the low-fidelity on the final optimum and its convergence speed. Two aspects are proposed: different sensitivities from certain variables between the low- and the high-fidelity models and random noise in the low-fidelity model. The first approach aims at quantifying the effect of removing or reversing the sensitivity of a group of variables on the low-fidelity model response in order to model a wrong prediction of the flow physics, while the second aims at quantifying the effect of noise emanating from bad convergence or bad discretisation of the low-fidelity model.

The study showed that depending on the group of variables modified, the effect did not induce the same negative effect on the convergence. For the highest sensitive variables, a lack of sensitivity will result in an early convergence since the design space won't be explored and hence won't be corrected while a wrong sensitivity will result in a slower convergence, since the correction will take time to be trained. If the less sensitive variables are not correctly captured, it seems that no sensitivity at all is beneficial since the correction will act as a SBO.

Regarding the effect of noise on the low-fidelity model, adding a relative small noise (relative to the response) will not result in a bad convergence: on the contrary, the convergence will be improved since the exploration of the design space will be facilitated leading to a better correction. The convergence will be worsen only if the noise becomes comparable to the response in terms of amplitude.

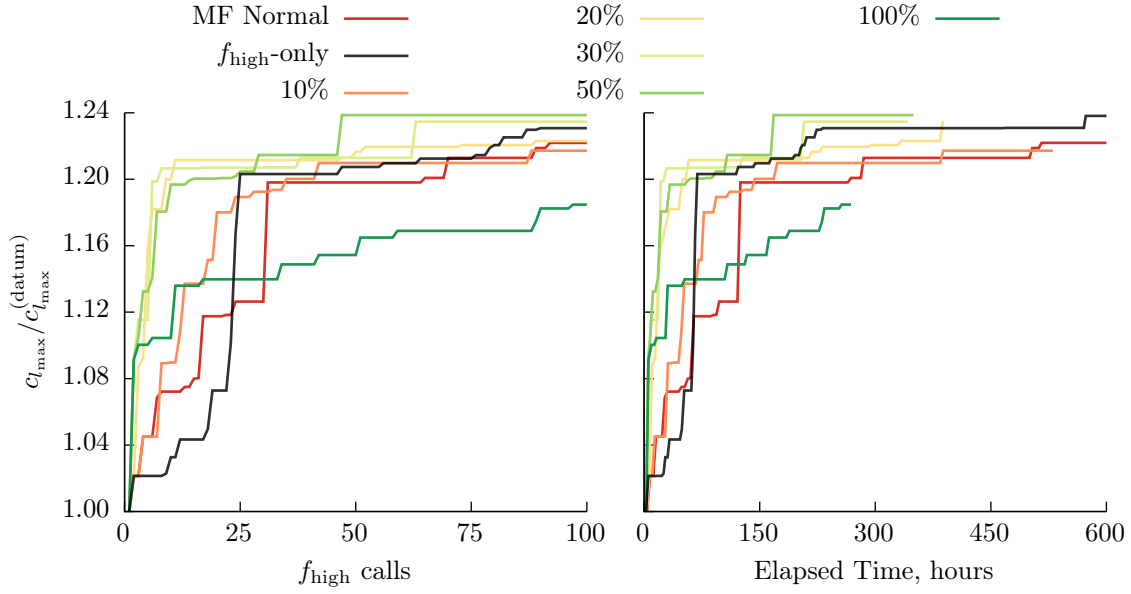


Figure 8.7: Normalised convergence with respect to the number of high-fidelity function calls (left) and to the wall-clock time (right) for the noise modifications on the low-fidelity model.

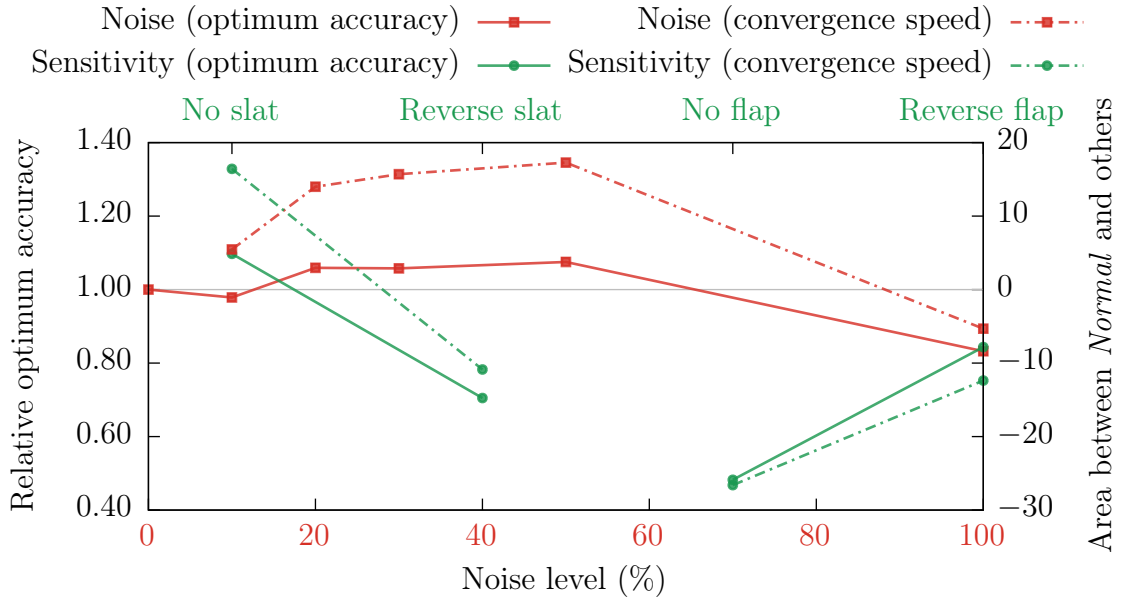


Figure 8.8: Optimum accuracy and convergence speed evolution with respect to the sensitivity and the noise level.

Three-Dimensional Transonic Wing Optimisation

The previous chapters were dedicated to low-speed aerodynamics on a two dimensional aerofoil. However the methodology is not only bound to this area and can be applied to any problem. During the time at the sponsor, some interest was shown for the application of multifidelity optimisation to a three-dimensional geometry (3D). However, 3D high-lift wing involves many intersecting parts when moved (between each other or with other components such as the fuselage) and the transformation from the CAD to the mesh requires a lot of manual work. Similarly, the mesh requires the intervention of a designer, especially when the geometry topology changes for different positions. Therefore the 3D application is done on a transonic wing, without any fuselage nor engines. This chapter does not aim at extracting information from an optimised wing shape but rather at analysing the effect of using such test case on the multifidelity framework. The geometry and optimisation problem are defined in section 9.1 and the high-fidelity only and multifidelity optimisation results are presented in section 9.2, first for the Kriging correction and then for the co-Kriging one. Conclusions on this case are finally drawn in section 9.3.

9.1 Geometry, Parametrisation and Objective

The geometry used in this chapter is the XRF1¹ wing which is used as a common research model inside Airbus and his partners.

The goal of this chapter is the wing twist optimisation to minimise the total drag at cruise speed. The flow is therefore transonic and the flow parameters are described in table 9.1. To prevent the optimiser from reducing the lift to reduce the drag, the lift must be constrained. This is achieved by a penalty method as the framework does not handle constraints yet in a more integrated manner. Openings are given in the future work chapter (chapter 10) to include constraints within the trust-region based algorithm. The objective, in our case the drag, is penalised where the constraint is

¹eXternal Research Forum: including Airbus, DLR, ONERA and QinetiQ (Ghazlane 2012).

Table 9.1: Aerodynamic parameters for the XRF1 wing.

Parameters	Value
Reynolds number	$6.5 \cdot 10^6$
Mach number	0.83
Angle of attack	3.02 deg

not ensured. Therefore the objective is defined in equation (9.1), with a targeted lift of 0.5. The lift to constraint is directly used in the penalty function providing feasibility information and, more importantly, keeping the function continuous, a mandatory requirement for the error correction. The constraint weight is selected to put sufficient emphasis on the targeted lift considering the accuracy of the CFD. The use of the power of two also ensures that the penalty rises quickly as the constraint is violated, to prevent constraint violation.

$$f_{\text{so}} = C_d + 10^5(C_l - 0.5)^2. \quad (9.1)$$

In order to keep a smooth twist through the wing and to keep the number of variables low, a twist law is used: only 4 variables are used to control the 8 wing sections. The top-view geometry along with the twist parametrisation are shown in figure 9.1. The initial and boundary values are shown in table 9.2. The bounds are purposely taken large to prevent any restrictions.

In the previous chapters, different models for the flow physics were used for the low- and the high-fidelity functions. In the literature, multifidelity encompasses as well the use of the same model but with some relaxed criteria to reduce its computational cost, entitled *numerical accuracy* in figure 4.1. As an example, Dalle and Fidkowski (2014) uses a coarser mesh discretisation for the low-fidelity model. This is the choice used in this chapter.

Both models are relying on the RANS model solved by the structured solver *elsA* used in Airbus (Cambier and Gazaix 2002) with the one-equation Spalart-Allmaras turbulence model. The solver uses a structured mesh and because the twist movements are relatively small compared to the wing size, a mesh deformation is used to produce the new candidate geometry mesh. Both coarse and fine meshes are run with a multigrid acceleration and for 5,000 iterations². The high-fidelity model uses a mesh of 5,505,835 points and requires 3 hours and 40 min to run on 24 cores. The

²Enough to provide convergence at 10^{-4} for C_l .

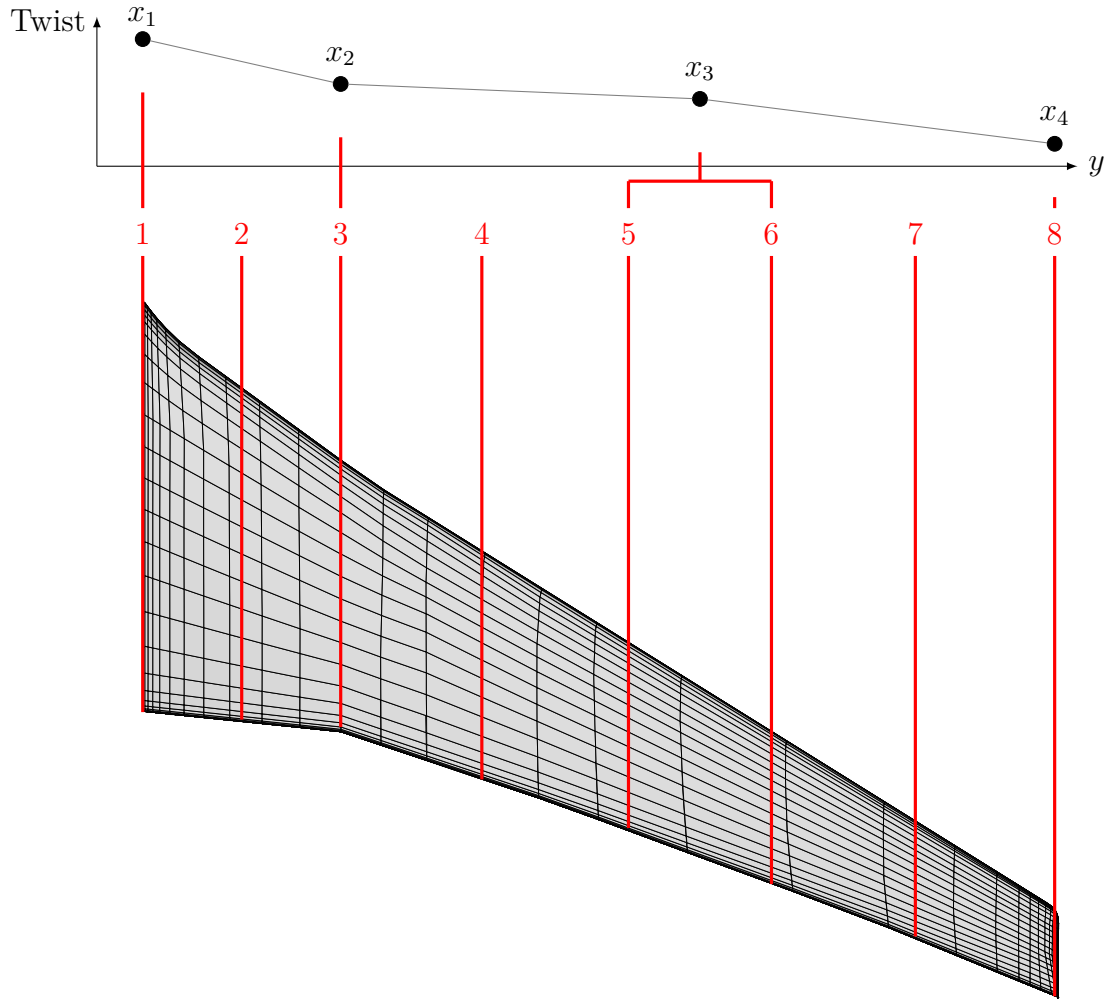


Figure 9.1: XRF1 coarse mesh top-view with the twisted section in red, the twist law variables x_i in black in an example configuration.

Table 9.2: Initial and boundaries values for the XRF1 test case.

Variable	Min (deg)	Initial (deg)	Max (deg)
x_1	-10.0	0.0	10.0
x_2	-10.0	0.0	10.0
x_3	-10.0	0.0	10.0
x_4	-10.0	0.0	10.0

Table 9.3: Datum integrated aerodynamic coefficients for both meshes. The drag is given in drag counts.

Coefficient	Fine	Coarse
C_L	0.499	0.499
C_D	216	237

low-fidelity model uses a mesh of only 252,928 points but runs only in 5 minutes and 38 seconds on 24 cores due to a better mesh balance on the cores. Hence, the low-fidelity function, despite using the same model, is around 40 times quicker and should provide the correct trends at first but be less accurate when reaching the optimum. The meshes top-view are shown in figure C.9. The initial lift and drag coefficients are given in table 9.3. Both meshes compute the same lift, but the drag, calculated with a farfield approach, is different. This is expected since larger cells affect the numerical cell diffusion which has a direct effect on the downstream pressure.

The multifidelity framework considers the low- and the high-fidelity functions as black box: the proven Airbus process handling the different steps. From the design variables, the process generates a geometry with an in-house kernel, and the differences between the initial and the new geometries are computed and fed to the mesh deformation tool. The mesh fitted on the new geometry is then used by elsA to solve the RANS equations. Finally the post-processing returns the integrated lift from the pressure integration and the wing total drag from a farfield approach. The test case is a wing created for academic partners, and the aerodynamic simulations fully validated by Airbus. The exercise intended here is to consider the low- and the high-fidelity models as pure black-boxes, that take the wing twist distribution as input and return the total integrated lift and drag.

9.2 Results

Although the Airbus framework is fully differentiated, the multifidelity framework was not developed to use gradients. The optimisation is therefore performed without gradients. As a reference method, the Nelder Simplex method, already presented in section 3.2 is used. Initially, the single-objective multifidelity framework is used with a Kriging model providing the error correction (section 9.2.1). As a second step, the improved multiobjective co-Kriging framework is used with a single objective in section 9.2.2.

Table 9.4: Multifidelity parameters for the initial XRF1 optimisation.

Parameters	Values
Trust-region	Step
Initial TR	2
TR expend	1
TR shrink	1
Suboptimiser	Simplex
Initial variables (all)	0 deg
Bounds (all)	−10 / 10 deg

9.2.1 Single Objective Framework

The single-objective framework is used with a Kriging correction. The suboptimisation is, as for the high-fidelity-only optimisation, undertaken by the Simplex, and the trust-region defined in number of steps. The remaining parameters are in table 9.4.

Figure 9.2 shows the multifidelity convergence compared to the high-fidelity-only one. It is clear that the multifidelity optimisation does not provide the same improvements. To understand this limitation, the figure 9.3 shows the corrected function evolution during each suboptimisation. For better distinction a different colour is used between even and odd global iterations. By definition, the initial point of each suboptimisation is exact, that is, it has the same value than the high-fidelity function. Furthermore, the same figure presents the low-fidelity function evolution for the same suboptimisations. The first iteration provides most of the improvements: the low-fidelity function value decreases, and the high-fidelity function improves. It is clear that, at first, the low-fidelity function provides good information to decrease the objective. The second iteration provides as well a similar convergence, and the final corrected function is very close to the true high-fidelity function. However, from the third suboptimisation, the correction function decreases but the low-fidelity function increases: the correction inverses the low-fidelity trends in the vicinity of the low-fidelity optimum. During the subsequent subiterations, the corrected function varies more than the low-fidelity until the last iterations. On the left side of figure 9.3 is shown the error distribution (between high- and low-fidelity functions at sampling points): the error does not show high variations, partly explained by the proximity of each suboptimum but also because the design space location is smooth in the region close to the datum, where lift constraint is fulfilled.

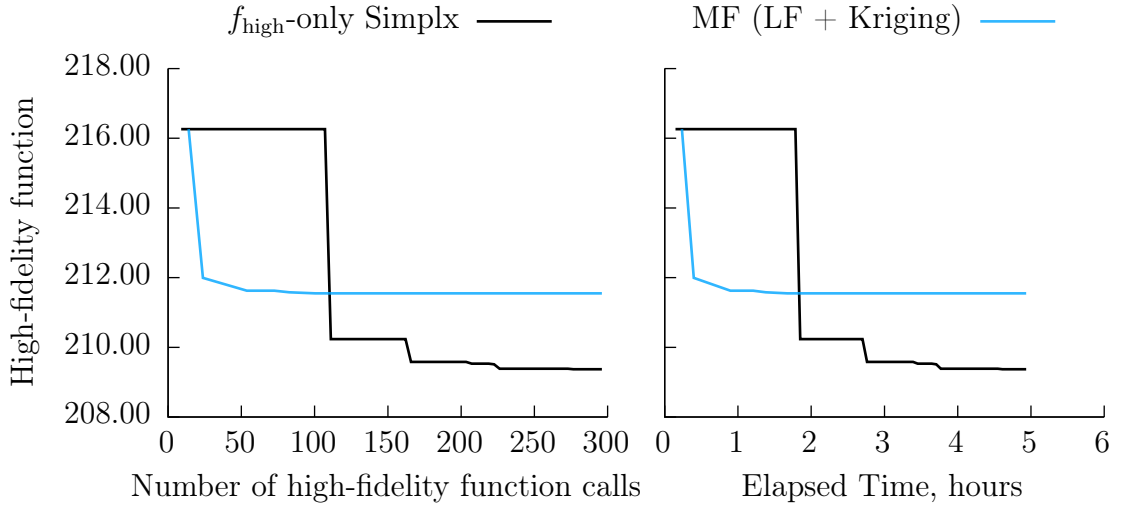


Figure 9.2: Convergence comparison between multifidelity (single-objective formulation) and high-fidelity-only XRF1 optimisations with respect to the number of iterations on the left and with respect to the elapsed time on the right.

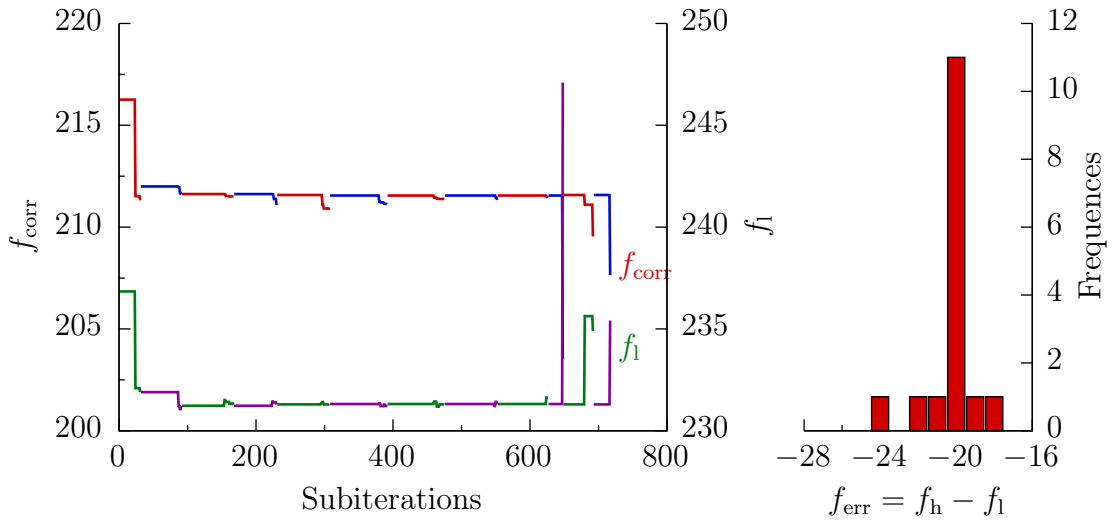


Figure 9.3: Suboptimisation convergence during multifidelity (single-objective formulation) XRF1 optimisation. Colours for global iterations alternate and both corrected and low-fidelity functions are shown. On the right hand side: sampling point error distribution.

To finish with the multifidelity analysis, the parallel coordinates plot showing the high- and multifidelity convergences is presented in figure 9.4. When the lift constraint is not respected, the objective value increases sharply. Only the high-fidelity-only evaluated such points, and those high values are clipped (only on f_{high} axis) to allow comparison with the multifidelity ones. The multifidelity approach avoids high-fidelity function evaluation in infeasible regions or regions of low interest. Focusing first on the high-fidelity convergence, the lowest objective function values are reached for the high numbers of iterations, as expected with a single-fidelity optimiser. The design variables were spread between -5 to 5 degrees but the converged ones prescribe a twist down for the first control point, almost no change for the second and third and a relatively large twist up for the last section; the explanation of such change are presented in section 9.2.2.

Focusing secondly on the multifidelity, the twist law is inverted, except for the tip that is also twisted up, but in a smaller extent. The multifidelity optimisation however only focuses on the same region, and no high-fidelity evaluations are done that result in high objective values. Because of the penalty imposed on the lift, changing design space region is difficult as it will most probably lead to a change in lift and therefore a large increase in the objective value. This is exacerbated by the trust-region since any point that does not lead to an improvement of the high-fidelity function is rejected. The improvements obtained in section 7 may be beneficial to increase the exploration despite the penalty constraint. The remaining of this section focuses on using the multiobjective framework, along with the co-Kriging model, but using a single objective.

9.2.2 Multiobjective Framework

The objective function has a complex shape mostly because of the lift penalty constraint. Therefore to increase the global exploration limited by the trust region with the Kriging correction, the multiobjective framework is used in this section. This framework mostly brings the co-Kriging model within the trust-region framework. Not considering tractability, any number of objectives can be specified in the framework, including a single one. The common parameters for all multifidelity runs are shown in table 9.5.

The optimisations convergence are displayed in figure 9.5, for different formulations of the multifidelity optimiser; the high-fidelity-only optimisation is the same as previously. First, in green, is displayed the multifidelity optimisation convergence

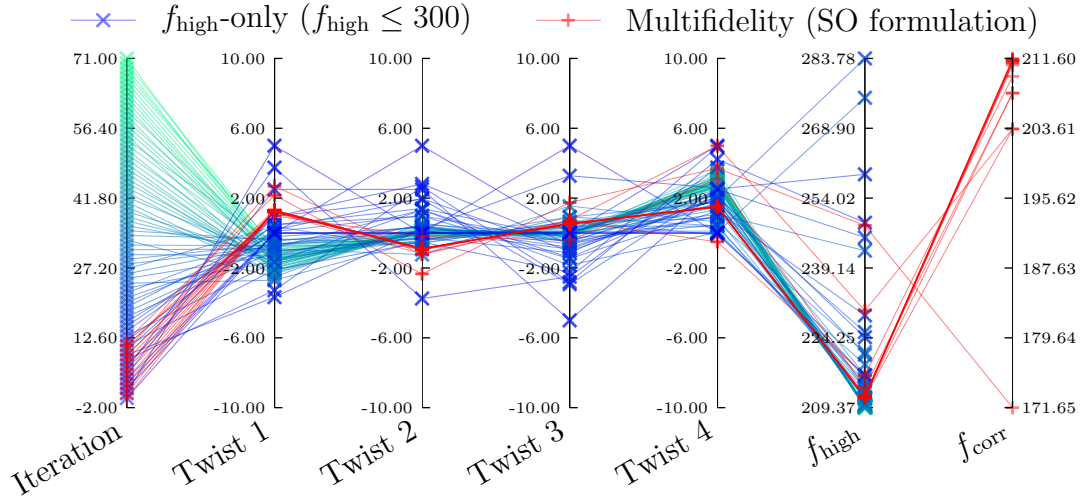


Figure 9.4: Parallel coordinates plot for single- and multifidelity XRF1 optimisation convergences. High-fidelity-only is coloured by iteration number and f_{high} axis is clipped at 300.

Table 9.5: Multiobjective multifidelity framework parameters for the XRF1 single-objective optimisation.

Parameters	Values
Trust-region	Step
Correction	co-Kriging
Initial TR	2
TR expend	1
TR shrink	1
Number of objectives	1
Number of points for successful iteration	1
Suboptimiser	Simplex
Initial variables (all)	0 deg
Bounds (all)	-10 / 10 deg

when both the co-Kriging and the low-fidelity models are used. As presented in the chapter 6, the co-Kriging model is used only when a sufficient number of points is available, in this case, that is after 9 high-fidelity points. To solely use the co-Kriging, an initial sampling of 10 high-fidelity and 60 low-fidelity points is performed prior starting the optimisation. Despite the objective function not representing a physical quantity, the combined use of co-Kriging and low-fidelity information provides the same optimum as the high-fidelity-only optimisation but in a reduced number of iterations. The elapsed time however is greater since in addition to the co-Kriging expensive training, the low-fidelity model needs to be evaluated at each suboptimisation candidate points.

To quantify the effect of the initial sampling, the next two runs are done without (orange) and with (blue) an initial sampling of 10 high-fidelity and 60 low-fidelity calls. First, when no sampling is provided, only the Kriging model is used, the same convergence as in the previous section is observed: the trust region converges quickly to the low-fidelity minimum and cannot escape, the exploration being very limited. Adding an initial sampling unlock the convergence, and the true high-fidelity optimum is found. However, this time, because of the initial sampling and the time to train the co-Kriging model, the convergence is slowed down, providing the same performances than the high-fidelity-only optimisation.

Lastly, an expected improvement criteria (see equation 4.6) is used instead of the traditional corrected function. The initial sampling permits the use of the co-Kriging from scratch, and the predicted error is used to predict where the improvements are most likely to be. Despite reducing greatly the number of high-fidelity calls, once again the burden imposed by the co-Kriging training is reflected on the convergence elapsed time.

The parallel coordinates plot presenting the optima points for the various multi-fidelity runs along with the initial sampling are shown in figure 9.6. The penalty constraint renders the function to optimise very complicated with high variations in some regions and lower sensitivity close to the optimum, as seen in the large values encounter during the sampling but the proximity of the optima points. Thus, it is a difficult problem for surrogate modelling such as for the co-Kriging used in the framework. This explains the long time required for the model training and the high number of high-fidelity calls to approximate the optimum. In terms of optimum, the final converged values are shown in table 9.6 along with the percentage of improvement between datum and the high-fidelity only optimum. Despite the large bounds

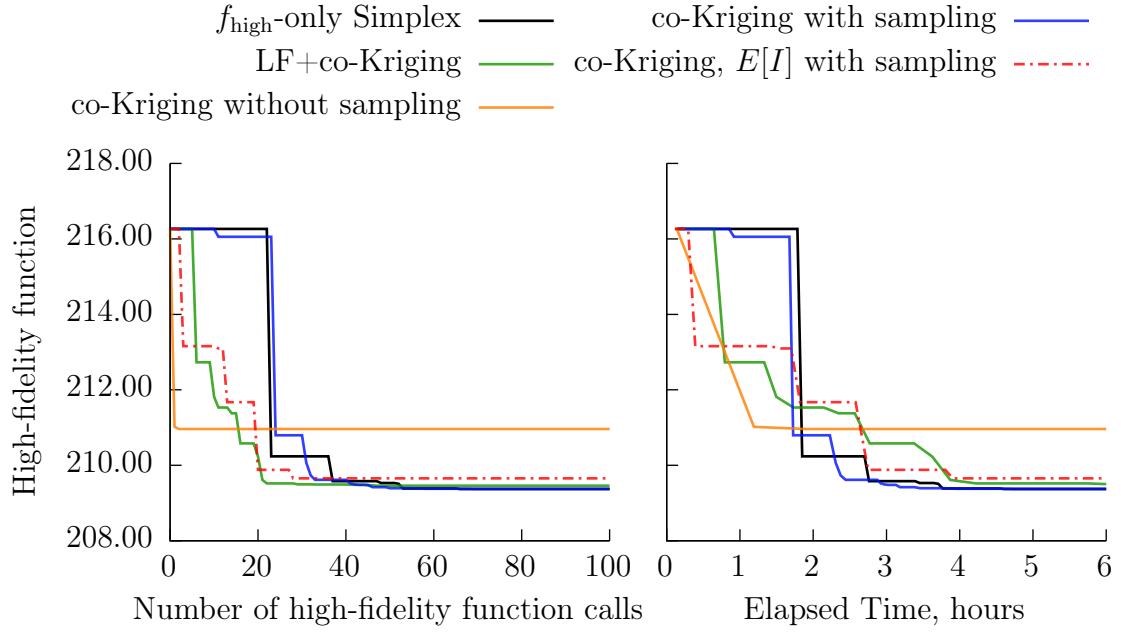


Figure 9.5: Convergence comparison between multifidelity (multiobjective formulation) and high-fidelity-only XRF1 optimisations with respect to the number of iterations on the left and with respect to the elapsed time on the right.

used, the multifidelity reaches indeed the same optimum values except for the one without initial sampling and therefore the one not using the co-Kriging correction.

To finish with the parallel coordinates plot, the final configurations can be analysed. And apart from the co-Kriging without sampling, that is behaving as in the previous section, the other methods lead to the same arrangement as the high-fidelity only optimum: a small negative twist close to the root section and a large positive twist for the tip. The resulting spanwise distribution is shown in figure 9.7: again the use of Kriging leads to a different spanwise lift. Both the multifidelity without sampling and the single-objective framework (MF+Kriging) reaches the same distribution, less optimal since it is wavy and deviates more from the ideal elliptical one. However, despite the variable differences, the results are close: a reduced lift at the root section and an increased one at the tip. To understand the effect of such lift distribution change on the optimum objective values, the table 9.6 shows the lift and drag variations in percentage compared to the datum configuration, along with the source of drag change. First, we focus on the constraint: the lift variation is small but all optimum configurations loose 0.7% of lift, which has a small effect on the ob-

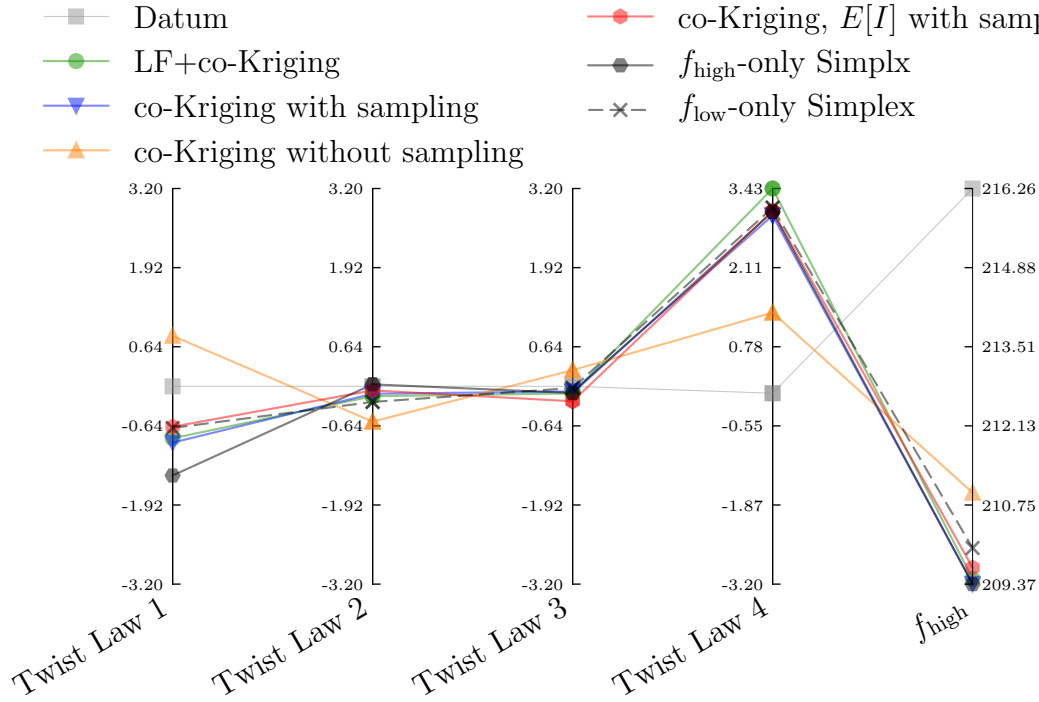


Figure 9.6: Parallel coordinates plot for single- and multifidelity optima from XRF1 optimisations.

Table 9.6: Optimum configurations lift and drag for XRF1 optimisation.

	f_{high}	C_L	C_{Dff}	C_{Df}	C_{Dvp}	C_{Di}	C_{Dw}
Datum	216.2	0.4986	216.07	37.140	32.020	111.52	35.390
f_{high} -only Simplex	209.7	0.4950	206.88				
LF + Kriging	211.5	0.4951	209.12	37.210	31.260	108.18	32.470
LF+co-Kriging	209.5	0.4950	206.93	37.100	30.910	106.56	32.360
co-Kriging w samp.	209.4	0.4948	206.63	37.110	30.810	106.52	32.190
co-Kriging w/o samp.	211.0	0.4961	209.66	37.170	31.360	108.19	32.940

jective value and does not explain the drop in drag. Indeed, the total drag, computed with a farfield approach, is reduced by more than 4%, representing 9 drag counts, except for the Kriging-based ones which have an improvement of 3%, representing 7 drag counts. This change in drag is mainly explained by a drop in induced and wave drag. The twist down change on the inboard sections and the twist up on the outboard sections changed the lift distribution which reduced the induced drag. The twist down movement induces a larger wave drag drop compared to the outboard increase (see figure C.10).

In addition to the high-fidelity only optimum, the low-fidelity one is also plotted in figure 9.6: it has been found by performing the optimisation solely on the low-fidelity function, and evaluating the optimum with the high-fidelity function for comparison. Despite the objective values being different, the low-fidelity model, providing enough iteration are performed, indicates the same design space region for the optimum configuration. However, the final values are different leading to a less optimal objective. The different configurations obtained with the use of the Kriging correction is explained by a change of the objective function as part of the process. It exacerbates the local optimum in this region and only the use of a co-Kriging model with the initial sampling permits to eliminate this phenomenon. Since the low-fidelity provides similar trends as the high-fidelity model, a higher initial trust region size would provide most of the improvements from the low-fidelity but will impose most of the computational cost from the low-fidelity function. The multifidelity framework presented in this thesis requires a quicker low-fidelity model to increase the ratio between the number of high- and low-fidelity function calls while ensuring that most of the computational cost lies in the high-fidelity function evaluation. Thus using the coarse mesh as the reference model and a quicker quasi-3D method as low-fidelity model may provide more time savings than keeping the fine mesh in the loop for a minimal optimality decrease.

9.3 Conclusion

This chapter applied the multifidelity optimisation to a three-dimensional geometry at transonic conditions. A different variation of the *fidelity* definition was used: a different discretisations of the same model was used for the high- and the low-fidelity models providing a time ratio of 40. For transonic design, the drag is of primary interest and is set as the objective to minimise. However the wing lift is usually

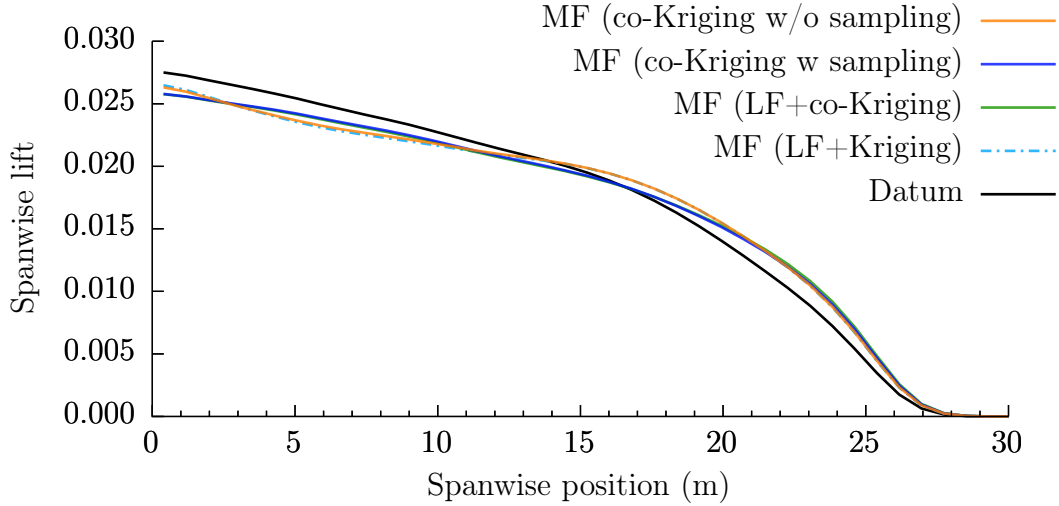


Figure 9.7: Spanwise lift distribution for the XRF1 optimum configurations.

fixed and set by the aircraft maximum weight. Hence a target lift is ensured which is possible in the multifidelity framework only with the use of a penalty function. The shape change is done by a mesh movement and parametrised by four twist variables.

The lack of exploration and the complex function shape prevented the initial single-objective multifidelity optimisation to escape the first local minimum induced by the correction. Performing the same optimisation directly with the high-fidelity function with a Simplex optimiser ended in a different optimum. Results are similar but less optimal when performing the optimisation solely on the low-fidelity model.

The co-Kriging model enabled the convergence to the same optimum as the high-fidelity-only albeit not providing any saving in terms of elapsed time due to a longer training time.

Because of the similarities between the low- and the high-fidelity functions along with the complex objective function shapes induced by the penalty constraint, the multifidelity optimisation presents no gain over the high-fidelity-only optimisation. In this case, an optimisation on the coarse mesh for the high-fidelity function and a rapid method such as MSES coupled with a lifting line would provide more time savings.

Conclusion & Future Work

10.1 Conclusion

The thesis aimed at developing and assessing a multifidelity multiobjective optimisation framework for maximum lift estimation. Hence the various objectives set to fulfil the aim were: 1. implement the multifidelity methodology, 2. apply it to a single-objective multielement aerofoil optimisation, 3. extend the framework for multiobjective optimisation on a similar case, 4. assess the effect of the low-fidelity accuracy on the multifidelity methodology, and 5. apply the framework to a three-dimensional case. The method is based on a zeroth order trust-region-based multifidelity framework introduced in the literature. Because the solution of the Reynolds-Averaged Navier-Stokes equations at high lift conditions are not accurate, the function gradients are not available; the framework therefore does not require such gradients.

Correction model The low-fidelity correction is initially ensured by a Radial Basis Function method that lacks independent control of the norm for each variable. However, the element movements and deflections do not have the same range and need to be controlled independently. The Kriging model was shown to be more accurate albeit more costly to train since it has more parameters to tune. Similarly, the additive correction (ensured by the Kriging model) was shown not to be sufficient for the multiobjective optimisation. A co-Kriging model, again more expensive to train, is shown to provide a more accurate correction because it computes the correlation between the low- and the high-fidelity data, improving in fine the multifidelity prediction.

Trust region definition Two trust region definitions were introduced: the Euclidean- and the step-based trust regions; no work existed in the literature that compared the two formulations. The former is found to be more expensive because it solves the subproblem more accurately and hence requires more low-fidelity function calls. The latter, however, is shown to be less accurate in terms of optimum found. This is explained by the subproblem resolution that is more approximatively solved.

Application to multielement aerofoil The suboptimisers used throughout the thesis has a stochastic component to improve the global exploration while limiting the increase in the number of function evaluations. Accordingly to the literature, the multifidelity optimisation decreases by more than 50% the number of high-fidelity function calls to obtain a converged optimum albeit, it also increases the variability in convergence because of the high number of subproblem resolutions that increases the randomness effect. No literature existed on the use of zeroth order trust region based algorithm on the complex aerodynamic problem of maximum lift prediction, but the first-order method has been shown to provide significant savings. This was not seen in the present work that uses the zeroth order formulation on two more complex multielement cases. A reduced accuracy in the optima and a higher variance between multiple runs were shown but a clear time saving on the second case with reduced boundaries was observed. An analysis of the effect of the initial trust region size was also done as it has not been found in existing literature. The Euclidean trust region was shown to be insensitive to the initial size whereas it affected the step-based trust region convergence: large number of steps allow the corrected function to leave areas where the corrections has an effect whereas the correction affects the whole Euclidean trust region (the whole design space).

Therefore an increase in problem complexity was translated as a decrease of the method performance. The reasons are multiple:

- the models accuracy differ due to different levels of flow physics captured which requires a more complex correction to capture the non-linearities;
- more local optima (different slat and flap positions lead to local good solutions) are present which requires a more thorough globalisation strategy;
- because the optimiser makes less progress, the sampled point distribution is less uniform which introduces numerical noise in the correction, limiting again the accuracy of the multifidelity method;
- the low-fidelity tools may be less robust than the high-fidelity ones because they are designed to work for specific conditions and may diverge when outside. The multifidelity solutions will always be limited by the low- and high-fidelity tools domains of validity.

An improved correction, using a co-Kriging multifidelity model as part of the trust-region optimisation (and not requiring low-fidelity evaluation as part of the

suboptimisation), led to a convergence improvement of 80% both in terms of number of function calls and in terms of elapsed time on the second single-objective test case.

Extension to multiobjective The main contribution to knowledge is the extension of the zeroth-order trust-region-based multifidelity method to multiobjective problems as no trust-region-based multiobjective optimisation was found in the literature. Applied on the same test case and despite a quicker Pareto front discovery, compared to the direct optimisation on the high-fidelity function, the final Pareto front was found to be dominated. The reason is that the additive correction is not sufficient to capture the differences in trends between the low- and the high-fidelity models at maximum lift conditions due to large non-linearities.

For the multiobjective formulation also, replacing the low-fidelity function and its correction by a single surrogate model such as a co-Kriging improves greatly the correction quality and hence the multifidelity Pareto front. The conclusion is a savings between 15 to 30% of CPU time compared to the optimisation on the high-fidelity model only. A second multifidelity approach from the literature was used as a reference method. It relies only on an adaptive surrogate model (Surrogate Based Optimiser —SBO) trained on the high-fidelity model. The trust-region-based multifidelity approach reached a better Pareto front (some points dominate the SBO approach) for a similar elapsed time. It is also believed that improvements can be brought to the correction model implementation and training.

Effect of the low-fidelity accuracy on the single-objective multifidelity optimiser

As pointed earlier, the low-fidelity model affects the multifidelity convergence speed and accuracy. An original contribution of this thesis is a quantitative analysis of the low-fidelity model accuracy on the implemented zeroth order trust region method on such complex aerodynamic problem. To do so, the low-fidelity model was artificially modified, first by changing the slat or flap variable effects, and second by adding various noise levels in the low-fidelity response. The more sensitive the variables group, the more detrimental the effect. For less sensitive variables, such as the slat, the optimisation behaves similarly to a SBO. Adding noise to the low-fidelity response however increased the tool accuracy since it provides more exploration until the noise amplitude reaches the model response one. As part of this work, a novel visualisation tool to understand the differences in sensitivities between the low- and the high-fidelity tools was introduced.

Three-dimensional geometry Finally, the method was applied to a single-objective multifidelity optimisation using a three-dimensional case at transonic conditions because of the complexity required to implement three-dimensional high-lift devices auto-meshing. The problem definition was not suitable to multifidelity surrogate-based optimisation due to the penalty constraint introducing discontinuities in the function to optimise. Despite this problem, the multifidelity optimisation showed that a similar convergence was possible, and improvements in the model implementation and robustness would certainly provide time savings.

Summary

- The method behaves as in the literature when applied to simple, in the sense of robust and accurate aerodynamic simulation tools, test case;
- More complex aerodynamic problems reduce the savings in single-objective because of more discrepancies in the model fidelities (physics is not correctly captured by the low-fidelity tools) and less robust tools (more inclined to diverge);
- The Euclidean trust-region is more accurate but more expensive than the step-based one because of the different levels of resolution of the subproblem;
- The method is extended to multiobjective problems but the additive correction is shown not to be enough;
- A co-Kriging model to fuse low- and high-fidelity function is shown to be enough to correct the low-fidelity tools for maximum lift estimation;
- With a co-Kriging model, the multifidelity multiobjective method performs better than the single-fidelity one: the Pareto front is more diversified and is not dominated;
- The low-fidelity model accuracy is shown to have effect on the multifidelity convergence: the most sensitive variables should be correctly captured by the low-fidelity one whereas the less sensitive variables can be accurately captured by the correction;
- Because a surrogate model is used for the correction, penalty-based constraints should be avoided.

10.2 Future work

Despite all the work achieved in this work, many improvements and validations should be performed to conclude the study and definitely conclude on the suitability of the trust-region based multifidelity approach to the application to complex aerodynamic optimisation. The various research axis are:

Method improvements: point selection For the multiobjective multifidelity trust-region-based optimisation, after each successful iteration, a new point is selected. Various point selections, inspired by the Particle Swarm Optimiser world, were introduced based on domination and crowdedness. However, some regions may not have enough points due to the Pareto front shape which may result in the point selection always selecting the same region. A memory can be implemented, as done in MOTS implementation, to prevent reselecting the same region too many times and forcing the exploration of other regions.

Method improvements: surrogate model The surrogate model quality is strongly coupled to the training point distribution. To improve the model quality, a more uniform sampling can be performed instead of a single point, in parallel. Parallel execution were tested in the single-objective formulation, using the first three best solutions. Using the best and few points around it may lead to an improvement in the model quality.

Adaptive trust region Initial work to adapt dynamically the trust region size was started without results. This lead should be explored more in details to allow a faster convergence when the model is known to be accurate, and slower steps when the region is known to be inaccurate. Variance information, provided by the Gaussian Process-based methods (Kriging, co-Kriging), should be used to calculate uncertainties in the trust region.

Suboptimiser High variations in the multifidelity optimisation were seen depending on the optimiser used; Such optimisers included Tabu Search, Particle Swarms, genetic algorithms, and Simplex-based methods. Most of those optimisers require parameters to be set according to the problem solved. A parametric study on the effect of each parameter (population size, cognitive-social behaviour, step sizes, ...) should be performed in order to highlight the best optimisation technique for the subproblem optimisation.

Gradients Gradient-based methods were discarded from the beginning because of the function to optimise that could not provide them accurately. The last test-case however can be used with such algorithm to enhanced the correction models for example. A gradient-based method can also be used to solve the subproblem for the multielement aerofoil case when the co-Kriging model is used. Since no noisy simulations are used in the suboptimisation, the analytical gradients from the co-Kriging can directly be used to efficiently optimise the corrected function. Care needs to be taken to ensure a noise-free co-Kriging.

Constraints The last case showed that penalty constraints should not be used within a multifidelity optimisation (including SBO and physics-based multifidelity) as it cannot be modelled by the surrogate. Hence, constraints should be implemented in the trust-region formulations adapting the Trust Region Sequential Quadratic Programming (TRSQP) for example (Nocedal and Wright 1999).

Maximum lift estimation from clean configuration The work final goal is the prediction of an aerofoil maximum lift, but wings and aerofoils are designed for cruise conditions. Therefore, to be complete, the maximum lift capabilities should be estimated from the clean configuration in two steps: 1. creation of the high-lift devices from the clean configuration and 2. maximum lift estimation by high-lift devices position optimisation. Work has already been initiated as part of the thesis and is presented in appendix D.

Bibliography

- Abbott, Ira H, Von Doenhoff, Albert E, and Stivers Jr, Louis (1945). *Summary of Airfoil Data*. Tech. rep. 824. National Advisory Committee for Aeronautics.
- Alexandrov, Natalia M., Dennis, J. E., et al. (1998). “A Trust-Region Framework for Managing the Use of Approximation Models in Optimization”. In: *Structural optimization* 15.1, pp. 16–23. DOI: [10.1007/BF01197433](https://doi.org/10.1007/BF01197433).
- Alexandrov, Natalia M. and Lewis, Robert Michael (2001). “An Overview of First-Order Model Management for Engineering Optimization”. In: *Optimization and Engineering* 2.4, pp. 413–430. ISSN: 1573-2924. DOI: [10.1023/A:1016042505922](https://doi.org/10.1023/A:1016042505922). URL: <http://dx.doi.org/10.1023/A:1016042505922>.
- Alexandrov, Natalia M., Lewis, Robert Michael, et al. (2001). “Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models”. In: *Journal of Aircraft* 38.6, pp. 1093–1101. DOI: [10.2514/2.2877](https://doi.org/10.2514/2.2877).
- Alexandrov, Natalia M., Nielsen, E. J., et al. (2000). “First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization”. In: *8th Symposium on Multidisciplinary Analysis and Optimization*. paper 2000-4886. Long Beach, USA: AIAA. DOI: [10.2514/6.2000-4886](https://doi.org/10.2514/6.2000-4886).
- Alonso, D., Vega, J. M., and Velazquez, A. (2010). “Reduced-Order Model for Viscous Aerodynamic Flow Past an Airfoil”. In: *AIAA Journal* 48.9, pp. 1946–1958. ISSN: 0001-1452. DOI: [10.2514/1.j050153](https://doi.org/10.2514/1.j050153). URL: <https://doi.org/10.2514/1.J050153>.
- Alvarez-Benitez, Julio E., Everson, Richard M., and Fieldsend, Jonathan E. (2005). “A Mopso Algorithm Based Exclusively on Pareto Dominance Concepts”. In: *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings*. Ed. by Coello Coello, Carlos A., Hernández Aguirre, Arturo, and Zitzler, Eckart. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 459–473. ISBN: 978-3-540-31880-4. DOI: [10.1007/978-3-540-31880-4_32](https://doi.org/10.1007/978-3-540-31880-4_32).
- Amrit, A., Leifsson, L., and Koziel, S. (2017). “Design strategies for multi-objective optimization of aerodynamic surfaces”. English. In: *Engineering Computations (Swansea, Wales)* 34.5. cited By 0, pp. 1724–1753. DOI: [10.1108/EC-07-2016-0239](https://doi.org/10.1108/EC-07-2016-0239). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0>.

- 0 - 85025835862&doi=10.1108%2fEC-07-2016-0239&partnerID=40&md5=bc613a59f07f62a5f81245a1a42e7b13.
- Amrit, Anand et al. (2016). “Efficient Multi-Objective Aerodynamic Optimization by Design Space Dimension Reduction and Co-Kriging”. In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Washington, USA, p. 3515. DOI: [10.2514/6.2016-3515](https://doi.org/10.2514/6.2016-3515).
- Anderson Jr, John David (2010). *Fundamentals of aerodynamics*. Tata McGraw-Hill Education.
- ANSYS, Fluent 14.5 (2013a). *ANSYS Fluent Theory Guide*. Ed. by ANSYS, Inc. Lebanon.
- (2013b). *ANSYS Fluent User’s Guide*. Ed. by ANSYS, Inc. Lebanon.
- ANSYS, ICEM CFD 14.5 (2013). *ANSYS ICEM User’s Guide*. Ed. by ANSYS, Inc. Lebanon.
- Argüelles, Pedro and al. (2001). “European Aeronautics: A Vision for 2020, Meeting Society’s Needs and Winning Global Leadership”. In:
- Bandler, John W et al. (1994). “Space Mapping Technique for Electromagnetic Optimization”. In: *IEEE Transactions on Microwave Theory and Techniques* 42.12, pp. 2536–2544. DOI: [10.1109/22.339794](https://doi.org/10.1109/22.339794).
- Bardina, J. et al. (1997). “Turbulence modeling validation”. In: *Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics. DOI: [10.2514/6.1997-2121](https://doi.org/10.2514/6.1997-2121). URL: <https://doi.org/10.2514/6.1997-2121>.
- Baudin, Michael et al. (2017). *pyDoE*. <https://pythonhosted.org/pyDOE/>. Accessed 18/11/2017.
- Belyaev, M.G. and Burnaev, E.V. (2013). “Approximation of a multidimensional dependency based on a linear expansion in a dictionary of parametric functions”. In: *Informatics and its Applications* 7.3, pp. 114–125.
- Benini, E., Ponza, R., and Massaro, A. (2011). “High-lift multi-element airfoil shape and setting optimization using multi-objective evolutionary algorithms”. English. In: *Journal of Aircraft* 48.2. cited By 9, pp. 683–696. ISSN: 00218669. DOI: [10.2514/1.C031233](https://www.scopus.com/inward/record.uri?eid=2-s2.0-79953655423&doi=10.2514%2f1.C031233&partnerID=40&md5=c8c9342af4d4c007294ff41e9d3d8209). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79953655423&doi=10.2514%2f1.C031233&partnerID=40&md5=c8c9342af4d4c007294ff41e9d3d8209>.
- Berkooz, Gal, Holmes, Philip, and Lumley, John L (1993). “The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows”. In: *Annual Review of Fluid Mechanics* 25.1, pp. 539–575. DOI: [10.1146/annurev.fl.25.010193.002543](https://doi.org/10.1146/annurev.fl.25.010193.002543).

- eprint: <https://doi.org/10.1146/annurev.fl.25.010193.002543>. URL: <https://doi.org/10.1146/annurev.fl.25.010193.002543>.
- Bianchi, Leonora et al. (2009). “A Survey on Metaheuristics for Stochastic Combinatorial Optimization”. English. In: *Natural Computing* 8.2, pp. 239–287. ISSN: 1567-7818. DOI: [10.1007/s11047-008-9098-4](https://doi.org/10.1007/s11047-008-9098-4).
- Björkman, Mattias and Holmström, Kenneth (2000). “Global Optimization of Costly Nonconvex Functions Using Radial Basis Functions”. English. In: *Optimization and Engineering* 1.4, pp. 373–397. ISSN: 1389-4420. DOI: [10.1023/A:1011584207202](https://doi.org/10.1023/A:1011584207202).
- Brezillon, Jöel, Dwight, RP, and Wild, Jochen (2008). “Numerical Aerodynamic Optimisation of 3d High-Lift Configurations”. In: *26th ICAS Congress*. Anchorage, USA.
- Brooks, Thomas F., Pope, D. Stuart, and Marcolini, Michael A. (1989). *Airfoil Self-Noise and prediction*. Tech. rep. Reference Publication 1218. NASA. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19890016302.pdf>.
- Brune, G W and McMasters, J H (1990). “Computational Aerodynamics Applied to High-Lift Systems”. In: *Applied Computational Aerodynamics*. Ed. by Henne, P. Progress in Astronautics and Aeronautics. American Institute of Aeronautics and Astronautics, pp. 389–433. DOI: [10.2514/5.9781600865985.0389.0433](https://doi.org/10.2514/5.9781600865985.0389.0433). URL: <https://doi.org/10.2514/5.9781600865985.0389.0433>.
- Burnaev, E. V. and Zaytsev, A. A. (2015). “Surrogate modeling of multifidelity data for large samples”. In: *Journal of Communications Technology and Electronics* 60.12, pp. 1348–1355. ISSN: 1555-6557. DOI: [10.1134/S1064226915120037](https://doi.org/10.1134/S1064226915120037). URL: <http://dx.doi.org/10.1134/S1064226915120037>.
- Cambier, L. and Gizaix, M. (2002). “elsA - An efficient object-oriented solution to CFD complexity”. In: *40th AIAA Aerospace Sciences Meeting & Exhibit, Aerospace Sciences Meetings*. AIAA paper 2002-108. Reno, USA: AIAA. DOI: [10.2514/6.2002-108](https://doi.org/10.2514/6.2002-108). URL: <https://doi.org/10.2514/6.2002-108>.
- Cao, Yongtao, Smucker, Byran J., and Robinson, Timothy J. (2015). “On Using the Hypervolume Indicator to Compare Pareto Fronts: Applications to Multi-Criteria Optimal Experimental Design”. In: *Journal of Statistical Planning and Inference* 160, pp. 60–74. ISSN: 0378-3758. DOI: [10.1016/j.jspi.2014.12.004](https://doi.org/10.1016/j.jspi.2014.12.004). URL: <http://www.sciencedirect.com/science/article/pii/S0378375814002006>.
- Carter, James E. (1979). “A new boundary-layer inviscid iteration technique for separated flow”. In: *4th Computational Fluid Dynamics Conference and Co-located*

- Conferences*, AIAA Paper 79-1450. Williamsburg, USA. DOI: [10.2514/6.1979-1450](https://doi.org/10.2514/6.1979-1450). URL: <https://doi.org/10.2514/6.1979-1450>.
- Cebeci, Tuncer, Besnard, Eric, and Chen, Hsun H (1998). “An Interactive Boundary-Layer Method for Multielement Airfoils”. In: *Computers & fluids* 27.5, pp. 651–661.
- Cebeci, T et al. (1986). *Viscid/Inviscid Separated Flows*. Tech. rep. AFWAL-TR-86-3048. McDonnell Aircraft Company.
- Chien, Kuei-Yuan (1982). “Predictions of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model”. In: *AIAA Journal* 20.1, pp. 33–38. ISSN: 0001-1452. DOI: [10.2514/3.51043](https://arc.aiaa.org/doi/abs/10.2514/3.51043). URL: <https://arc.aiaa.org/doi/abs/10.2514/3.51043>.
- Choi, Seongim, Alonso, Juan J, and Kroo, Ilan M (2009). “Two-Level Multifidelity Design Optimization Studies for Supersonic Jets”. In: *Journal of Aircraft* 46.3, pp. 776–790. DOI: [10.2514/1.34362](https://doi.org/10.2514/1.34362).
- Choi, Seongim, Alonso, Juan J, Kroo, Ilan M, et al. (2008). “Multifidelity Design Optimization of Low-Boom Supersonic Jets”. In: *Journal of Aircraft* 45.1, pp. 106–118. DOI: [10.2514/1.28948](https://doi.org/10.2514/1.28948).
- Conn, Andrew R., Scheinberg, Katya, and Vicente, Luís N. (2009). “Global Convergence of General Derivative-Free Trust-Region Algorithms to First and Second-Order Critical Points”. In: *SIAM Journal on Optimization* 20.1, pp. 387–415. DOI: [10.1137/060673424](https://doi.org/10.1137/060673424).
- Couckuyt, Ivo, Deschrijver, Dirk, and Dhaene, Tom (2014). “Fast calculation of multi-objective probability of improvement and expected improvement criteria for Pareto optimization”. In: *Journal of Global Optimization* 60.3, pp. 575–594. ISSN: 1573-2916. DOI: [10.1007/s10898-013-0118-2](https://doi.org/10.1007/s10898-013-0118-2). URL: <https://doi.org/10.1007/s10898-013-0118-2>.
- Courrier, N., Boucard, P.-A., and Soulier, B. (2014). “The Use of Partially Converged Simulations in Building Surrogate Models”. In: *Advances in Engineering Software* 67, pp. 186–197. ISSN: 0965-9978. DOI: [10.1016/j.advengsoft.2013.09.008](http://www.sciencedirect.com/science/article/pii/S0965997813001622). URL: <http://www.sciencedirect.com/science/article/pii/S0965997813001622>.
- Cressie, Noel (1992). “Statistics for Spatial Data”. In: *Terra Nova* 4.5, pp. 613–617.
- Dalle, Derek J. and Fidkowski, Krzysztof (2014). “Multifidelity Airfoil Shape Optimization Using Adaptive Meshing”. In: *32nd AIAA Applied Aerodynamics Conference*. paper 2014-2996. Atlanta, USA: AIAA. DOI: [10.2514/6.2014-2996](https://doi.org/10.2514/6.2014-2996). URL:

- <http://www.scopus.com/inward/record.url?eid=2-s2.0-84903881421&partnerID=40&md5=5f3e2ca87d695f91fc5c2dc1c32ae736>.
- Dam, C. P. van (2002). “The Aerodynamic Design of Multi-Element High-Lift Systems for Transport Airplanes”. In: *Progress in Aerospace Sciences* 38.2, pp. 101–144. DOI: [10.1016/S0376-0421\(02\)00002-7](https://doi.org/10.1016/S0376-0421(02)00002-7).
- DATADVANCE (2016). *Macros 2.4 User Manual*. <https://www.datadvance.net/product/macros/documentation.html>.
- Deb, Kalyanmoy et al. (2002). “A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-Ii”. In: *Evolutionary Computation, IEEE Transactions on* 6.2, pp. 182–197.
- Demange, Jean (2015). “Multifidelity Optimisation For High-Lift Aerofoils”. Airbus DiPaRT.
- (2016). “Multifidelity Optimisation For High-Lift Aerofoils: Effect of Low-Fidelity Model”. Airbus DiPaRT.
 - (2017). “Multifidelity Multiobjective Optimisation for High-Lift Aerofoils”. Airbus PhD day.
- Demange, Jean, Savill, A. Mark, and Kipouros, Timoleon (2016a). “A Multifidelity Multiobjective Optimization Framework for High-Lift Airfoils”. In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. paper 2016-3367. Washington, USA: AIAA, p. 3367. DOI: [10.2514/6.2016-3367](https://doi.org/10.2514/6.2016-3367).
- (2016b). “Multifidelity Optimization for High-Lift Airfoils”. In: *54th AIAA Aerospace Sciences Meeting*. paper 2016-0557. San Diego, USA: AIAA, p. 0557. DOI: [10.2514/6.2016-0557](https://doi.org/10.2514/6.2016-0557).
 - (2017a). “A Trust-Region-Based Co-Kriging Multifidelity Multiobjective Particle Swarm Optimizer for High-Lift Airfoil Design”. In preparation.
 - (2017b). “A Trust-Region-Based Multifidelity Multiobjective Optimization Framework for High-Lift Airfoil Design”. In preparation.
- Drela, M. (1986). “Two-dimensional transonic aerodynamic design and analysis using the Euler equations”. PhD thesis. Massachusetts Institute of Technology.
- Drela, Mark (1989). “Xfoil: An Analysis and Design System for Low Reynolds Number Airfoils”. In: *Low Reynolds number aerodynamics*. Springer, pp. 1–12.
- (1990). “Newton Solution of Coupled Viscous/inviscid Multielement Airfoil Flows”. In: *21st Fluid Dynamics, Plasma Dynamics and Lasers Conference*. paper 1990-1470. Seattle, USA: AIAA, p. 1990. DOI: [10.2514/6.1990-1470](https://doi.org/10.2514/6.1990-1470).

- Dwight, Richard P. and Brezillon, J  el (2006). “Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization”. In: *AIAA Journal* 44.12, pp. 3022–3031. ISSN: 0001-1452. DOI: [10.2514/1.21744](https://doi.org/10.2514/1.21744).
- Eberhart, Russell and Kennedy, James (1995). “A New Optimizer Using Particle Swarm Theory”. In: *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan*. IEEE. IEEE, pp. 39–43. DOI: [10.1109/MHS.1995.494215](https://doi.org/10.1109/MHS.1995.494215).
- Edwards, Jack R and Chandra, Suresh (1996). “Comparison of Eddy Viscosity-Transport Turbulence Models for Three-Dimensional, Shock-Separated Flow-fields”. In: *AIAA journal* 34.4, pp. 756–763.
- Emmerich, M. T. M., Giannakoglou, K. C., and Naujoks, B. (2006). “Single- and multiobjective evolutionary optimization assisted by Gaussian random field meta-models”. In: *IEEE Transactions on Evolutionary Computation* 10.4, pp. 421–439. ISSN: 1089-778X. DOI: [10.1109/TEVC.2005.859463](https://doi.org/10.1109/TEVC.2005.859463).
- Engineering Sciences Data Unit (1994). *Increments in Aerofoil Lift Coefficient at Zero Angle of Attack and in Maximum Lift Coefficient Due to Deployment of Various Leading-Edge High-Lift Devices at Low Speeds*. Tech. rep. 94027. ESDU.
- (1995a). *Increments in Aerofoil Lift Coefficient at Zero Angle of Attack and in Maximum Lift Coefficient Due to Deployment of a Single-Slotted Trailing-Edge Flap, with or without a Leading-Edge High-Lift Device, at Low Speeds*. Tech. rep. 94030. ESDU.
- (1995b). *Introduction to the Estimation of the Lift Coefficients at Zero Angle of Attack and at Maximum Lift for Aerofoils with High-Lift Devices at Low Speeds*. Tech. rep. 94026. ESDU.
- (2006). *Low-Speed Drag Coefficient Increment at Constant Lift Due to Full-Span Plain Flaps or Controls*. Tech. rep. 87024. ESDU.
- (2008). *Increment in Aerofoil Profile Drag Coefficient Due to the Deployment of a Single-Slotted Flap*. Tech. rep. 87005. ESDU.
- Epureanu, B.I. (2003). “A parametric analysis of reduced order models of viscous flows in turbomachinery”. In: *Journal of Fluids and Structures* 17.7, pp. 971–982. ISSN: 0889-9746. DOI: [http://dx.doi.org/10.1016/S0889-9746\(03\)00044-6](https://doi.org/10.1016/S0889-9746(03)00044-6). URL: <http://www.sciencedirect.com/science/article/pii/S0889974603000446>.

- Erickson, Larry L. (1990). *Panel Methods: An Introduction*. Tech. rep. Technical Paper 2995. NASA. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19910009745.pdf>.
- Fang, Hongbing and Horstemeyer, Mark F. (2006). “Global Response Approximation with Radial Basis Functions”. In: *Engineering Optimization* 38.4, pp. 407–424. DOI: [10.1080/03052150500422294](https://doi.org/10.1080/03052150500422294).
- Federal Aviation Regulations (2002). “Part 25: Airworthiness Standards: Transport Category Airplanes”. In: *Federal Aviation Administration* 7. Washington, USA.
- Fefferman, Charles L. (2017). *Existence and Smoothness of the Navier–Stokes Equation*. <http://www.claymath.org/sites/default/files/navierstokes.pdf>. Accessed 12/11/2017.
- Fernández-Godino, M. Giselle et al. (2016). “Review of multi-fidelity models”. In: *ARXIV*. URL: <https://arxiv.org/abs/1609.07196>.
- Flaig, A. and Hilbilg, R. (1991). “High-Lift Design for Large Civil Aircraft”. In: *High Lift System Aerodynamics*. AGARD CP-515, pp. 31-1–31-12.
- Forrester, Alexander I. J. and Keane, Andy J. (2009). “Recent Advances in Surrogate-Based Optimization”. In: *Progress in Aerospace Sciences* 45.1-3, pp. 50–79. URL: <http://eprints.soton.ac.uk/65935/>.
- Forrester, Alexander I. J., Sóbester, András, and Keane, Andy J. (2007). “Multi-Fidelity Optimization Via Surrogate Modelling”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2088, pp. 3251–3269. ISSN: 1364-5021. DOI: [10.1098/rspa.2007.1900](https://doi.org/10.1098/rspa.2007.1900). eprint: <http://rspa.royalsocietypublishing.org/content/463/2088/3251.full.pdf>. URL: <http://rspa.royalsocietypublishing.org/content/463/2088/3251>.
- (2008). *Engineering Design Via Surrogate Modelling: A Practical Guide*. Wiley.
- Fortin, Félix-Antoine et al. (2012). “DEAP: Evolutionary Algorithms Made Easy”. In: *Journal of Machine Learning Research* 13, pp. 2171–2175.
- Foster, D. N. (1972). “Flow around Wing Sections with High-Lift Devices”. In: *Journal of Aircraft* 9.3, pp. 205–210.
- Ghazlane, Imane (2012). “Adjoint-based aerostructural sensitivity analysis for wing design”. Theses. Université Nice Sophia Antipolis. URL: <https://tel.archives-ouvertes.fr/tel-00925210>.
- Giannakoglou, K.C. (2002). “Design of Optimal Aerodynamic Shapes Using Stochastic Optimization Methods and Computational Intelligence”. In: *Progress in*

- Aerospace Sciences* 1.38, pp. 43–76. DOI: [10.1016/S0376-0421\(01\)00019-7](https://doi.org/10.1016/S0376-0421(01)00019-7). URL: [http://doi.org/10.1016/S0376-0421\(01\)00019-7](http://doi.org/10.1016/S0376-0421(01)00019-7).
- Gill, Philip E. and Murray, Walter (1972). “Quasi-Newton Methods for Unconstrained Optimization”. In: *IMA Journal of Applied Mathematics* 9.1, pp. 91–108.
- Giunta, Anthony A. et al. (1994). “Noisy Aerodynamic Response and Smooth Approximations in HSCT Design”. In: *5th Symposium on Multidisciplinary Analysis and Optimization*. Panama City Beach, USA: AIAA. DOI: [10.2514/6.1994-4376](https://doi.org/10.2514/6.1994-4376).
- Glover, Fred and Laguna, Manuel (1997). *Tabu Search*. Tech. rep. Kluwer Academic Publishers, Boston, MA.
- Green, J.E (2005). *Innovative Configurations and Advanced Concepts for Future Civil Aircraft*. Ed. by Fluid Dynamics, Von Karman Institute for. Lecture Series 2005-06, Von Karman Institute for Fluid Dynamics.
- Greenman, Roxana M and Roth, Karlin R (1999). “High-Lift Optimization Design Using Neural Networks on a Multi-Element Airfoil”. In: *Journal of fluids engineering* 121.2, pp. 434–440. DOI: [10.1115/1.2822228](https://doi.org/10.1115/1.2822228).
- Gutmann, H.-M. (2001). “A Radial Basis Function Method for Global Optimization”. English. In: *Journal of Global Optimization* 19.3, pp. 201–227. ISSN: 0925-5001. DOI: [10.1023/A:1011255519438](https://doi.org/10.1023/A:1011255519438).
- Han, Zhong-Hua, Görtz, Stefan, and Zimmermann, Ralf (2013). “Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function”. In: *Aerospace Science and Technology* 25.1, pp. 177–189. ISSN: 1270-9638. DOI: [http://dx.doi.org/10.1016/j.ast.2012.01.006](https://dx.doi.org/10.1016/j.ast.2012.01.006). URL: <http://www.sciencedirect.com/science/article/pii/S127096381200017X>.
- Hancock, B. J. and Mattson, C. A. (2013). “The Smart Normal Constraint Method for Directly Generating a Smart Pareto Set”. In: *Structural and Multidisciplinary Optimization* 48.4, pp. 763–775. ISSN: 1615-1488. DOI: [10.1007/s00158-013-0925-6](https://doi.org/10.1007/s00158-013-0925-6).
- Haykin, S.S. (1999). *Neural Networks: A Comprehensive Foundation*. Second. International edition. Prentice Hall. ISBN: 9780132733502. URL: <http://books.google.co.uk/books?id=bX4pAQAAMAAJ>.
- Hooke, Robert and Jeeves, T A (1961). “‘Direct Search’ Solution of Numerical and Statistical Problems”. In: *Journal of the ACM* 8.2, pp. 212–229. DOI: [10.1145/321062.321069](https://doi.org/10.1145/321062.321069).
- Huang, Likeng, Gao, Zhenghong, and Zhang, Dehu (2013). “Research on multi-fidelity aerodynamic optimization methods”. In: *Chinese Journal of Aeronautics*

- 26.2, pp. 279–286. ISSN: 1000-9361. DOI: <http://dx.doi.org/10.1016/j.cja.2013.02.004>. URL: <http://www.sciencedirect.com/science/article/pii/S1000936113000162>.
- ICAO (2017). *Airline Operating Costs and Productivity*. Accessed 25/09/2016. Tehran, Iran. URL: <https://www.icao.int/MID/Documents/2017/Aviation%20Data%20and%20Analysis%20Seminar/PPT3%20-%20Airlines%20operating%20costs%20and%20productivity.pdf>.
- Inselberg, Alfred and Dimsdale, Bernard (1991). “Parallel Coordinates”. In: *Human-Machine Interactive Systems*. Springer, pp. 199–233.
- Iuliano, E. (2017). “Global optimization of benchmark aerodynamic cases using physics-based surrogate models”. In: *Aerospace Science and Technology* 67. cited By 0, pp. 273–286. DOI: [10.1016/j.ast.2017.04.013](https://doi.org/10.1016/j.ast.2017.04.013). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85018636566&doi=10.1016/j.ast.2017.04.013&partnerID=40&md5=e2562cc0e01958d3f7e6c975e19f48c4>.
- Jaeggi, Daniel M., Parks, G T, et al. (2008). “The Development of a Multi-Objective Tabu Search Algorithm for Continuous Optimisation Problems”. In: *European Journal of Operational Research* 185.3, pp. 1192–1212. DOI: [10.1016/j.ejor.2006.06.048](https://doi.org/10.1016/j.ejor.2006.06.048).
- Jaeggi, Daniel M., Parks, Geoffrey T., et al. (2008). “Robust Multi-Fidelity Aerodynamic Design Optimization Using Surrogate Models”. In: *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. paper 2008-6052. Victoria, Canada.
- Jameson, A. (1989). “Computational Aerodynamics for Aircraft Design”. In: *Science* 245.4916, pp. 361–371. ISSN: 0036-8075. DOI: [10.1126/science.245.4916.361](https://doi.org/10.1126/science.245.4916.361). eprint: <http://science.sciencemag.org/content/245/4916/361.full.pdf>. URL: <http://science.sciencemag.org/content/245/4916/361>.
- Jansen, Peter W. and Perez, Ruben E. (2011). “Constrained Structural Design Optimization Via a Parallel Augmented Lagrangian Particle Swarm Optimization Approach”. In: *Computers & Structures* 89.13-14, pp. 1352–1366. ISSN: 0045-7949. DOI: [10.1016/j.compstruc.2011.03.011](https://doi.org/10.1016/j.compstruc.2011.03.011). URL: <http://www.sciencedirect.com/science/article/pii/S0045794911000824>.
- Jarrett, Jerome P and Ghisu, Tiziano (2012). “An Approach to Multi-Fidelity Optimization of Aeroengine Compression Systems”. In: *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidis-*

- ciplinary Analysis and Optimization Conference*. paper 2012-5634. Indianapolis, USA: AIAA. DOI: [10.2514/6.2012-5634](https://doi.org/10.2514/6.2012-5634).
- Jin, R., Chen, W., and Simpson, Timothy W. (2001). “Comparative Studies of Meta-modelling Techniques under Multiple Modelling Criteria”. English. In: *Structural and Multidisciplinary Optimization* 23.1, pp. 1–13. ISSN: 1615-147X. DOI: [10.1007/s00158-001-0160-4](https://doi.org/10.1007/s00158-001-0160-4).
- Jin, Ruichen, Chen, Wei, and Sudjianto, Agus (2002). “On Sequential Sampling for Global Metamodeling in Engineering Design”. In: *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. Montreal, Canada, pp. 539–548.
- Johnson, Steven G. (2016). *The Nlopt Nonlinear-Optimization Package*. <http://ab-initio.mit.edu/nlopt>. URL: <http://ab-initio.mit.edu/nlopt>.
- Jones, Donald R. (2001). “A Taxonomy of Global Optimization Methods Based on Response Surfaces”. In: *Journal of Global Optimization* 21, pp. 345–383. DOI: [10.1023/A:1012771025575](https://doi.org/10.1023/A:1012771025575).
- Jones, Donald R., Schonlau, Matthias, and Welch, William J. (1998). “Efficient Global Optimization of Expensive Back-Box Functions”. In: *Journal of Global Optimization* 13.4, pp. 455–492.
- Jonsson, Ingi M. et al. (2015). “Shape Optimization of Trawl-doors Using Variable-fidelity Models and Space Mapping”. In: *Procedia Computer Science* 51. International Conference On Computational Science, ICCS 2015, pp. 905–913. ISSN: 1877-0509. DOI: [http://dx.doi.org/10.1016/j.procs.2015.05.223](https://dx.doi.org/10.1016/j.procs.2015.05.223). URL: <http://www.sciencedirect.com/science/article/pii/S1877050915010315>.
- Kanazaki, Masahiro et al. (2006). “Multi-Objective Aerodynamic Optimization of Elements’ Setting for High-lift Airfoil Using Kriging Model”. In: *44th AIAA Aerospace Sciences Meeting and Exhibit*. paper 2006-1471. Reno, USA: AIAA. DOI: [10.2514/6.2006-1471](https://doi.org/10.2514/6.2006-1471). URL: <https://doi.org/10.2514/6.2006-1471>.
- Keane, A. J. (2003). “Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods”. In: *Journal of Aircraft* 40.4, pp. 741–750. ISSN: 0021-8669. DOI: [10.2514/2.3153](https://doi.org/10.2514/2.3153). URL: <https://doi.org/10.2514/2.3153>.
- Kennedy, Marc C and O’Hagan, Anthony (2001). “Bayesian Calibration of Computer Models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.3, pp. 425–464.

- Kim, Sangho, Alonso, Juan J, and Jameson, Antony (2004). “Multi-Element High-Lift Configuration Design Optimization Using Viscous Continuous Adjoint Method”. In: *Journal of Aircraft* 41.5, pp. 1082–1097.
- Kontogiannis, Spyridon G. et al. (2018). “A Comparison Study of Two Multifidelity Methods for Aerodynamic Optimization”. In: *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. paper 2018-0415. Florida, USA: AIAA, p. 0415. DOI: [10.2514/6.2018-0415](https://doi.org/10.2514/6.2018-0415).
- Koziel, S. et al. (2017). “Pareto ranking bisection algorithm for rapid multi-objective design of antenna structures”. English. In: cited By 0. Institute of Electrical and Electronics Engineers Inc., pp. 131–133. ISBN: 9781509048373. DOI: [10.1109/NEMO.2017.7964210](https://doi.org/10.1109/NEMO.2017.7964210). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85028500834&doi=10.1109%2fNEMO.2017.7964210&partnerID=40&md5=444d111d264e52dc2111999cbc4ebc8a>.
- Lam, Nina Siu-Ngan (1983). “Spatial Interpolation Methods: A Review”. In: *The American Cartographer* 10.2, pp. 129–150.
- Le Balleur, JC and Neron, M (1991). “A Viscous-Inviscid Solver for High-Lift Incompressible Flows Over Multi-Element Airfoils at Deep Separation Conditions”. In: *High Lift System Aerodynamics*. AGARD CP-515. AGARD, pp. 11-1–11-12.
- Leifsson, L., Koziel, S., and Tesfahunegn, Y.A. (2016). “Multiobjective aerodynamic optimization by variable-fidelity models and response surface surrogates”. English. In: *AIAA Journal* 54.2. cited By 4, pp. 531–541. DOI: [10.2514/1.J054128](https://doi.org/10.2514/1.J054128). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84962376899&doi=10.2514%2f1.J054128&partnerID=40&md5=900cac543af912a6e19b7f3d7de44fac>.
- Leifsson, Leifur and Koziel, Slawomir (2015). *Simulation-driven Aerodynamic Design Using Variable-fidelity Models*. World Scientific Publishing Company. 444 pp. URL: http://www.ebook.de/de/product/23915137/leifur_leifsson_slawomir_koziel_simulation_driven_aerodynamic_design_using_variable_fidelity_models.html.
- Lewis, Robert Michael (1996). “A Trust Region Framework for Managing Approximation Models in Engineering Optimization”. In: *Proceedings of the Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Design*, pp. 1053–1055.
- Lian, Yongsheng, Oyama, Akira, and Liou, Meng-Sing (2010). “Progress in Design Optimization Using Evolutionary Algorithms for Aerodynamic Problems”. In:

- Progress in Aerospace Sciences* 46.5–6, pp. 199–223. DOI: [10.1016/j.paerosci.2009.08.003](https://doi.org/10.1016/j.paerosci.2009.08.003).
- Liang, Y.c. et al. (2002). “Proper Orthogonal Decomposition and Its Applications—part I: Theory”. In: *Journal of Sound and Vibration* 252.3, pp. 527–544. ISSN: 0022-460X. DOI: [10.1006/jsvi.2001.4041](https://doi.org/10.1006/jsvi.2001.4041). URL: <http://www.sciencedirect.com/science/article/pii/S0022460X01940416>.
- Madsen, Jens I. and Langthjem, Mikael (2001). “Multifidelity Response Surface Approximations for the Optimum Design of Diffuser Flows”. In: *Optimization and Engineering* 2.4, pp. 453–468. ISSN: 1573-2924. DOI: [10.1023/A:1016046606831](https://doi.org/10.1023/A:1016046606831). URL: <http://dx.doi.org/10.1023/A:1016046606831>.
- Manie, F., Piccin, O., and Ray, J. .P (1989). *Test report of the 2D Model M1 in the ONERA F1 Wind Tunnel*. Tech. rep. GARTEUR High-Lift Action Group AD(AG–08). GARTEUR.
- March, Andrew and Willcox, Karen (2012a). “Constrained Multifidelity Optimization Using Model Calibration”. In: *Structural and Multidisciplinary Optimization* 46.1, pp. 93–109. DOI: [10.1007/s00158-011-0749-1](https://doi.org/10.1007/s00158-011-0749-1).
- (2012b). “Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives”. In: *AIAA Journal* 50.5, pp. 1079–1089. DOI: [10.2514/1.J051125](https://doi.org/10.2514/1.J051125).
- Marler, R Timothy and Arora, Jasbir S (2004). “Survey of Multi-Objective Optimization Methods for Engineering”. In: *Structural and Multidisciplinary Optimization* 26.6, pp. 369–395. DOI: [10.1007/s00158-003-0368-6](https://doi.org/10.1007/s00158-003-0368-6).
- Mavriplis, Dimitri J et al. (2009). “Grid Quality and Resolution Issues from the Drag Prediction Workshop Series”. In: *Journal of Aircraft* 46.3, pp. 935–950.
- Mejía-Rodríguez, G., Renaud, J. E., and Tomar, V. (2010). “Multi-Objective Ceramic Matrix Composite Material Design Using the Variable Fidelity Model Management Optimization Framework”. In: *Engineering Optimization* 42.11, pp. 1055–1078. DOI: [10.1080/03052151003650100](https://doi.org/10.1080/03052151003650100). eprint: <http://dx.doi.org/10.1080/03052151003650100>.
- Melber-Wilkending, S et al. (2005). *Megaflow - Numerical Flow Simulation for Aircraft Design*. Ed. by Fassbender, Professor Dr. Norbert Kroll Dr.-Ing. Jens K. Vol. 89. ISBN 978-3-540-24383-0. Springer Berlin Heidelberg.
- Menter, F. R. (1994). “Two-equation eddy-viscosity turbulence models for engineering applications”. In: *AIAA Journal* 32.8, pp. 1598–1605.

- Meredith, P.T. (1991). “Viscous Phenomena Affecting High-Lift Systems and Suggestions for Future CFD Development”. In: *High Lift System Aerodynamics*. AGARD CP-515, pp. 19-1–19-8.
- Meunier, Mickaël (2009). “Simulation and Optimization of Flow Control Strategies for Novel High-Lift Configurations”. In: *AIAA journal* 47.5, pp. 1145–1157.
- Mifsud, M. J., MacManus, D. G., and Shaw, S.T. (2016). “A variable-fidelity aerodynamic model using proper orthogonal decomposition”. In: *International Journal for Numerical Methods in Fluids* 82.10. FLD-15-0283.R1, pp. 646–663. ISSN: 1097-0363. DOI: [10.1002/fld.4234](https://doi.org/10.1002/fld.4234). URL: <http://dx.doi.org/10.1002/fld.4234>.
- Newman III, James C et al. (1999). “Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations”. In: *Journal of Aircraft* 36.1, pp. 87–96. ISSN: 0021-8669. DOI: [10.2514/2.2416](https://doi.org/10.2514/2.2416). URL: <https://doi.org/10.2514/2.2416>.
- Niu, C. and Niu, M.C.Y. (1999). *Airframe Structural Design: Practical Design Information and Data on Aircraft Structures*. Airframe book series. Conmilit Press Limited. ISBN: 9789627128090. URL: <http://books.google.co.uk/books?id=yT15SwAACAAJ>.
- Nocedal, Jorge and Wright, Stephen J (1999). *Numerical Optimization*. Ed. by Glynn, Peter and Robinson, Stephen M. 1st. Springer.
- Oeuvray, Rodrigue and Bierlaire, Michel (2009). “Boosters: A Derivative-Free Algorithm Based on Radial Basis Functions”. In: *International Journal of Modelling and Simulation* 29.1, pp. 26–36. DOI: [10.1080/02286203.2009.11442507](https://doi.org/10.1080/02286203.2009.11442507).
- O’Hagan, A. (2006). “Bayesian Analysis of Computer Code Outputs: A Tutorial”. In: *Reliability Engineering & System Safety* 91.10-11. The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004) {SAMO} 2004 The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004), pp. 1290–1300. ISSN: 0951-8320. DOI: [10.1016/j.ress.2005.11.025](https://doi.org/10.1016/j.ress.2005.11.025). URL: <http://www.sciencedirect.com/science/article/pii/S0951832005002383>.
- Peherstorfer, Benjamin, Willcox, Karen, and Gunzburger, Max (2016). “Survey of multifidelity methods in uncertainty propagation, inference, and optimization”. In: *Preprint*, pp. 1–57.
- Perez, Ruben E., Jansen, Peter W., and Martins, Joaquim R. R. A. (2012). “PyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization”. In: *Structures and Multidisciplinary Optimization* 45.1, pp. 101–118. DOI: [10.1007/s00158-011-0666-3](https://doi.org/10.1007/s00158-011-0666-3).

- Pospelov, Alexis and Gubarev, Fedor (2014). “Nonlinear Multi-Objective Constrained Optimization: Using Second Order Approximation of Pareto Frontier Local Geometry in Descent-Diffusion Approach”. In: *20th Conference of the International Federation of Operational Research Societies (IFORS 2014)*.
- Queipo, Nestor V. et al. (2005). “Surrogate-Based Analysis and Optimization”. In: *Progress in Aerospace Sciences* 41.1, pp. 1–28. ISSN: 0376-0421. DOI: [10.1016/j.paerosci.2005.02.001](https://doi.org/10.1016/j.paerosci.2005.02.001). URL: <http://www.sciencedirect.com/science/article/pii/S0376042105000102>.
- Rasmussen, Carl Edward (2006). *Gaussian Processes for Machine Learning*. The MIT Press.
- Reckzeh, Daniel (2003). “Aerodynamic Design of the High-Lift-Wing for a Megaliner Aircraft”. In: *Aerospace Science and technology* 7.2, pp. 107–119. DOI: [10.1016/S1270-9638\(02\)00002-0](https://doi.org/10.1016/S1270-9638(02)00002-0).
- Rosenbrock, H. H. (1960). “An Automatic Method for Finding the Greatest or Least Value of a Function”. In: *The Computer Journal* 3.3, p. 175. DOI: [10.1093/comjnl/3.3.175](https://doi.org/10.1093/comjnl/3.3.175). eprint: [/oup/backfile/content_public/journal/comjnl/3/3/10.1093/comjnl/3.3.175/2/030175.pdf](http://oup/backfile/content_public/journal/comjnl/3/3/10.1093/comjnl/3.3.175/2/030175.pdf). URL: <http://dx.doi.org/10.1093/comjnl/3.3.175>.
- Rowan, Thomas Harvey (1990). “Functional Stability Analysis of Numerical Algorithms”. PhD thesis. University of Texas at Austin.
- Rudolph, Peter KC (1996). *High-Lift Systems on Commercial Subsonic Airlines*. Tech. rep. CR-4746. NASA.
- Rumsey, Christopher L. and Ying, Susan X. (2002). “Prediction of High Lift: Review of Present CFD Capability”. In: *Progress in Aerospace Sciences* 38.2, pp. 145–180. ISSN: 0376-0421. DOI: [10.1016/S0376-0421\(02\)00003-9](https://doi.org/10.1016/S0376-0421(02)00003-9). URL: <http://www.sciencedirect.com/science/article/pii/S0376042102000039>.
- Sacks, Jerome et al. (1989). “Design and Analysis of Computer Experiments”. In: *Statistical Science* 4.4, pp. 409–423. DOI: [10.1214/ss/1177012413](https://doi.org/10.1214/ss/1177012413). URL: <http://projecteuclid.org/euclid.ss/1177012413>.
- Schonlau, Matthias and Welch, William J (2006). “Screening the Input Variables to a Computer Model Via Analysis of Variance and Visualization”. In: *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics*. Ed. by Dean, Angela and Lewis, Susan. New York, NY: Springer New York, pp. 308–327. ISBN: 978-0-387-28014-1. DOI: [10.1007/0-387-28014-6_14](https://doi.org/10.1007/0-387-28014-6_14).

- Schwamborn, Dieter, Gerhold, Thomas, and Heinrich, Ralf (2006). “The DLR Tau-Code: Recent Applications in Research and Industry”. In: *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS). Egmond aan Zee, The Netherlands.
- Sederberg, Thomas W and Parry, Scott R (1986). “Free-Form Deformation of Solid Geometric Models”. In: *ACM Siggraph Computer Graphics*. Vol. 20. 4. ACM, pp. 151–160.
- Sefrioui, M, Srinivas, K, and Periaux, J (2000). “Aerodynamic Shape Optimization Using a Hierarchical Genetic Algorithm”. In: *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS*. Barcelona, Spain, p. 20.
- Shamos, Michael Ian and Hoey, Dan (1976). “Geometric Intersection Problems”. In: *17th Annual Symposium on Foundations of Computer Science*, pp. 208–215. DOI: [10.1109/SFCS.1976.16](https://doi.org/10.1109/SFCS.1976.16).
- Shan, Songqing and Wang, G. Gary (2010a). “Metamodeling for High Dimensional Simulation-Based Design Problems”. In: *Journal of Mechanical Design* 132.5, pp. 051009–051009. ISSN: 1050-0472. DOI: [10.1115/1.4001597](https://doi.org/10.1115/1.4001597).
- (2010b). “Survey of Modeling and Optimization Strategies to Solve High-Dimensional Design Problems with Computationally-Expensive Black-Box Functions”. English. In: *Structural and Multidisciplinary Optimization* 41.2, pp. 219–241. ISSN: 1615-147X. DOI: [10.1007/s00158-009-0420-2](https://doi.org/10.1007/s00158-009-0420-2).
- Singh, P. et al. (2017). “Multi-objective Geometry Optimization of a Gas Cyclone Using Triple-Fidelity Co-Kriging Surrogate Models”. English. In: *Journal of Optimization Theory and Applications*. cited By 0; Article in Press, pp. 1–22. DOI: [10.1007/s10957-017-1114-3](https://doi.org/10.1007/s10957-017-1114-3). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85019098275&doi=10.1007%2fs10957-017-1114-3&partnerID=40&md5=0856f9c742144c7d1105c710185e6ad3>.
- Smith, A. M. O. (1975). “High-Lift Aerodynamics”. In: *Journal of Aircraft* 12.6, pp. 501–530. DOI: [10.2514/3.59830](https://doi.org/10.2514/3.59830).
- Sobieczky, Helmut (1999). “Parametric Airfoils and Wings”. In: *Recent Development of Aerodynamic Design Methodologies*. Springer, pp. 71–87.
- Spalart, Philipe R, Allmaras, Steven R, et al. (1994). “A one equation turbulence model for aerodynamic flows”. In: *La Recherche Aéronautique* 1, pp. 5–21.
- Thibert, J. J. (1991). “The Garteau High Lift Research Programme”. In: *High Lift System Aerodynamics*. AGARD CP-515, pp. 16-1–16-21.

- Timme, S., Marques, S., and Badcock, K. J. (2011). “Transonic Aeroelastic Stability Analysis Using a Kriging-Based Schur Complement Formulation”. In: *AIAA Journal* 49.6, pp. 1202–1213. ISSN: 0001-1452. DOI: [10.2514/1.j050975](https://doi.org/10.2514/1.j050975). URL: <https://doi.org/10.2514/1.J050975>.
- Tooley, Mike et al. (2009). *Aerospace Engineering E-Mega Reference*. Butterworth-Heinemann.
- Trapani, G, Kipouros, T, and Savill, AM (2012). “The design of multi-element airfoils through multi-objective optimization techniques”. In: *CMES-Computer Modeling in Engineering and Sciences* 88, pp. 107–138.
- Trapani, G., Kipouros, T., and Savill, A.M. (2012). “The design of multi-element airfoils through multi-objective optimization techniques”. English. In: *CMES - Computer Modeling in Engineering and Sciences* 88.2. cited By 6, pp. 107–138. ISSN: 15261492. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84872101439&partnerID=40&md5=873b334d6da1c6221dbf6821858633b1>.
- Trapani, Giuseppe (2014). “The Design of High-Lift Aircraft Configurations through Muti-Objective Optimisation”. PhD thesis. Cranfield University.
- Umakant, J. et al. (2007). “Ranking Based Uncertainty Quantification for a Multifidelity Design Approach”. In: *Journal of Aircraft* 44.2, pp. 410–419. ISSN: 0021-8669. DOI: [10.2514/1.22424](https://doi.org/10.2514/1.22424). URL: <https://doi.org/10.2514/1.22424>.
- Vassberg, John C. and Jameson, Antony (2014). *Industrial Applications of Aerodynamic Shape Optimization*. Tech. rep. Von Karman Institute.
- Wahls, Rich (2010). “N+3 Technologies and Concepts”. In: *Green Aviation Summit*.
- Walter, Éric (2014). *Numerical Methods and Optimization*. How About a Free Lunch? Springer International Publishing. DOI: [10.1007/978-3-319-07671-3](https://doi.org/10.1007/978-3-319-07671-3).
- Wang, D. et al. (2014). “Aerodynamic shape optimization design of the swept wing based on the Kriging surrogate model”. In: *Applied Mechanics and Materials* 444-445. cited By 1, pp. 1277–1282. DOI: [10.4028/www.scientific.net/AMM.444-445.1277](https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887344044&doi=10.4028%2fwww.scientific.net%2fAMM.444-445.1277&partnerID=40&md5=c7d38f329e088e5196a458f6c6caf5d8). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887344044&doi=10.4028%2fwww.scientific.net%2fAMM.444-445.1277&partnerID=40&md5=c7d38f329e088e5196a458f6c6caf5d8>.
- Wang, G. Gary and Shan, S. (2006). “Review of Metamodeling Techniques in Support of Engineering Design Optimization”. In: *Journal of Mechanical Design* 129.4, pp. 370–380. ISSN: 1050-0472. DOI: [10.1115/1.2429697](https://doi.org/10.1115/1.2429697).
- Wei, Zhang and Meijian, Sun (2014). “Design Optimization of Aerodynamic Shapes of a Wing and Its Winglet Using Modified Quantum-Behaved Particle Swarm Op-

- timization Algorithm”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 228.9, pp. 1638–1647. DOI: [10.1177/0954410013499841](https://doi.org/10.1177/0954410013499841). eprint: <http://dx.doi.org/10.1177/0954410013499841>.
- Wilcox, David C (1988). “Reassessment of the scale-determining equation for advanced turbulence models”. In: *AIAA Journal* 26.11, pp. 1299–1310. ISSN: 0001-1452. DOI: [10.2514/3.10041](https://doi.org/10.2514/3.10041). URL: <https://doi.org/10.2514/3.10041>.
- (1998). *Turbulence Modeling for Cfd*. Vol. 2. DCW industries La Canada, CA.
- Wild, Jochen (1998). “Direct Optimization of Multi-Element Airfoils for High-Lift Using Navier-Stokes Equations”. In: *Computational Fluid Dynamics 98 - Proceedings of the 4th European Computational Fluid Dynamics Conference*. Chichester, UK, pp. 383–390.
- (2002). “Validation of Numerical Optimization of High-Lift Multi-Element Airfoils Based on Navier-Stokes-Equations”. In: *20th AIAA Applied Aerodynamics Conference*. Vol. 2939. paper 2002-2939. St. Louis, USA, p. 2002.
- (2008). “Multi-Objective Constrained Optimisation in Aerodynamic Design of High-Lift Systems”. In: *International Journal of Computational Fluid Dynamics* 22.3, pp. 153–168. DOI: [10.1080/10618560701868420](https://doi.org/10.1080/10618560701868420).
- Wild, Jochen and Brezillon, Joël (2005). “Application of Multi Objective Constrained Optimization in Aerodynamic High Lift Design”. In: *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems EUROGEN 2005*. Munich, Germany.
- Wild, Stefan Martin, Regis, Rommel G, and Shoemaker, Christine A. (2008). “ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions”. In: *SIAM Journal on Scientific Computing* 30.6, pp. 3197–3219. DOI: [10.1137/070691814](https://doi.org/10.1137/070691814).
- Wild, Stefan Martin and Shoemaker, Christine A. (2013). “Global Convergence of Radial Basis Function Trust-Region Algorithms for Derivative-Free Optimization”. In: *SIAM Review* 55.2, pp. 349–371. DOI: [10.1137/120902434](https://doi.org/10.1137/120902434).
- Yang, BS, Yeun, Y-S, and Ruy, W-S (2002). “Managing Approximation Models in Multiobjective Optimization”. In: *Structural and Multidisciplinary Optimization* 24.2, pp. 141–156.
- Ye, Kenny Q. (1998). “Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments”. In: *Journal of the American Statistical Association* 93.444, pp. 1430–1439. DOI: [10.1080/01621459.1998.10473803](https://doi.org/10.1080/01621459.1998.10473803). eprint: <http://www.tandfonline.com/doi/pdf/10.1080/01621459.1998.10473803>.

Bibliography

- Yuan, Ya-Xiang (1999). “A Review of Trust Region Algorithms for Optimization”. In: *4th international congress on industrial and applied mathematics*, pp. 271–282.
- Zhu, Jiandao, Wang, Yi-Jen, and Collette, Matthew (2014). “A Multi-Objective Variable-Fidelity Optimization Method for Genetic Algorithms”. In: *Engineering Optimization* 46.4, pp. 521–542. DOI: [10.1080/0305215X.2013.786063](https://doi.org/10.1080/0305215X.2013.786063). eprint: <http://dx.doi.org/10.1080/0305215X.2013.786063>.

Publications

A.1 Published Conference Papers

Demange, Jean, Savill, A. Mark, and Kipouros, Timoleon (2016b). “Multifidelity Optimization for High-Lift Airfoils”. In: *54th AIAA Aerospace Sciences Meeting*. paper 2016-0557. San Diego, USA: AIAA, p. 0557. DOI: [10.2514/6.2016-0557](https://doi.org/10.2514/6.2016-0557)

Demange, Jean, Savill, A. Mark, and Kipouros, Timoleon (2016a). “A Multifidelity Multiobjective Optimization Framework for High-Lift Airfoils”. In: *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. paper 2016-3367. Washington, USA: AIAA, p. 3367. DOI: [10.2514/6.2016-3367](https://doi.org/10.2514/6.2016-3367)

Kontogiannis, Spyridon G., Demange, Jean, Kipouros, Timoleon, and Savill, A. Mark (2018). “A Comparison Study of Two Multifidelity Methods for Aerodynamic Optimization”. In: *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. paper 2018-0415. Florida, USA: AIAA, p. 0415. DOI: [10.2514/6.2018-0415](https://doi.org/10.2514/6.2018-0415)

A.2 Journal Papers in Preparation

Demange, Jean, Savill, A. Mark, and Kipouros, Timoleon (2017b). “A Trust-Region-Based Multifidelity Multiobjective Optimization Framework for High-Lift Airfoil Design”. In preparation

Demange, Jean, Savill, A. Mark, and Kipouros, Timoleon (2017a). “A Trust-Region-Based Co-Kriging Multifidelity Multiobjective Particle Swarm Optimizer for High-Lift Airfoil Design”. In preparation

A.3 Presentations & Posters

Demange, Jean (2015). “Multifidelity Optimisation For High-Lift Aerofoils”. Airbus DiPaRT

Appendix A Publications

Demange, Jean (2016). “Multifidelity Optimisation For High-Lift Aerofoils: Effect of Low-Fidelity Model”. Airbus DiPaRT

Demange, Jean (2017). “Multifidelity Multiobjective Optimisation for High-Lift Aerofoils”. Airbus PhD day

Code Implementation

B.1 MultiObjective Particle Swarm Optimiser JS

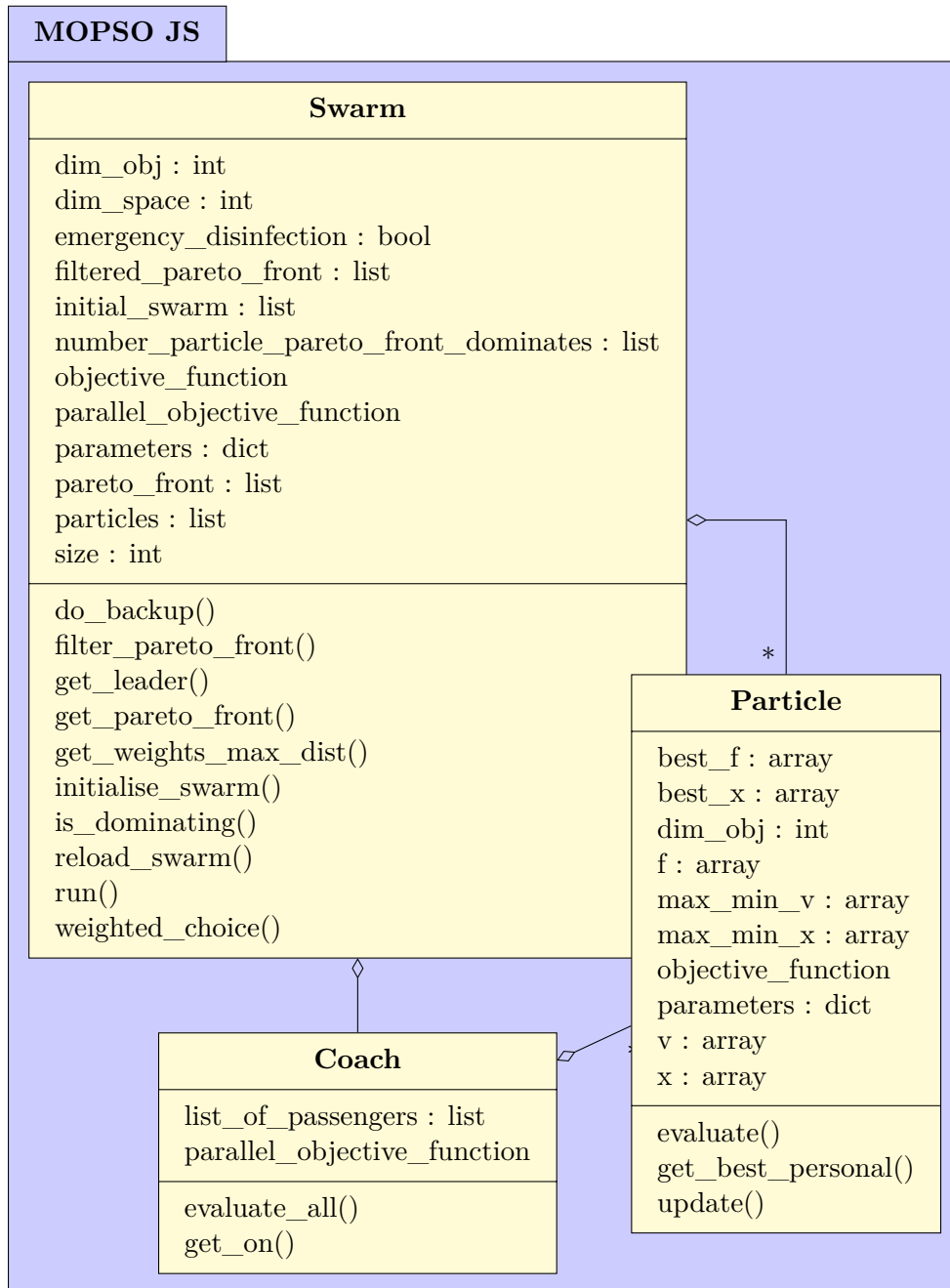


Figure B.1: MOPSO JS simplified UML diagram.

B.2 Surrogate Models

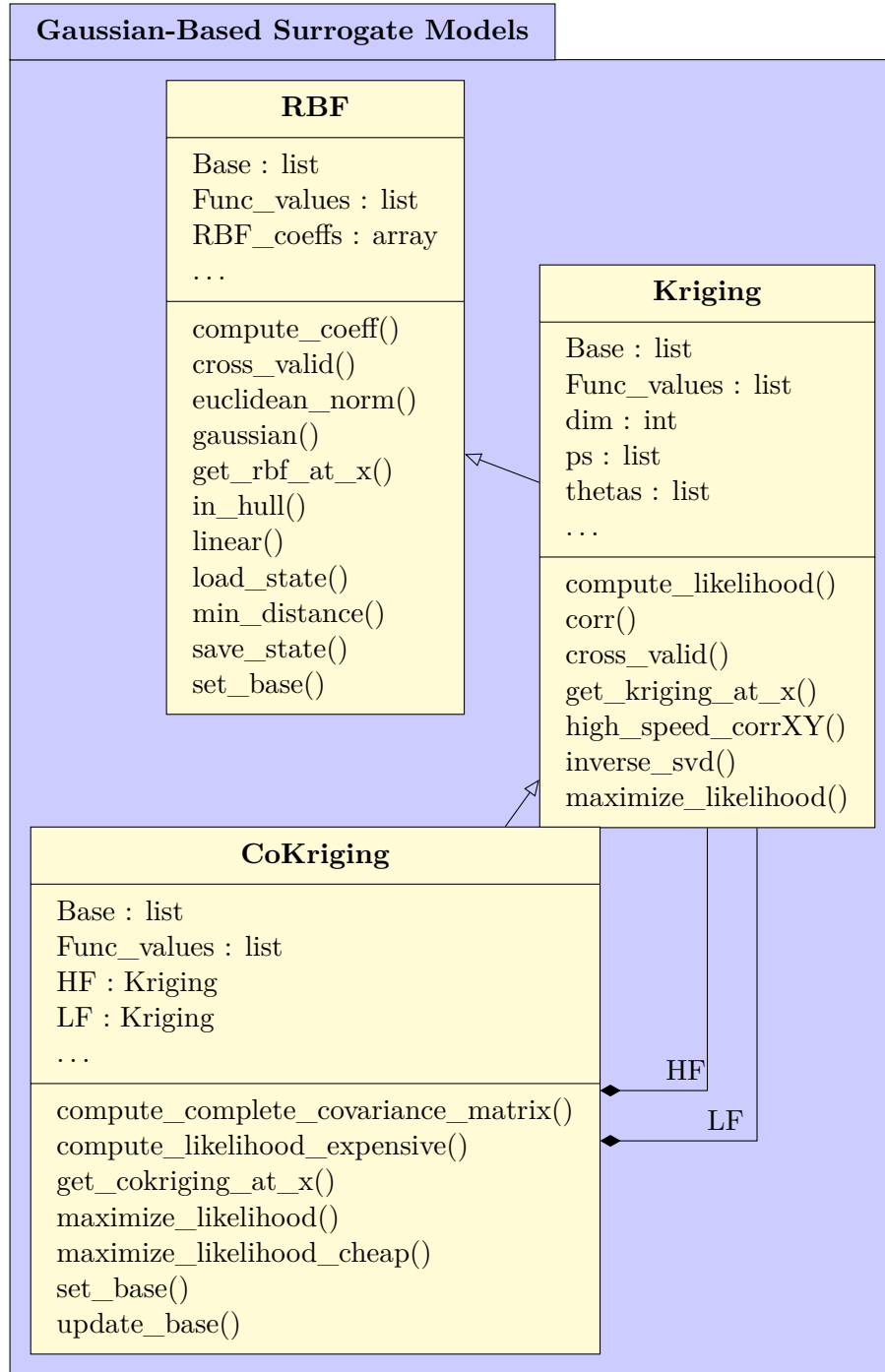


Figure B.2: Surrogate models simplified UML diagram.

B.3 Single-Objective MultiFidelity

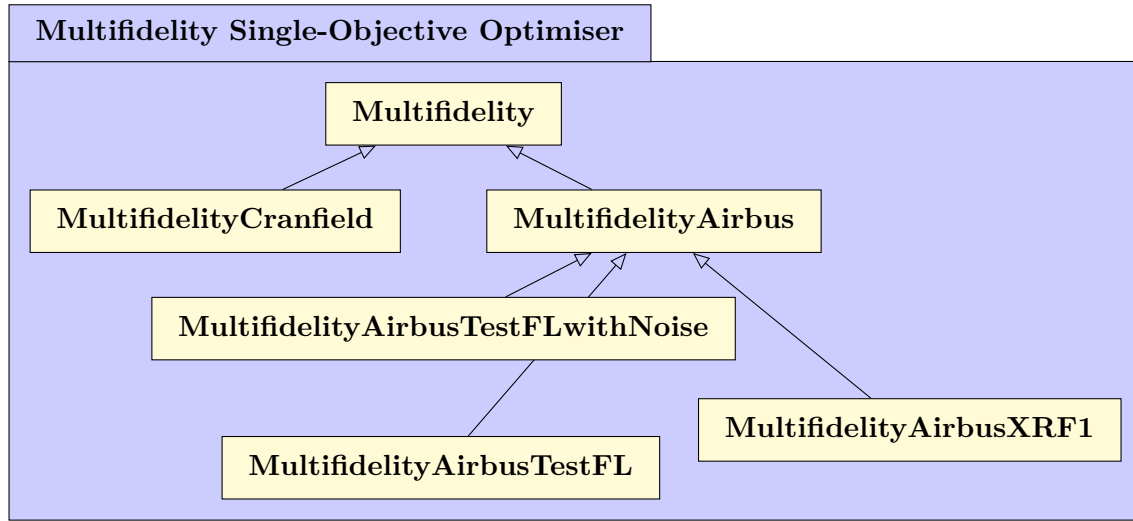


Figure B.3: Single-objective multifidelity code simplified UML diagram.

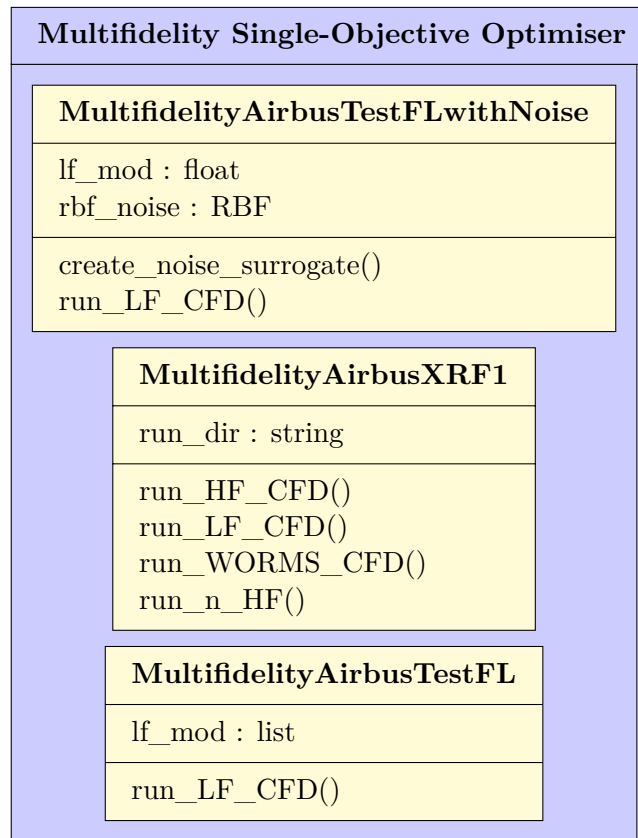


Figure B.4: Single-objective multifidelity code detailed UML diagram.

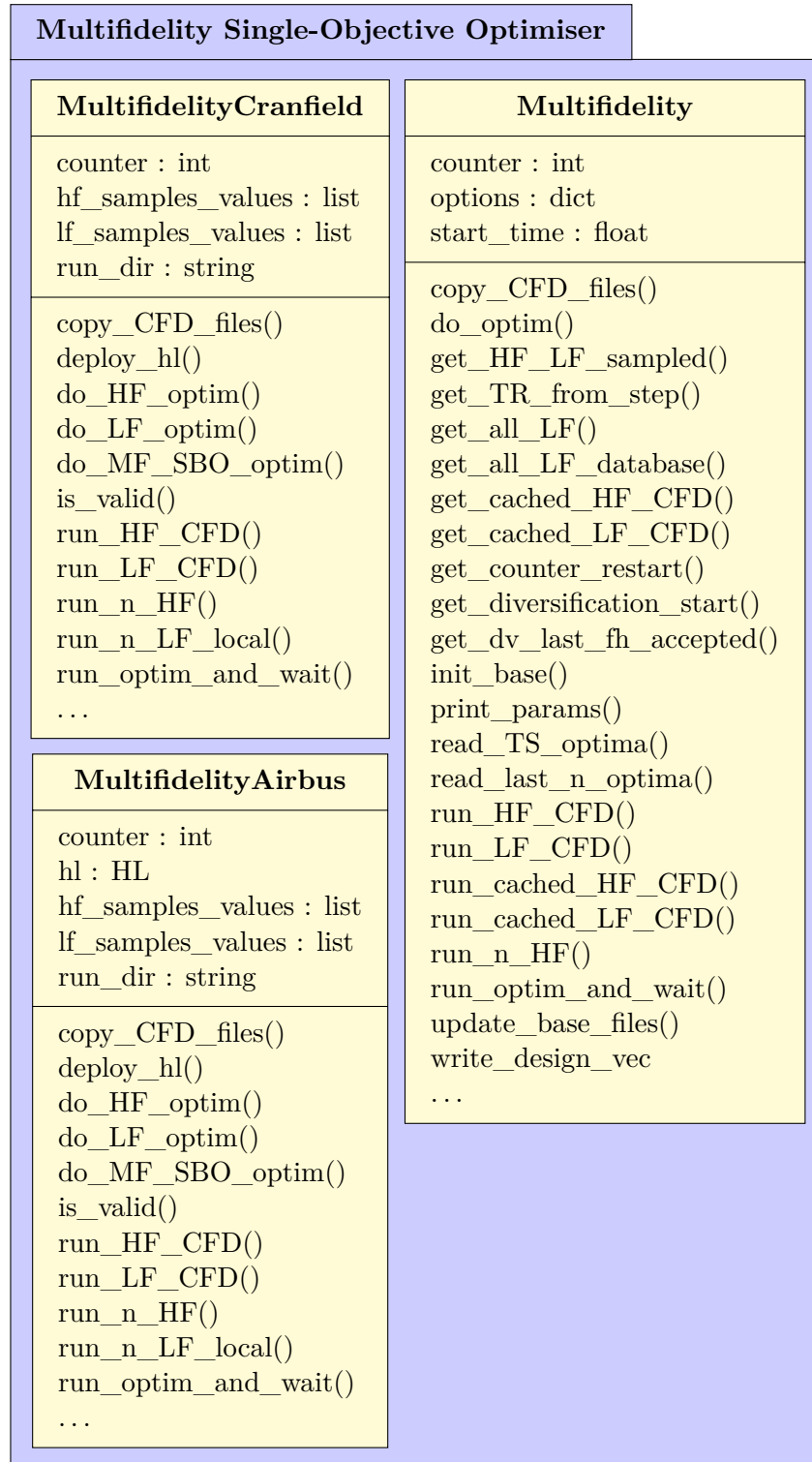


Figure B.5: Single-objective multifidelity code detailed UML diagram (end).

B.4 MultiObjective MultiFidelity

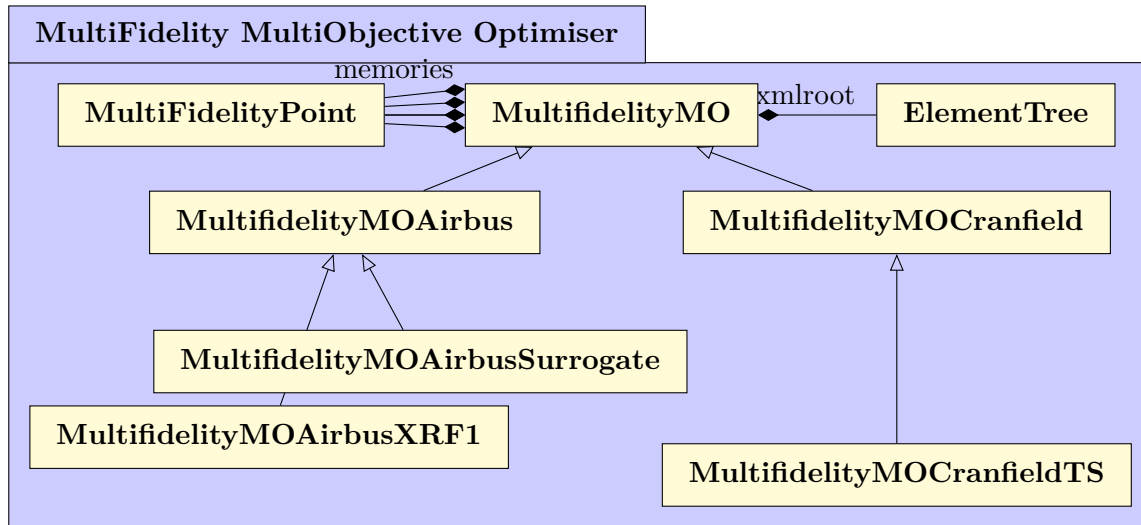


Figure B.6: MultiObjective multifidelity code simplified UML diagram.

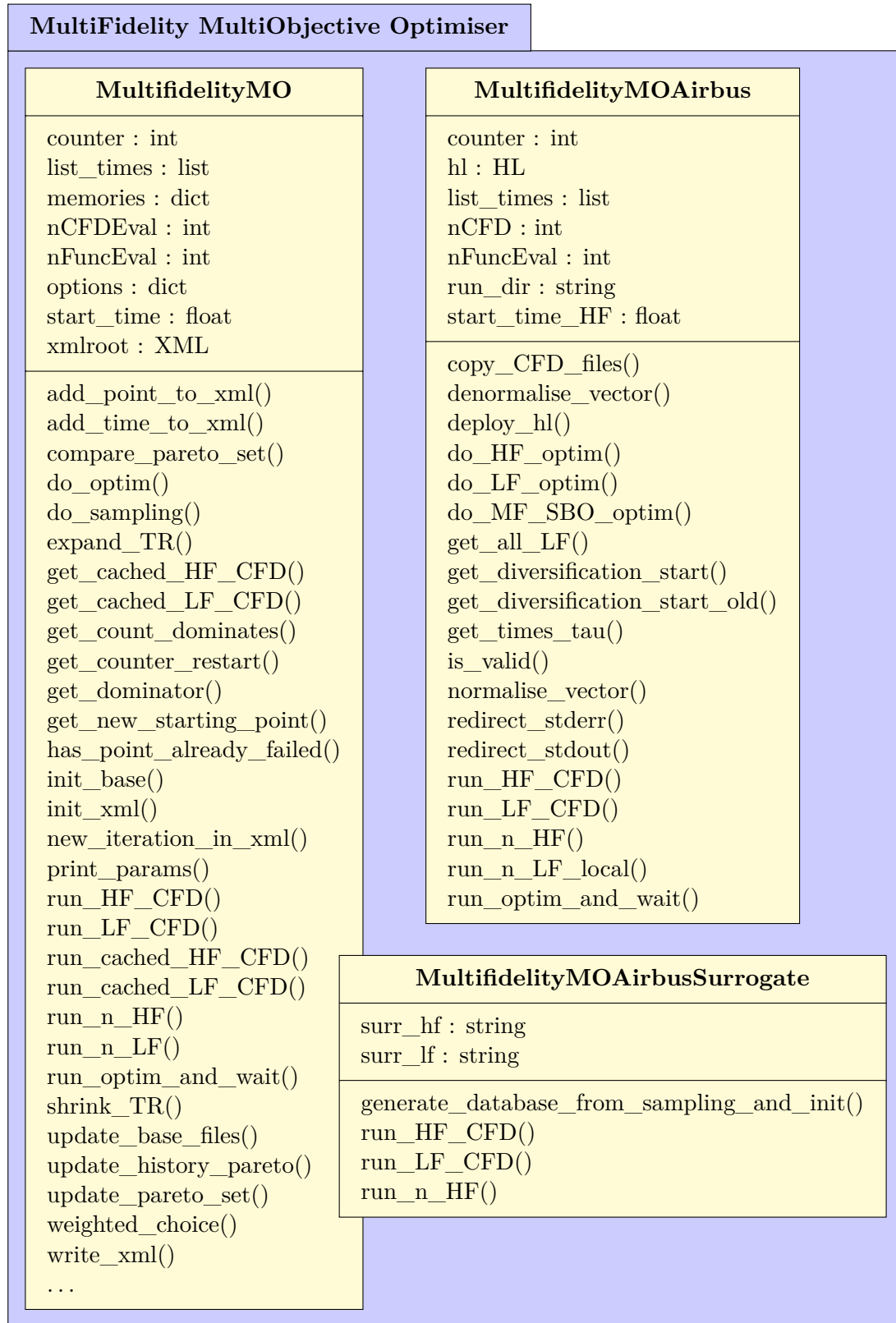


Figure B.7: MultiObjective multifidelity code detailed UML diagram (end).

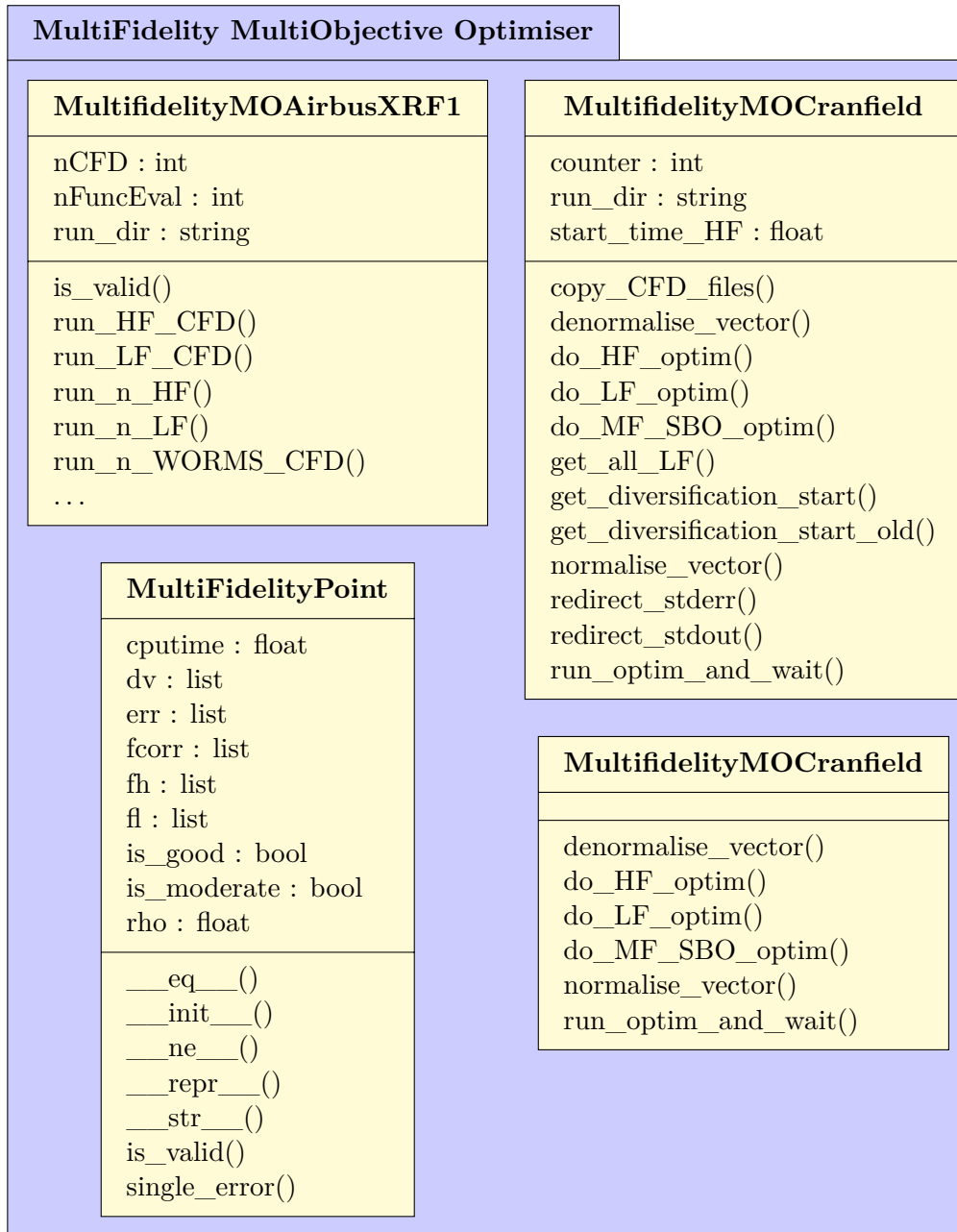


Figure B.8: MultiObjective multifidelity code detailed UML diagram.

Extra Figures

C.1 Multifidelity Single-Objective Optimisation

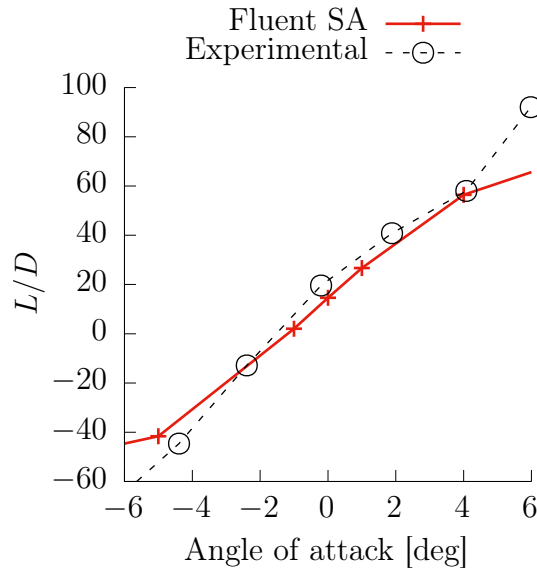


Figure C.1: Fluent (high-fidelity) and wind tunnel validation.

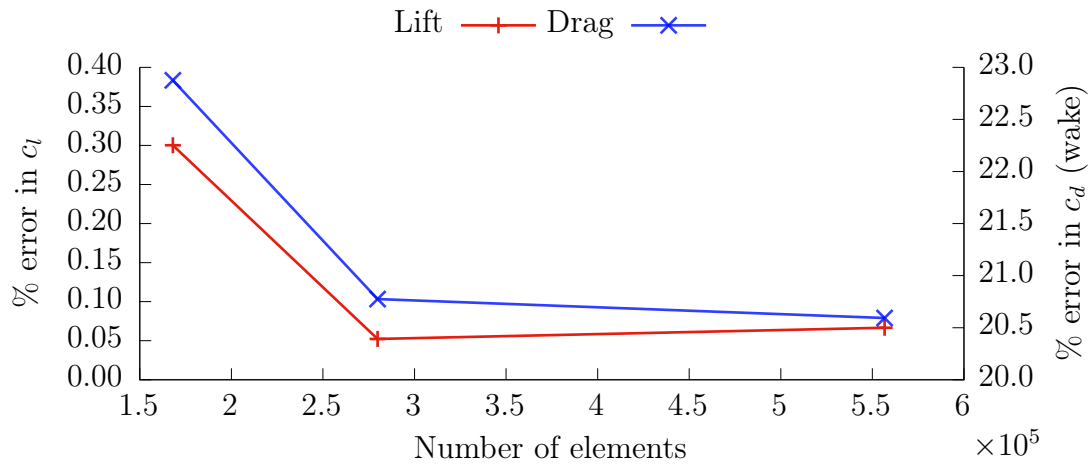


Figure C.2: Grid convergence for the Garteur Fluent CFD.

C.2 Multifidelity Multiobjective Optimisation

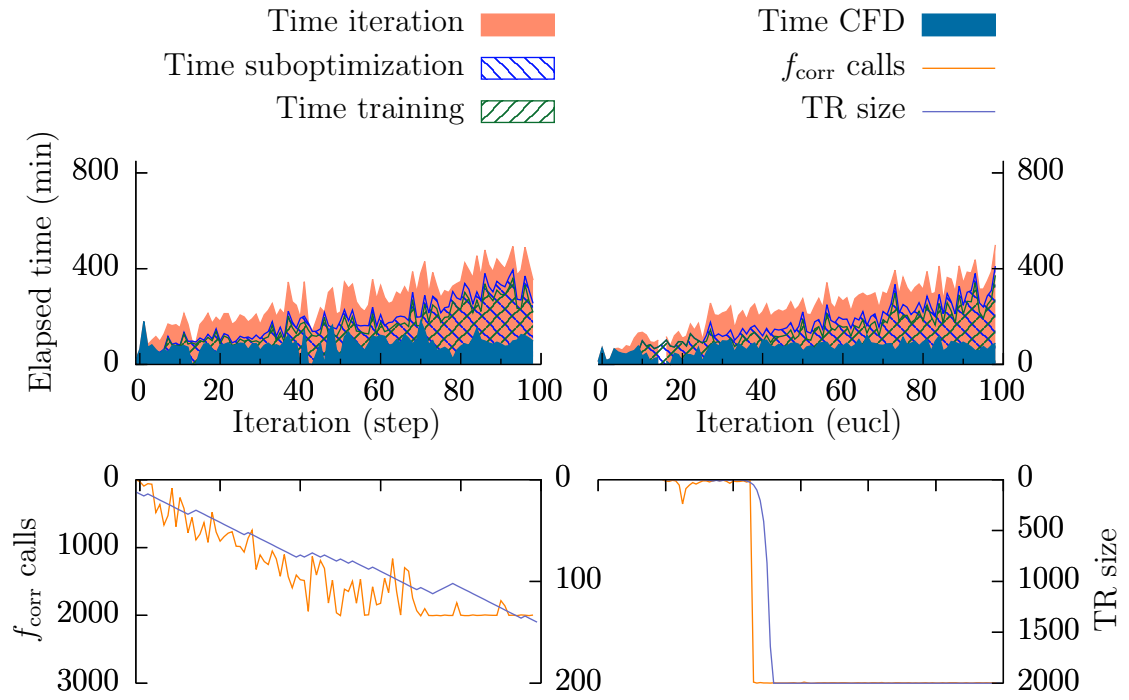


Figure C.3: Time repartition for the multifidelity optimisation when ESDU is used as low-fidelity tool (top) and trust region size along with number of high-fidelity function calls.

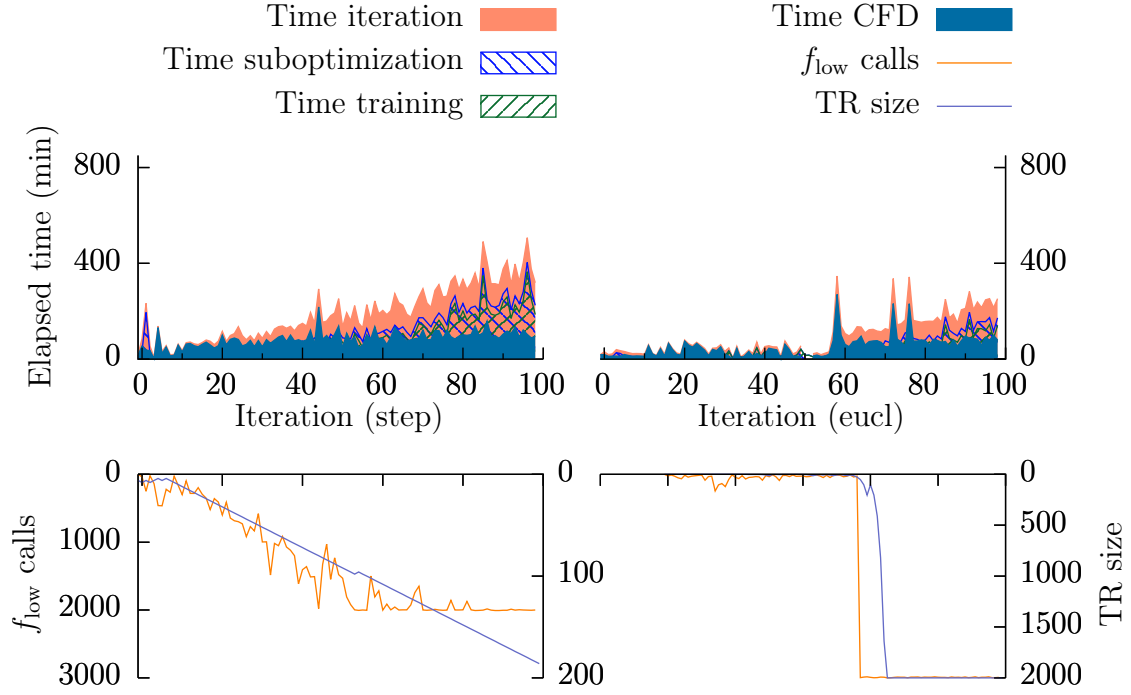


Figure C.4: Time repartition for the multifidelity optimisation when MSES is used as low-fidelity tool without low-fidelity sample (top) and trust region size along with number of high-fidelity function calls.

C.3 Influence of the low-fidelity tool accuracy on convergence

C.3.1 Lack of Sensitivity

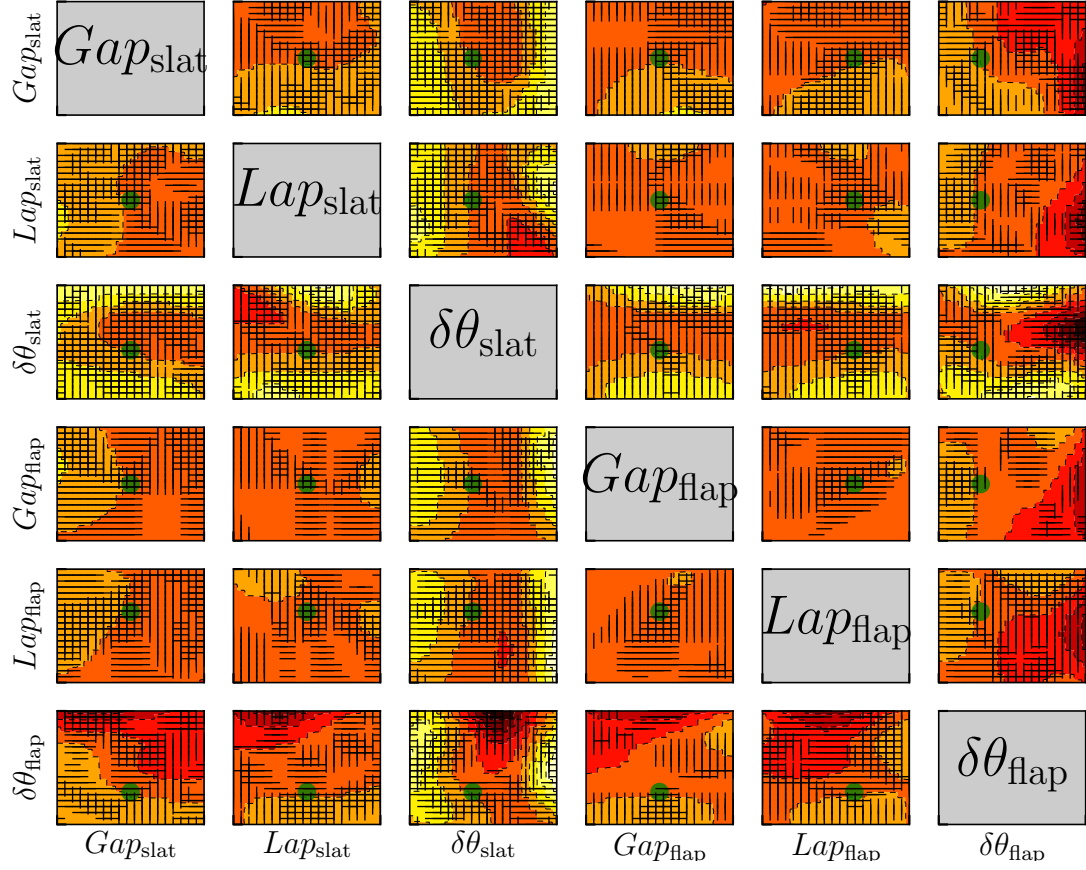


Figure C.5: Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with no sensitivity on the slat (modified low-fidelity)

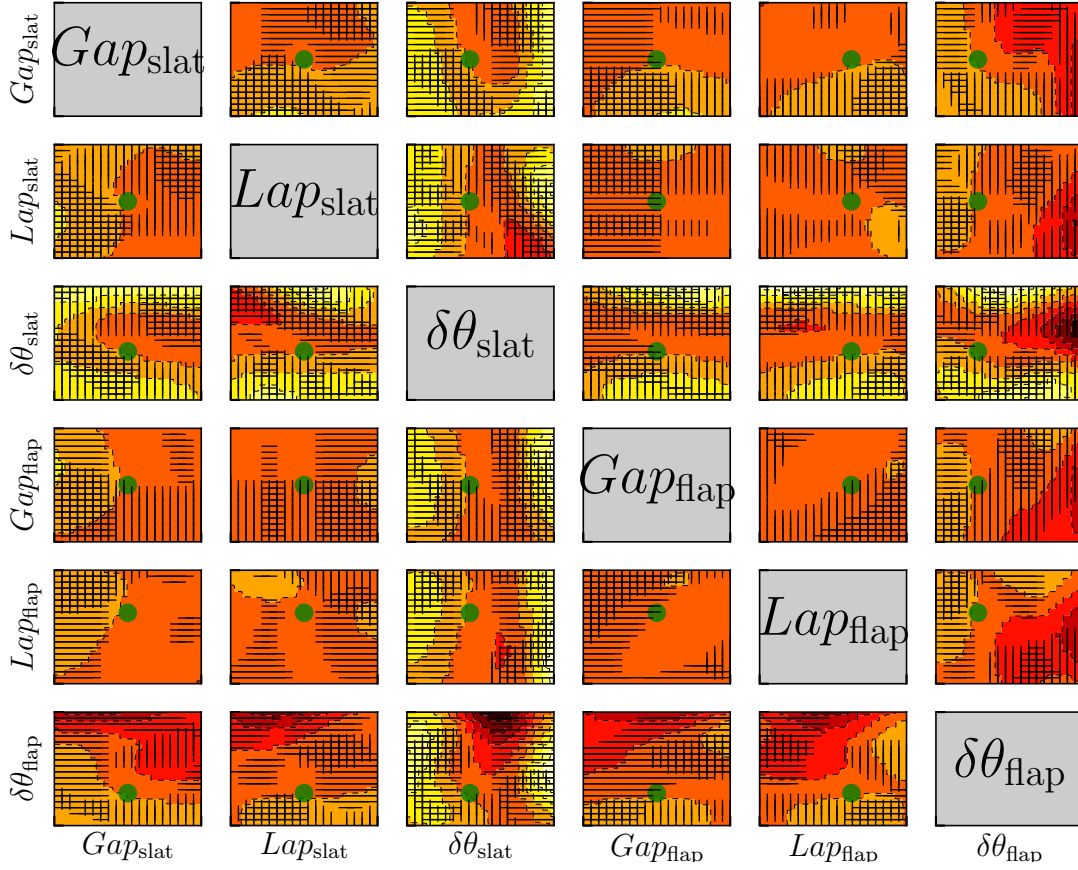


Figure C.6: Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with no sensitivity on the flap (modified low-fidelity)

C.3.2 Reverse Sensitivity

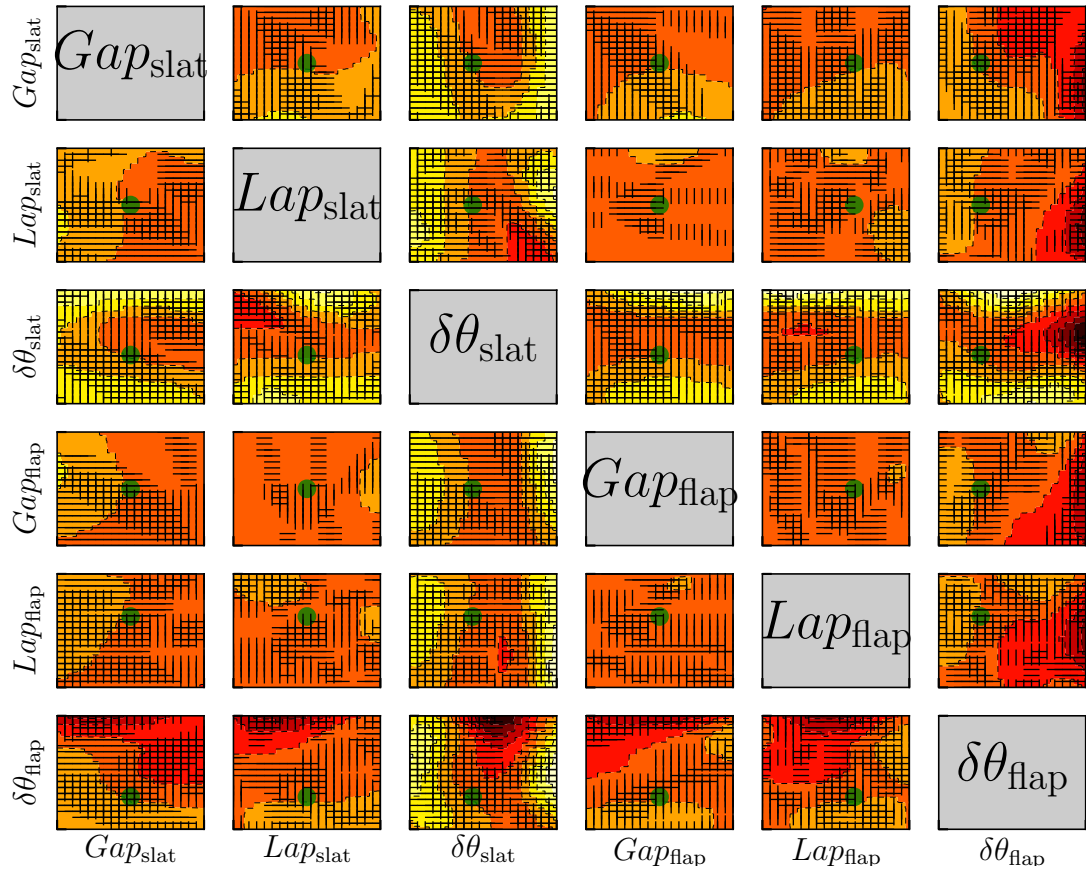


Figure C.7: Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with reverse sensitivity on the slat (modified low-fidelity)

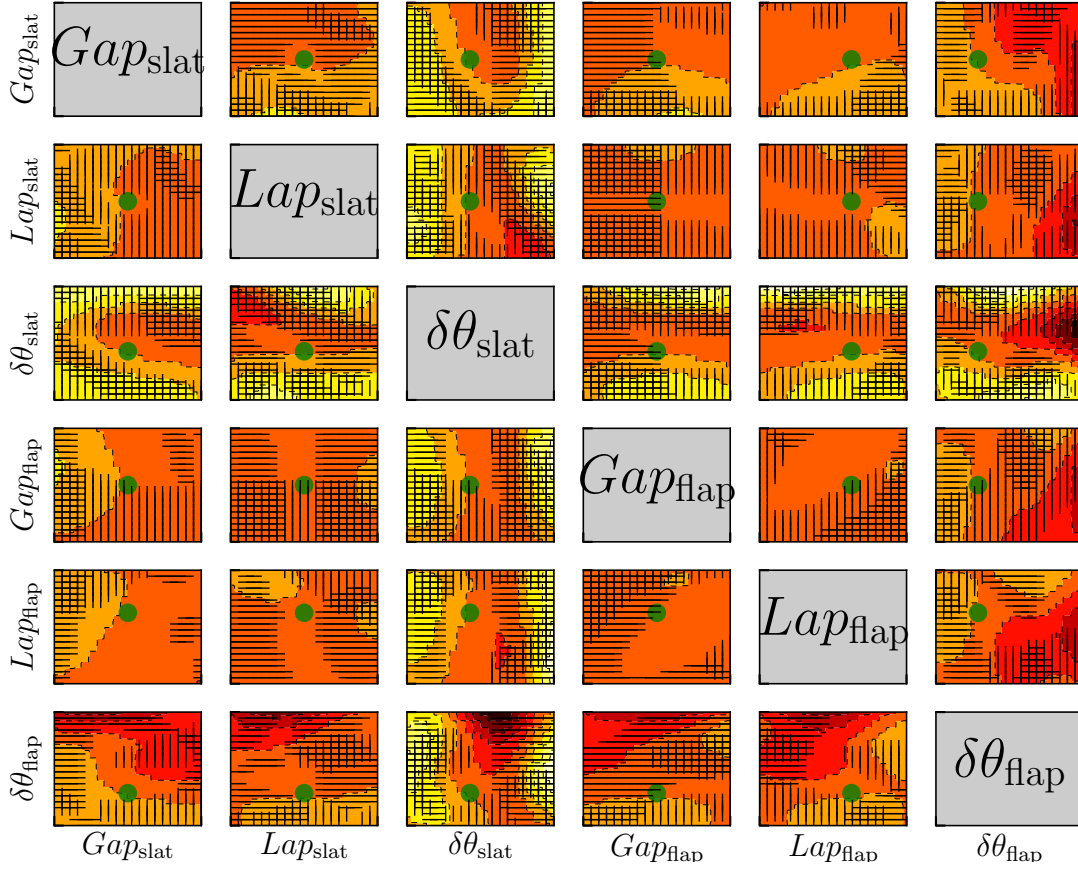


Figure C.8: Multifidelity matrix contour plot (starting point) for the maximum c_l for TAU (high-fidelity) and HiLi with reverse sensitivity on the flap (modified low-fidelity)

C.4 Three-Dimensional Transonic Wing Optimisation

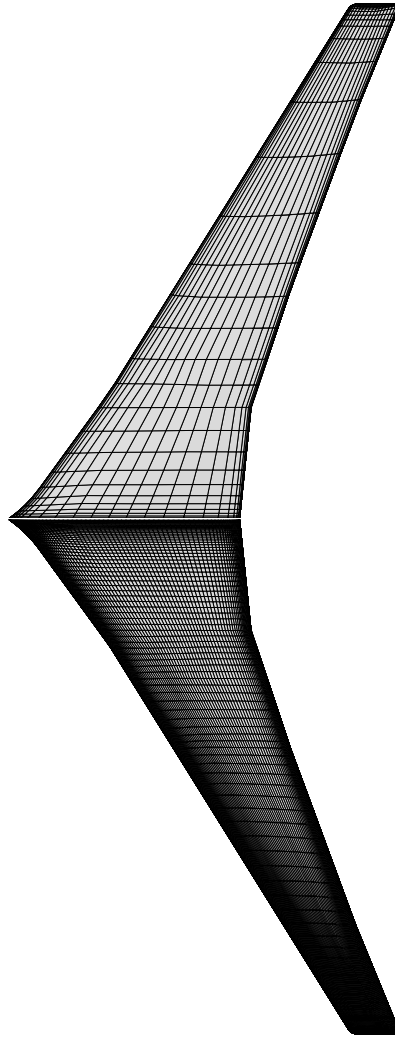


Figure C.9: Coarse and fine meshes of the XRF1 wing.

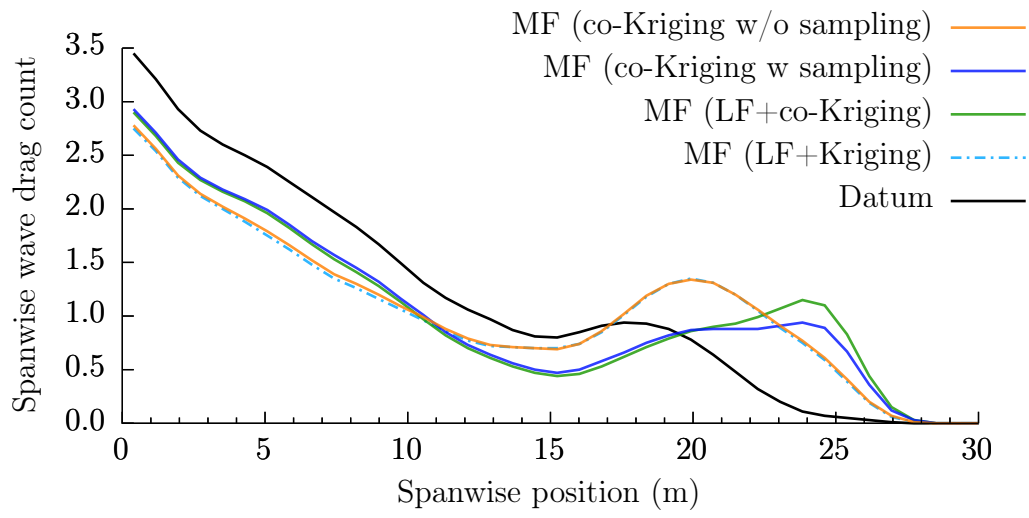


Figure C.10: Spanwise wave drag distribution for XRF1 the optimum configurations.

Maximum Lift Estimation from Clean Configuration

D.1 Generating High-Lift Aerofoil From a Clean Configuration

An aircraft initial design is usually done by the planform definition and the high-level aircraft performance estimations. To increase the fidelity of the maximum lift prediction, a geometry needs to be created prior the CAD shapes definition. The clean aerofoil and the high-lift devices are created with parametrisations originating from previous proven designs and used with different aerodynamic tools for the performances prediction. The rough outline of the framework is described in figure D.1 and rapidly enumerated below:

1. *Clean aerofoil*: for each section described by the planform, a two-dimensional aerofoil is created. In the work implemented, aerofoils are based on Bezier shapes. The shape could also be fixed and defined by the clean geometry. Here already, potential for optimisation is present.
2. *Cut-outs*: from the clean aerofoil, high-lift devices need to be created according to chapter 2 description. In the work implemented, the cut outs are also parametrised by Bezier curves, and can also be optimised as part of a multifidelity optimisation. Figure D.2 shows an example of such cuts for a slat and a flap¹.
3. *Deploy devices*: this step has extensively been described in this thesis: it corresponds to the free optimisation of the flap and slat positions to obtain the maximum lift.
4. *Create kinematics*: once optimal maximum deployment has been found, the kinematics need to be fitted. Depending on the number of positions to fit (eg for take-off or landing), the kinematic type (see chapter 2) is selected and optimised. An example of drop hinge slat and flap fitted on a maximum deflection configuration is shown in figure D.3 for two gap, overlap and deflection configurations.
5. *Loads*: from the aerodynamics, the loads could be extracted and propagated in the kinematic system for sizing.

¹Notice the first order continuity between slat, main and flap elements.

The final result of such integrated framework would be the maximum achievable lift, the best devices positions, the kinematics that suits the best, the loads for actuator sizing and the CFD databases.

A part of figure D.1 can be covered by multifidelity optimisation. This is the subject of the final section.

D.2 Kinematics and Multifidelity Optimisation

The high-lift devices position is crucial for take-off and landing but it influences greatly the systems used for the deployment. However the major part of the flight is spent in cruise and therefore deployment systems have to be chosen so that it does not impact significantly the cruise performances. Indeed, the more freedom in the position, the more complex the system is, and a more complex system means a higher weight, power and size which impact the overall aircraft. Therefore the deployment systems must be considered during the optimisation. A first step towards this can be achieved by including kinematics inside the optimisation loop.

The simple drop hinge implementation is firstly presented and then a method to include it in the optimisation is sketched.

Simple drop hinge In its review, Rudolph (1996) highlights that most of the commercial aircraft are using a *three-position slat* that have different deployment settings for take-off and landing. This can be achieved with tracks like in recent Airbus aircraft (Reckzeh 2003) despite their complexity but also with a single hinge located underneath the wing. This system has the advantage of being simple and therefore not costly to maintain, however it is quite restrictive on its deployed positions.

To include the kinematics in the aerodynamic design of the high-lift devices, one can first find the maximum deflected position, optimised on gap and lap, which targets the landing configuration (maximum lift), to find the hinge position. Once known, an intermediate deflection can be optimised for take-off, that usually requires compromise between maximum lift and lift-over-drag ratio. The method to find the hinge point h is shown in figure D.4: the hinge location is found with the stowed (blue) and deployed configuration (red).

A framework was developed, initially for the sponsor, to generate a high-lift profile for any deflection angle according to the single-hinge kinematics law. The framework uses as input a one-element profile and few parameters. The slat and flap are then generated according to the parameter and input geometry. The flap and slat can be deployed according to a hinge point, either manually given, or from a stowed and deployed configuration (as shown in figure D.4). Then the flap and slat are moved around the hinge point to generate the intermediate geometries.

Multifidelity kinematics-aerodynamics optimisation As stated previously, an optimisation on high-lift devices requires the kinematic to be industrially feasible.

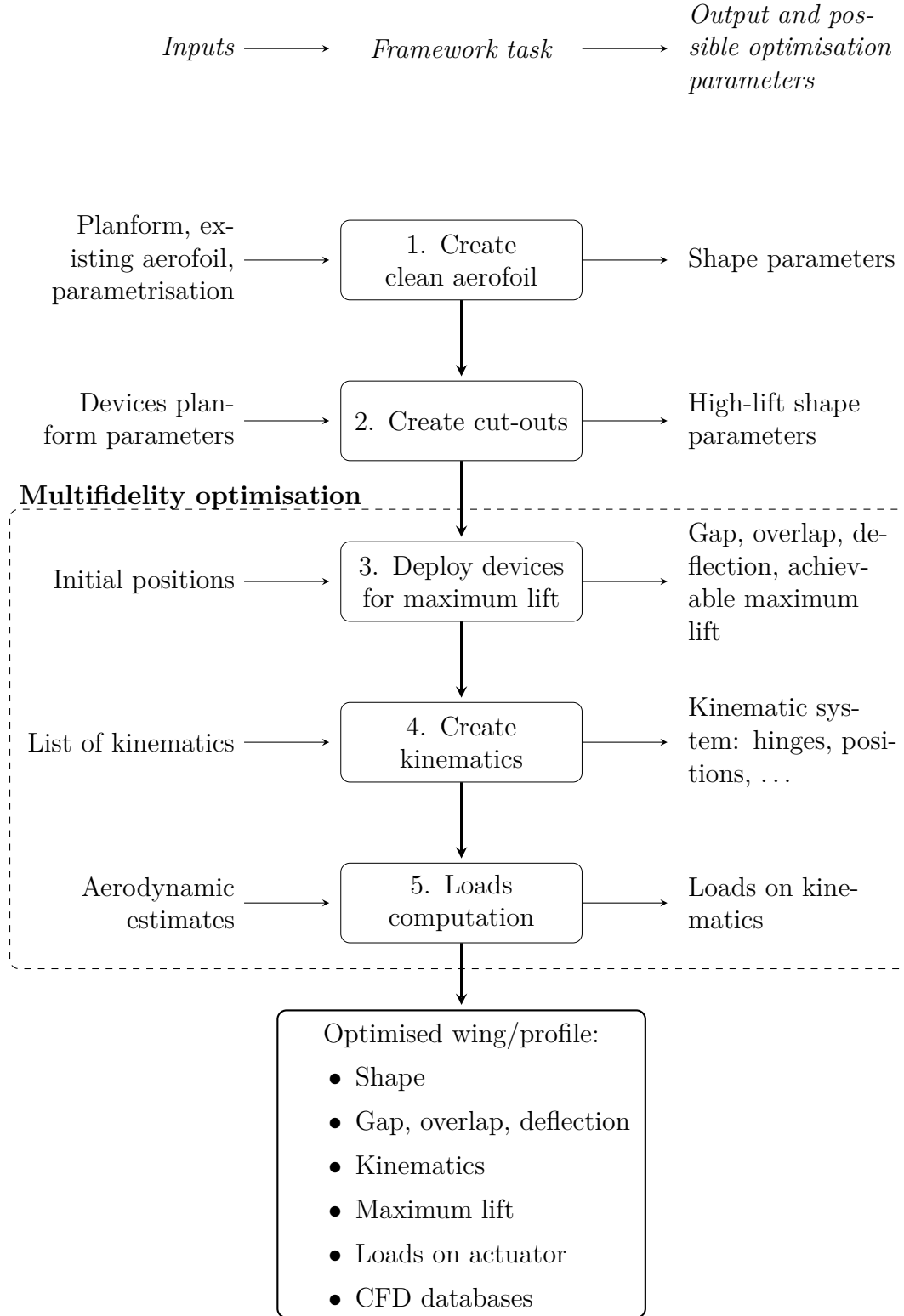


Figure D.1: High-level framework for load and maximum lift estimation from a clean aerofoil.

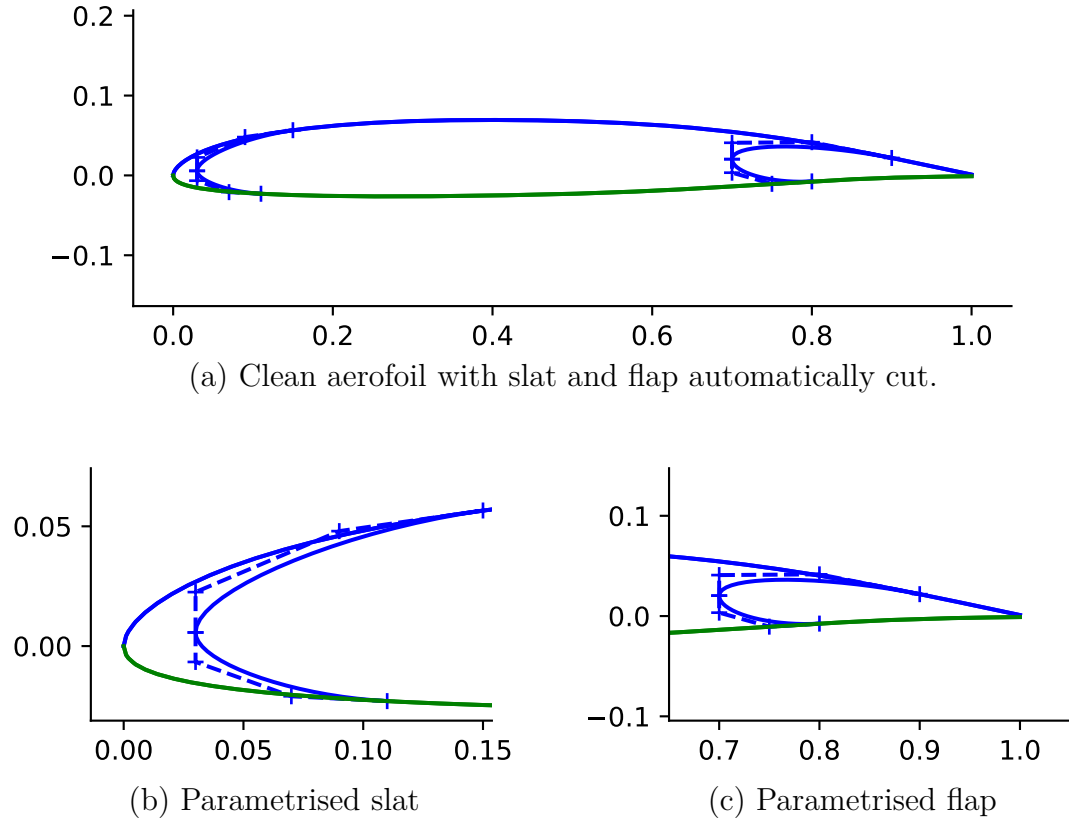


Figure D.2: Clean aerofoil and parametrised slat and flap.

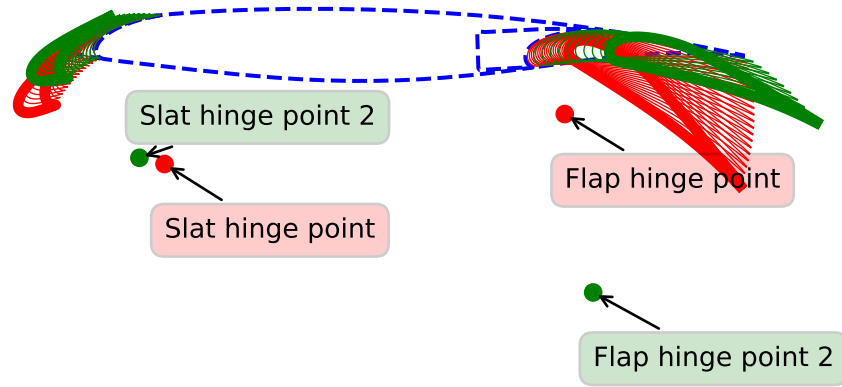


Figure D.3: Example of two maximum deflection configurations leading to two single hinge positions.

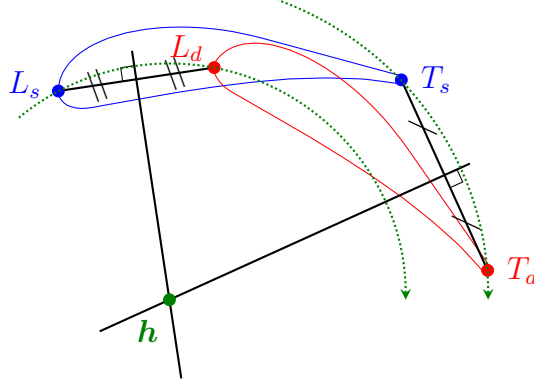


Figure D.4: A single drop hinge geometrical construction from stowed (blue) and deployed (red) configuration.

Therefore, the final goal of the multifidelity multiobjective high-lift devices optimisation is its application to a kinematic-aerodynamic optimisation. For such task, the methodology presented in figure D.5 is considered. The multifidelity suboptimisation is performed on a single design variable problem: the flap and slat deflection optimisation. Both optimisations are independent and can be conducted in parallel. Another optimiser is required to optimise the hinge point position according to the suboptimiser output. Ultimately, the method gives a Pareto front of maximum lift versus lift-over-drag, where each point relates to different hinge point and a single point relates to two different deflections.

Figure D.6 shows preliminary results performed on a limited surrogate trained on the high-fidelity CFD model. A Pareto front is obtained and the point with best $c_{l_{\max}} + L/D$ is represented on the right with, in blue the deflection angle providing the best lift-over-drag and in red the configuration providing the best maximum lift. Unfortunately the slat position is not physical due to the surrogate that provides information in this region only by extrapolation and cannot be relied on.

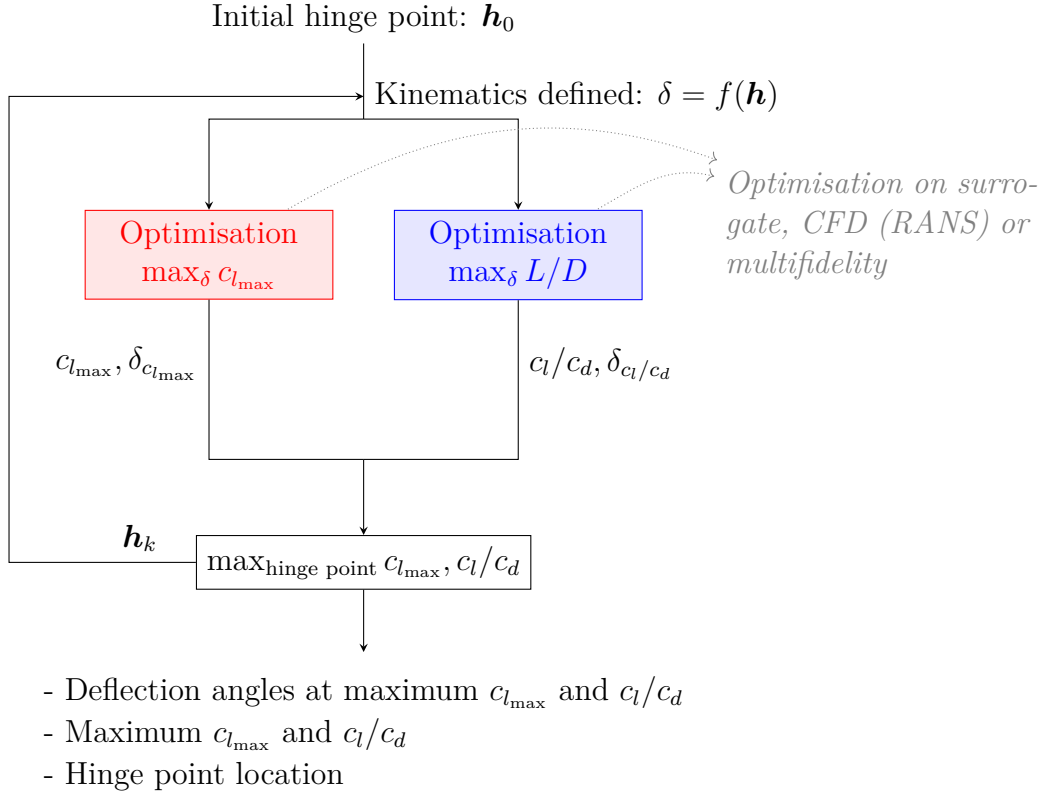


Figure D.5: Proposed aerodynamics multifidelity multiobjective optimisation framework for single-hinge including kinematics.

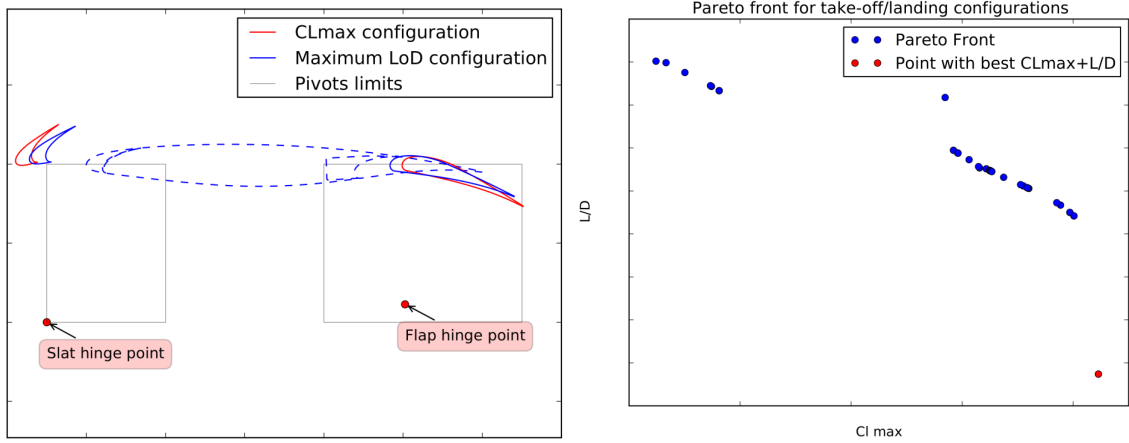


Figure D.6: Multiobjective kinematic optimisation initial results: positioning for the point highlighted on the Pareto front and hinge point position on the left and Pareto front on the right.