

**Reduced-Basis Methods Applied to Locally Non-affine and
Locally Non-linear Partial Differential Equations**

by

Yuri Olegovich Solodukhov

Submitted to the Mechanical Engineering Department
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

© Massachusetts Institute of Technology 2005. All rights reserved.

Author

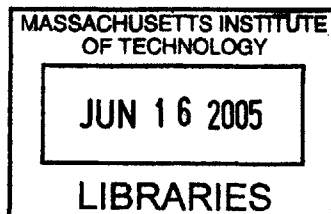
Mechanical Engineering Department
March 2, 2005

Certified by

Anthony T. Patera
Professor
Thesis Supervisor

Accepted by

Lallit Anand
Chairman, Department Committee on Graduate Students



BARKER

10/10/10

10/10/10

Reduced-Basis Methods Applied to Locally Non-affine and Locally Non-linear Partial Differential Equations

by

Yuri Olegovich Solodukhov

Submitted to the Mechanical Engineering Department
on March 2, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In modern engineering and scientific applications there is a huge demand for solutions of parameter-based partial differential equations and associated outputs of interest expressed as functionals of these solutions. Areas that require solving partial differential equations include – but are not restricted to – heat transfer, elasticity, and fluid dynamics. Since in most cases it is not feasible to obtain an analytic solution, many numerical approaches to obtain approximate numerical solutions – such as finite elements, finite differences, finite volumes – have been developed. For applications like optimization, design, and inverse problems, where it is crucial to evaluate the field solution/output repeatedly, it might be overly computationally expensive to apply conventional numerical methods.

To address this issue we present and compare two new reduced basis techniques for the rapid and reliable prediction of linear functional outputs of linear elliptic partial differential equations with locally non-affine parameter dependence: the partition of unity method (PUM) and the minimax coefficient approximation method (MCAM). We also describe the minimax coefficient approximation method (MCAM) in application to locally non-linear elliptic partial differential equations. The essential components for both the PUM and the MCAM are (i) (provably) rapidly convergent global reduced basis approximations – Galerkin projection onto a low-dimensional space spanned by the solutions of the governing partial differential equation at N selected points in the parameter space; (ii) a posteriori error estimation – relaxations of the error-residual equation that provide inexpensive yet sharp and rigorous bounds for the error in the outputs of interest; and (iii) off-line/on-line computational procedures – methods which decouple the generation and projection stages of the approximation process. The operation count for the on-line stage – in which, given a new parameter value, we calculate the output of interest and associated error bound – depends only on N (typically very small), the affine parametric complexity of the problem and the number of points in the region where the non-affine/non-linear dependence is observed. The ratio of the error bound to the real error (which we call effectivity or the sharpness of our error estimate) typically does not exceed 100.

The partition of unity approach relies on domain decomposition with respect to the separation of the affine part from the non-affine part and estimation of contributions to the error bound from these two parts. The minimax coefficient approximation approach is based on approximating the non-affine/non-linear dependence with an affine-like approximation and the subsequent treatment of the problem based on the ideas previously developed for affine problems.

As a test for these new methods we consider several model problems involving steady heat transfer. Numerical results are provided with respect to the accuracy and computational savings provided by the described reduced basis methods.

Thesis Supervisor: Anthony T. Patera
Title: Professor

Acknowledgments

I would like to thank my thesis advisor, Professor Anthony Patera for providing me with guidance throughout my Ph.D. track at MIT. He really taught me a lot and showed me many alternative ways to address different problems in mechanical engineering and applied mathematics. I would also like to thank Professor Bora Mikic and Professor Jaume Peraire for finding time to serve on my thesis committee and giving me lots of useful feedback.

I would like to express my gratitude to the whole Prof. Patera's research group, especially Dimitrios Rovas (now Professor). My special thanks to Prof. Patera's administrative assistant, Debra Blanchard for helping me with some technical aspects of this thesis providing good company and creating nice atmosphere within in the group.

I am very grateful to my mother, Elena Avgustovna Solodukhova, and father, Oleg Orestovich Solodukhov, for providing me with an opportunity to be admitted to M.I.T. and to successfully obtain a Ph.D. degree.

I would also like to mention all friends of mine, especially Nikolay Ivanter and Martin Cohn who helped to sometimes get distracted from my research work, so that I could later return back to it refreshed and energetic.

Contents

1	Introduction and Motivation	21
1.1	Motivation: 2D Fin with Non-affine Geometry Variation and Radiation Boundary Conditions	22
1.1.1	Problem Statement	23
	Inputs	23
	Governing Partial Differential Equations and Field Solution	24
	Outputs	24
1.2	Classical Reduced Basis Method	25
1.2.1	Reduced Basis Approximation	27
1.2.2	A Posteriori Error Estimation	28
1.2.3	Off-line/On-line Procedure: Fallacies and Limitations with respect to Non-affine and Non-linear Components	30
	Reduced Basis Approximation	30
	Error Bound	32
1.3	Thesis Goals, Scope, Contributions, and Limitations	33
1.3.1	Goals	34
1.3.2	Classes of Problems Considered	34
	Affine Coercive Problems	34
	Locally Non-affine Coercive Problems	35
	Locally Non-linear Problems	35
1.3.3	Limitations	36
	Generality of Problems Classes	36
	Operation Count for Locally Non-affine and Non-linear Problems	37

1.4	Approaches	37
1.4.1	"Classical" Reduced Basis Method	37
1.4.2	Partition of Unity Method	38
1.4.3	Minimax Coefficient Approximation Method	38
1.4.4	Key Ingredients from Prior Work	39
	Fast and Accurate Reduced Basis Approximation	39
	Rigorous and Reliable Error Estimation	39
	Framework	39
1.5	Thesis Outline	40
2	Affine Coercive Problems	41
2.1	General Abstract Problem Statement	42
2.2	Dimension Reduction Idea	44
2.3	"Classical" Reduced Basis Method	45
2.3.1	Reduced Basis Approximation	45
2.3.2	A Priori Convergence Arguments	46
2.3.3	A Posteriori Error Estimation	47
	A Posteriori Error Bound for $\ e_N(\mu)\ _Y$ and Compliant Output	47
	A Priori Effectivity Analysis	49
	A Posteriori Error Bound for Linear Outputs	50
2.3.4	Sample Selection Algorithm	50
2.3.5	Off-line/On-line Procedure	52
	Reduced Basis Approximation	52
	Error Bound	53
	Numerical Algorithm	54
	Off-line	54
	On-line	54
2.4	Model Problem: 2D Thermal Fin	54
2.4.1	Problem Statement	54
2.4.2	Governing Partial Differential Equations	56
	Strong form	56
	Weak form	56

2.4.3	Reduction to Abstract Form	57
	Affine Mappings	57
	Abstract Form	58
2.4.4	Numerical Results	60
	Convergence	60
	Effectivity	62
	Computational Costs	63
	Conclusions	65
3	Locally Non-affine Coercive Problems: Partition of Unity Method	67
3.1	General Abstract Problem Statement	67
3.1.1	Failure of Classical Reduced Basis Method	69
3.2	Partition of Unity Method	72
3.2.1	Partition of Unity	72
3.2.2	Reduced Basis Approximation	73
3.2.3	A Posteriori Error Estimation	74
	A Posteriori Error Bound for $\ e_N(\mu)\ _Y$ and Compliant Output	75
	A Priori Effectivity Analysis	79
	A Priori Effectivity Analysis for 1D Heat Conduction Problem	81
3.2.4	Off-line/On-line Procedure	87
	Reduced Basis Approximation	87
	Error Bound	89
	Numerical Algorithm	91
	Off-line	91
	On-line	91
3.3	Model Problem: 2D Heat Conduction Example	91
3.3.1	Governing Partial Differential Equations	93
	Strong Form	93
	Weak Form	94
3.3.2	Reduction to Abstract Form	94
3.3.3	Numerical Results	96
	Effect of Non-affine Parameter τ	96

Convergence	98
Effectivity	99
Efficient Partition of Unity Selection	100
Computational Costs	101
Conclusions	103
3.4 Application of Partition of Unity Method to Affine Coercive Problems	104
3.4.1 General Abstract Problem Statement	104
3.4.2 Reduced Basis Approximation	104
3.4.3 A Posteriori Error Estimation	106
Numerical Algorithm	109
Off-line	109
On-line	109
3.4.4 Numerical Results in Comparison to Classical Reduced Basis Method	110
Convergence	110
Effectivities	110
Computational Costs	111
Conclusions	114
4 Locally Non-affine Problems: Minimax Coefficient Approximation Method	117
4.1 Minimax Coefficient Approximation Method	117
4.1.1 Parametric Functional Dependency	117
4.1.2 S_M^g Sample Selection	119
4.1.3 "Magic Points" Selection	121
4.1.4 Reduced Basis Approximation	123
4.1.5 A Posteriori Error Estimation	124
4.1.6 Off-line/Online Procedure	128
Reduced Basis Approximation	128
Error Bound	128
Numerical Algorithm	130
Off-line	130
On-line	130
4.2 Model Problem: 2D Heat Conduction Example	131

4.2.1	Weak Form	131
4.2.2	Numerical Results	132
	Approximation of Non-affine Functional Dependency	132
	Convergence	134
	Effectivity	136
	Error Bound Decomposition	137
	Further $M - N$ analysis	140
	Computational Costs	141
	Conclusions	142
4.2.3	Comparison of Partition of Unity and Minimax Coefficient Approximation	
	Methods	143
	Convergence and Effectivity	143
	Computational Costs	144
	Conclusions	145
5	Locally Non-linear Problems: Minimax Coefficient Approximation Method	149
5.1	General Abstract Problem Statement	149
5.1.1	Newton Method	151
5.1.2	Limitations of Classical Reduced Basis Method	152
	Reduced Basis Approximation	152
	Error Estimation	154
5.2	Minimax Coefficient Approximation Method	155
5.2.1	S_M^g Sample Selection	155
5.2.2	"Magic Points" Selection	156
5.3	Application of Minimax Coefficient Approximation Method to Locally Non-linear	
	Problems	158
5.3.1	Reduced Basis Approximation	158
5.3.2	A Posteriori Error Estimation Based on Positivity of Non-linear Contribution	159
5.3.3	Off-line/On-line Procedure	163
	Reduced Basis Approximation	163
	Error Bound	164
	Numerical Algorithm	165

Off-line	165
On-line	166
5.4 Model Problem: 2D Heat Conduction Example with Radiation Boundary Conditions	166
5.4.1 Governing Partial Differential Equations	167
Strong Form	167
Weak Form	168
5.4.2 Numerical Results	170
Effect of Radiation	170
Approximation of Non-linearity	171
Convergence	172
Effectivity	175
Error Bound Decomposition	176
Further $M - N$ Analysis	180
Computational Costs	182
Conclusions	183
6 Summary and Future Work	185
6.1 Summary	185
6.2 Future Work	186
A Proof of Formula for $\mathcal{T}(N, \kappa)$	189

List of Figures

1-1	2D Locally non-affine and locally non-linear model problem: a thermal fin with radiation boundary conditions and non-affine parametric dependence in conductivity.	22
2-1	Critical observation illustration: a low dimensional manifold.	45
2-2	2D affine model problem: a thermal fin with affine shape variation.	55
2-3	2D affine model problem: the mapping $\mathcal{G}(\mu) : \tilde{\Omega}(\mu) \rightarrow \Omega$ between the varying shape domain $\tilde{\Omega}$ (left) and fixed reference domain Ω (right) and their respective boundaries.	58
2-4	2D affine model problem: the field solution $u(\mu)$ for (a) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10\}$, (b) $\mu = \{3.40, 3.40, 3.40, 3.40, 0.34, 2.33, 0.23, 2.33, 0.23, 2.33, 0.23, 2.33, 0.23\}$, (c) $\mu = \{6.70, 6.70, 6.70, 6.70, 0.67, 2.67, 0.37, 2.67, 0.37, 2.67, 0.37, 2.67, 0.37\}$, (d) $\mu = \{10.00, 10.00, 10.00, 10.00, 1.00, 3.00, 0.50, 3.00, 0.50, 3.00, 0.50, 3.00, 0.50\}$.	61
2-5	Classical reduced basis method, 2D affine model problem: $\ e_N\ _Y$ and Δ_N as functions of N .	62
2-6	Classical reduced basis method, 2D affine model problem: $a(e_N, e_N)$ and $\tilde{\Delta}_N$ as functions of N .	62
2-7	Classical reduced basis method, 2D affine model problem: effectivities η_N and $\tilde{\eta}_N$ as functions of N .	64
3-1	Partition of unity method: relation between T_N , the "truth" discretization, and T_S , the coarse discretization.	73
3-2	Partition of unity method, 1D model problem: domain configuration.	81
3-3	1D model problem: $\Phi_i(x)$, $i = 1, \dots, S$, partition of unity.	82
3-4	2D locally non-affine model problem: a thermal fin with affine shape and non-affine conductivity variation.	93

3-5	2D locally non-affine model problem: the mapping $\mathcal{G}(\mu) : \tilde{\Omega}(\mu) \rightarrow \Omega$ between the varying shape domain $\tilde{\Omega}$ (left) and fixed reference domain Ω (right) and their respective boundaries.	95
3-6	2D affine model problem: the field solution $u(\mu)$ for (a) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, -0.7571\}$, (b) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, -0.2524\}$, (c) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, 0.2524\}$, (d) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, 0.7571\}$. . .	96
3-7	2D locally non-affine model problem: variation of u_{root} as a function of τ , $k_i = 0.10$, $i = 1, \dots, 4$, $\text{Bi} = 0.01$, $\beta_1 = 3.00$, $\alpha_1 = 0.50$, $\beta_i = 2.00$, $\alpha_i = 0.10$, $i = 2, \dots, 4$	97
3-8	2D locally non-affine model problem: relation between T_N , the "truth" discretization, and T_S , the coarse discretization.	98
3-9	Partition of unity method, 2D locally non-affine model problem: $\ e_N\ _Y$ and Δ_N as functions of N	99
3-10	Partition of unity method, 2D locally non-affine model problem: $a(e_N, e_N)$ and $\tilde{\Delta}_N$ as functions of N	99
3-11	Partition of unity method, 2D locally non-affine model problem: effectivities $\tilde{\eta}_N$ and η_N as functions of N	101
3-12	Partition of unity method: the idea behind efficient partition of unity selection.	102
3-13	Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: $\ e_N\ _Y$ and Δ_N as functions of N	110
3-14	Comparison of classical reduced basis (blue) and partition of unity (red) methods, 2D affine model problem: effectivity η_N as a function of N	111
3-15	Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: the dominating operation count for construction of error bound.	115
4-1	Minimax coefficient approximation method, 2D locally non-affine problem: functional parametric dependence approximation.	133
4-2	Minimax coefficient approximation method, 2D locally non-affine problem: $\ e_{NM}\ _Y$ as a function of M while N is fixed.	135

4-3	Minimax coefficient approximation method, 2D locally non-affine problem: $\ e_{NM}\ _Y$ as a function of N while M is fixed.	135
4-4	Minimax coefficient approximation method, 2D locally non-affine problem: $\ e_{NM}\ _Y$ and Δ_{NM} as functions of N while M is fixed.	136
4-5	Minimax coefficient approximation method, 2D locally non-affine problem: effectivity η_{NM} as a function of M while N is fixed.	138
4-6	Minimax coefficient approximation method, 2D locally non-affine problem: effectivity η_{NM} as a function of N while M is fixed.	138
4-7	Minimax coefficient approximation method, 2D locally non-affine problem: $\tilde{\Delta}_{NM,1}$ and $ R_1(e_{NM}) $ as functions of N while M is fixed.	138
4-8	Minimax coefficient approximation method, 2D locally non-affine problem: $\tilde{\Delta}_{NM,1}Y$ and $ R_1(e_{NM}) $ as functions of M while N is fixed.	138
4-9	Minimax coefficient approximation method, 2D locally non-affine problem: $\tilde{\Delta}_{NM,2}$ and $ R_2(e_{NM}) $ as functions of N while M is fixed.	139
4-10	Minimax coefficient approximation method, 2D locally non-affine problem: $\tilde{\Delta}_{NM,2}$ and $ R_2(e_{NM}) $ as functions of M while N is fixed.	139
4-11	Minimax coefficient approximation method, 2D locally non-affine problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of M while N is fixed.	140
4-12	Minimax coefficient approximation method, 2D locally non-affine problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of N while M is fixed.	140
4-13	Minimax coefficient approximation method, 2D locally non-affine problem: M^* and M^{**} as functions of N	141
4-14	Minimax coefficient approximation method, 2D locally non-affine problem: $\ e_{NM}\ _Y$ and Δ_{NM} as functions of N and $M^*(N)$, $M^{**}(N)$	142
4-15	Minimax coefficient approximation method, 2D locally non-affine problem: effectivity η_{NM} as a function of N and $M^*(N)$, $M^{**}(N)$	143
4-16	Minimax coefficient approximation method, 2D locally non-affine problem: on-line computational times as functions of M while N is fixed.	144
4-17	Minimax coefficient approximation method, 2D locally non-affine problem: t_{tot} , the total on-line computational times as a function of N and $M^*(N)$, $M^{**}(N)$	145

4-18	Comparison of the partition of unity (left) and the minimax coefficient approximation (right) methods: $\ e_N\ _Y$ and Δ_{NS} as functions of N ; $\ e_{NM}\ _Y$ and Δ_{NM} as functions of N and $M^{**}(N)$	147
4-19	Comparison of the partition of unity and the minimax coefficient approximation methods, effectivities η_N as function of N and η_{NM} as a function of N and $M^{**}(N)$	148
5-1	2D locally non-linear model problem: a thermal fin with convective and radiation boundary conditions.	167
5-2	2D locally non-linear model problem: the field solution $u(\mu)$ for (a) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 0.01\}$, (b) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 0.25\}$, (c) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 0.50\}$, (d) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 1.00\}$	170
5-3	2D locally non-linear model problem: variation of u_{root} as a function of σ , $k_1 = 0.50$, $k_2 = 0.50$, $k_3 = 0.50$, $k_4 = 0.50$, $\text{Bi} = 0.50$	171
5-4	Minimax coefficient approximation method, 2D locally non-linear problem: non-linear contribution approximation.	172
5-5	Minimax coefficient approximation method, 2D locally non-linear problem: $\ e_{NM}\ _Y$ as a function of M while N is fixed.	174
5-6	Minimax coefficient approximation method, 2D locally non-linear problem: $\ e_{NM}\ _Y$ as a function of N while M is fixed.	174
5-7	Minimax coefficient approximation method, 2D locally non-linear problem: $\ e_{NM}\ _Y$ and Δ_{NM} as functions of N while M is fixed.	174
5-8	Minimax coefficient approximation method, 2D locally non-linear problem: effectivity η_{NM} as a function of M while N is fixed.	176
5-9	Minimax coefficient approximation method, 2D locally non-linear problem: effectivity η_{NM} as a function of N while M is fixed.	176
5-10	Minimax coefficient approximation method, 2D locally non-linear problem: $\tilde{\Delta}_{NM,1}$ and $ R_1(e_{NM}) $ as functions of N while M is fixed.	177
5-11	Minimax coefficient approximation method, 2D locally non-linear problem: $\tilde{\Delta}_{NM,1}$ and $ R_1(e_{NM}) $ as functions of M while N is fixed.	177

5-12	Minimax coefficient approximation method, 2D locally non-linear problem: $\tilde{\Delta}_{NM,2}$ and $ R_2(e_{NM}) $ as functions of N while M is fixed.	177
5-13	Minimax coefficient approximation method, 2D locally non-linear problem: $\tilde{\Delta}_{NM,2}$ and $ R_2(e_{NM}) $ as functions of M while N is fixed.	177
5-14	Minimax coefficient approximation method, 2D locally non-linear problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of M while N is fixed.	178
5-15	Minimax coefficient approximation method, 2D locally non-linear problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of N while M is fixed.	178
5-16	Minimax coefficient approximation method, 2D locally non-linear problem: study of contribution of non-linear term: $\frac{\Xi_{NM}^{NL}}{\Xi_{NM}^A}$ and as a function of M while N is fixed.	179
5-17	Minimax coefficient approximation method, 2D locally non-linear problem: study of contribution of non-linear term: $\frac{\Xi_{NM}^{NL}}{\Xi_{NM}^A}$ and as a function of N while M is fixed.	179
5-18	Minimax coefficient approximation method, 2D locally non-linear problem: M^* and M^{**} as functions of N	180
5-19	Minimax coefficient approximation method, 2D locally non-linear problem: $\ e_{NM}\ _Y$ and Δ_{NM} as functions of N and $M^*(N)$, $M^{**}(N)$	181
5-20	Minimax coefficient approximation method, 2D locally non-linear problem: effectiveness η_{NM} as a function of N , and $M^*(N)$, $M^{**}(N)$	182
5-21	Minimax coefficient approximation method, 2D locally non-linear problem: on-line computational times as functions of N while M is fixed.	183
5-22	Minimax coefficient approximation method, 2D locally non-linear problem: on-line computational times as functions of M while N is fixed.	183

List of Tables

2.1	2D affine model problem: $u_{\text{root}}(\mu)$ variation.	60
2.2	Classical reduced basis method, 2D affine model problem: relative convergence of $\ e_N\ $, Δ_N and $a(e_N, e_N)$, $\tilde{\Delta}_N$ with respect to N	63
2.3	Classical reduced basis method, 2D affine model problem: on-line computational times.	65
3.1	Partition of unity method, 1D model problem: a priori and a posteriori effectivity analysis.	88
3.2	Partition of unity method, 2D locally non-affine model problem: relative convergence of $\ e_N\ _Y$, Δ_N and $a(e, e)$, $\tilde{\Delta}_N$ with respect to N	100
3.3	Partition of unity method, 2D locally non-affine model problem: on-line computational times.	103
3.4	Partition of unity method, 2D affine model problem: on-line computational times.	112
3.5	Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: on-line computational times as functions of $\ e_N\ _Y$	113
3.6	Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: on-line computational times as functions of the error bound.	114
4.1	Minimax coefficient approximation method, 2D locally non-affine problem: $\text{cond}(B^M)$ and Λ_M as functions of M	134
4.2	Minimax coefficient approximation method, 2D locally non-affine problem: relative convergence of $\ e_{NM}\ _Y$, Δ_{NM} as functions of N , $M = 7$	137
4.3	Comparison of partition of unity method and minimax coefficient approximation method: on-line computational times as functions of error bound.	146

4.4	Comparison of the partition of unity and the minimax coefficient approximation methods: on-line computational times as functions of Δ_{NS} and Δ_{NM}	146
5.1	Minimax coefficient approximation method, 2D locally non-linear problem: $\text{cond}(B^M)$ and Λ_M as functions of M	173
5.2	Minimax coefficient approximation method, 2D locally non-linear problem: relative convergence of $\ e_{NM}\ _Y$, Δ_{NM} as functions of N , $M = 30$	175
5.3	Minimax coefficient approximation method, 2D locally non-linear problem: t_{tot} , the total on line computational time as functions of N and $M^*(N)$, $M^{**}(N)$	184

Chapter 1

Introduction and Motivation

In modern engineering and scientific applications there is a huge demand for solutions of parameter-based partial differential equations and associated outputs of interest as functionals of these solutions. Areas that require solving partial differential equations include – but are not restricted to – heat transfer, elasticity, and fluid dynamics. Since in most cases it is not feasible to obtain an analytical solution, many numerical approaches to obtain approximate numerical solutions – such as finite elements, finite differences, finite volumes – have been developed. For applications like optimization, design, and inverse problems, where it is crucial to evaluate the field solution/output repeatedly, it might be overly computationally expensive to apply these conventional numerical methods.

To address this issue in [42, 49, 52] a reduced basis methodology was presented for the *rapid* and *reliable* prediction of linear-functional outputs of elliptic partial differential equations with affine parameter dependence. In this thesis we extend this methodology to address more general classes of problems. The new classes we consider include locally non-affine and locally non-linear problems. For the locally affine problems the affine parameter dependence is required almost everywhere in the domain except in a small region where we allow general parameter dependence. For the locally non-linear problems we allow the non-linearity in the partial differential equations in a small region, whereas in the main part of the domain the equations stay linear. Locally non-affine and non-linear problems are widely studied in science and engineering. To further motivate the introduction of these new methods we are going to consider a concrete model problem.

1.1 Motivation: 2D Fin with Non-affine Geometry Variation and Radiation Boundary Conditions

In this Section we present a description and a mathematical formulation of a particular model problem to which we intend to apply our reduced basis methodology. In general, we are going to apply our reduced basis methods to a class of problems which satisfies certain requirements (we this issue discuss later in this thesis), however it is more convenient to start by presenting a relatively simple model problem to motivate further discussion.

As a model problem we consider a special configuration of a two-dimensional thermal fin. This model problem is an extension to the model problem considered in [49]. The two-dimensional fin consists of a vertical central “post” and four horizontal “sub-fins” on each side of the post; the fin conducts heat from a prescribed uniform flux “source” at the root, Γ_{root} , through the large-surface-area sub-fins to surrounding flowing air by convection and also through the part of the boundary which we denote Ω_{nl} by the way of radiation; in Figure 1-1 it is depicted by a thick solid line.

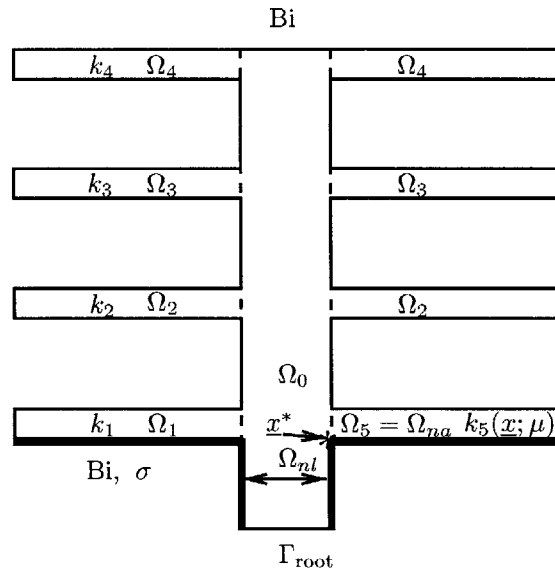


Figure 1-1: 2D Locally non-affine and locally non-linear model problem: a thermal fin with radiation boundary conditions and non-affine parametric dependence in conductivity.

1.1.1 Problem Statement

Inputs

The fin is characterized by a seven component parameter vector or “input”,

$$\mu = \left(\underbrace{k_1}_{\mu_1}, \underbrace{k_2}_{\mu_2}, \underbrace{k_3}_{\mu_3}, \underbrace{k_4}_{\mu_4}, \underbrace{\text{Bi}}_{\mu_5}, \underbrace{\sigma}_{\mu_6}, \underbrace{\tau}_{\mu_7} \right),$$

where k_i , $i = 1, \dots, 4$ is the thermal conductivity of the i^{th} subfin the area of which is denoted as Ω_i (note that subfins Ω_2 , Ω_3 , and Ω_4 are comprised of two parts located on both sides of the post, whereas Ω_1 and $\Omega_5 = \Omega_{na}$ are formed by a single part each as shown in Figure 1-1) ; k_i , $i = 1, \dots, 5$ is normalized relative to the post conductivity $k_0 \equiv 1$; Bi is the Biot number, a non-dimensional heat transfer coefficient reflecting convective transport to the air at the fin surfaces, σ is the non-dimensionalized emissivity of the radiative surface Ω_{nl} , the non-affine component is introduced into the problem by $k_5(\underline{x}, \mu)$, the conductivity of subfin $\Omega_5 = \Omega_{na}$ which is no longer a constant as k_i , $i = 1, \dots, 4$, but a function of $\mu_7 = \tau$ and $\underline{x} \in \Omega_5$:

$$k_5(\underline{x}; \mu) = \exp(\tau \sqrt{(x_1 - x_1^*)^2 + (x_1 - x_2^*)^2}), \quad (1.1)$$

where \underline{x}^* is the corner point of Ω_5 as depicted in Figure 1-1; μ may take on any value in a specified design parameter domain $\mathcal{D} \subset \mathbb{R}^7$. The total height of the fin (also measured relative to the post width) is fixed at $H = 4$.

For our parameter domain we choose $\mathcal{D} = [0.1, 10.0]^4 \times [0.01, 1.0]^2 \times [-1.1424, 1.1424]$, that is, $0.1 \leq k_i \leq 10.0$, $i = 1, \dots, 4$ for the conductivities, $0.01 \leq \text{Bi}, \sigma \leq 1.0$ for the Biot number and σ , the emissivity of the gray surface Ω_{nl} , and $-1.1424 \leq \tau \leq 1.1424$ for the non-affine parameter τ .

We consider two quantities of interests. The first quantity is the temperature distribution $u(\underline{x}; \mu)$ itself which we will often refer to as the “field solution”. We will often omit the dependence of $u(\underline{x}; \mu)$ on the space coordinate \underline{x} and refer to the former just as $u(\mu)$. The second one – which we also refer to as an output of interest since it is a scalar function of the solution $u(\underline{x}; \mu)$ – is u_{root} , the average temperature at the root of the fin normalized by the prescribed heat flux into the fin root. This particular output relates directly to the cooling efficiency of the fin — lower values of u_{root} imply better performance.

Governing Partial Differential Equations and Field Solution

The temperature distribution $u(\mu)$, is obtained by solution of the following elliptic partial differential equations:

$$-k_i \nabla^2 u_i(\mu) = 0 \text{ in } \Omega_i, \quad i = 0, \dots, 4, \quad (1.2)$$

$$-\nabla(k_5(\underline{x}; \mu) \nabla u_5(\mu)) = 0 \text{ in } \Omega_5 = \Omega_{na}, \quad (1.3)$$

where ∇^2 is the Laplacian operator, and $u_i(\mu) \equiv u(\mu)|_{\Omega_i}$ refers to the restriction of $u(\mu)$ to Ω_i . Here Ω_i is the region of the fin with conductivity k_i , $i = 0, \dots, 5$: Ω_0 is thus the central post, and Ω_i , $i = 1, \dots, 5$, corresponds to the five sub-fins. We must also ensure continuity of temperature and heat flux at the conductivity-discontinuity interfaces $\Gamma_i \equiv \partial\Omega_0 \cap \partial\Omega_i$, $i = 1, \dots, 5$, where $\partial\Omega_i$ denotes the inner boundary of Ω_i :

$$\left. \begin{aligned} u_0(\mu) &= u_i(\mu) \\ -(\nabla u_0(\mu) \cdot \hat{n}_i) &= -k_i(\nabla u_i(\mu) \cdot \hat{n}_i) \end{aligned} \right\} \text{ on } \Gamma_i, \quad i = 1, \dots, 5;$$

here \hat{n}_i is the outward normal on $\partial\Omega_i$. We introduce a Neumann boundary condition on the fin root:

$$-(\nabla u_0(\mu) \cdot \hat{n}_o) = -1 \text{ on } \Gamma_{\text{root}},$$

which models the heat source; and the Robin boundary condition:

$$-k_i(\nabla u_i(\mu) \cdot \hat{n}_i) = \text{Bi } u_i(\mu) \text{ on } \Gamma_{\text{ext } i} \setminus (\Omega_{nl} \cup \Gamma_{\text{root}}), \quad i = 0, \dots, 4,$$

which models the convective heat losses. Here $\Gamma_{\text{ext } i}$ is that part of the boundary of Ω_i exposed to the fluid. Finally we introduce the mixed radiation-convection non-linear boundary condition on Ω_{nl} :

$$-k_i(\nabla u_i(\mu) \cdot \hat{n}_i) = \text{Bi } u_i(\mu) + \sigma (u_i(\mu))^4 \text{ on } \Omega_{nl \ i}, \quad i = 0, 1, 5. \quad (1.4)$$

Note that we assume the ambient temperature to be zero.

Outputs

For every choice of the design parameter-vector μ — which is determined by $k_i, i = 1, \dots, 4$ Bi, σ and τ — the solution of the equations above yields the temperature distribution $u(\mu)$, or the field

solution. The average temperature at the root, u_{root} , can then be obtained from $s(\mu) \equiv u_{\text{root}} = f(u(\mu))$, where

$$f(v) = \int_{\Gamma_{\text{root}}} v \quad (1.5)$$

(recall Γ_{root} is of length unity).

The thermal fin problem exercises many aspects of our methods as there is a relatively large number of input parameters that appear in the problem equations and boundary conditions. The conductivity function (1.1) introduces the locally non-affine component into the problem, whereas the non-linear radiation boundary condition brings in the locally non-linear contribution. Note that the dimensional analysis plays an important role in the modelling process since it allows us to reduce the number of parameters that define the problem without any loss of rigor or generality.

1.2 Classical Reduced Basis Method

Typically the problems similar to the one introduced in Section 1.1 are solved numerically using the finite element method. In order to apply the finite element method we turn to the weak form of the partial differential equations stated in 1.1.1. The details of this procedure for particular model problems are presented in Sections 2.4, 3.3, and 5.4.

As we mentioned earlier, our model problem is non-linear, therefore in order to obtain the field solution $u(\mu)$ we employ Newton method. We provide more detail on Newton method in Chapter 5. The cost per Newton iteration is a function of the number of nodes in finite element discretization of Ω — typically a very large number which we denote \mathcal{N} . Since \mathcal{N} is so big, it often becomes overly computationally expensive to use the conventional finite element method especially in the context of repeated evaluation of the field solution and the respective outputs of interest. This is why we are going to introduce the reduced basis method which as we hope will provide a way to evaluate approximations to the field solution and the outputs of interest in a more numerically efficient way.

The weak form of the equations presented in 1.1.1 is given as follows. Find $u(\mu) \in Y = H^1(\Omega)$ — which we later refer to as the weak solution — satisfying

$$a(u(\mu), v; \mu) = a^A(u(\mu), v; \mu) + a^{NA}(u(\mu), v; \mu) + a^{NL}(u(\mu), v; \mu) = f(v), \quad \forall v \in Y, \quad (1.6)$$

where

$$a^A(w, v; \mu) = \sum_{i=0}^4 k_i \int_{\Omega_i} \nabla w \nabla v + \text{Bi} \int_{\Gamma_{\text{ext}}} wv \quad (1.7)$$

represents the affine contribution;

$$a^{NA}(w, v; \mu) = \int_{\Omega_5 = \Omega_{na}} k_5(\underline{x}; \mu) \nabla w \nabla v \quad (1.8)$$

represents the locally non-affine contribution resulting from the functional dependence (1.1). We note that $a^L(w, v; \mu)$ – defined as

$$a^L(w, v; \mu) = a^A(w, v; \mu) + a^{NA}(w, v; \mu), \quad \forall w, v \in Y, \quad \forall \mu \in \mathcal{D}, \quad (1.9)$$

is coercive, i.e. there exists $\alpha(\mu) > 0$ such that $\forall \mu \in \mathcal{D}$

$$\alpha(\mu)(\|v\|_Y)^2 \leq a^L(v, v; \mu), \quad \forall v \in Y. \quad (1.10)$$

In this context the term "locally" means that the area of Ω_{na} is small compared to the area of Ω which from the finite element standpoint results in the fact that n_{na} , the number of nodes in the finite element discretization of Ω_{na} is small compared to \mathcal{N} . We give a rigorous definition of what we mean by locally non-affine problems later in this Chapter and also in Section 3.1. We note that since $k_5(\underline{x}; \mu) > 0, \forall \underline{x} \in \Omega_{na}, \forall \mu \in \mathcal{D}$

$$a^{NA}(v, v; \mu) \geq 0, \quad \forall v \in Y. \quad (1.11)$$

Next,

$$a^{NL}(u(\mu), v; \mu) = \sigma \int_{\Gamma_{\text{top}}} (u(\mu))^4 v \quad (1.12)$$

represents the locally non-linear contribution due to the radiative heat transfer. The term locally in the context of non-linear problems means that Ω_{nl} is of lower dimension than Ω or that the area of Ω_{nl} is small compared to the area of Ω ; hence n_{nl} , the number of nodes in the finite element discretization of Ω_{nl} is small compared to \mathcal{N} . Again, we provide the rigorous mathematical definition of locally non-linear problems in Section 1.3.2 and also in Section 5.1.

Finally,

$$f(v) = \int_{\Gamma_{\text{root}}} v \quad (1.13)$$

represents the heat flux through the root of the fin.

1.2.1 Reduced Basis Approximation

As mentioned earlier, the operation count for solution of (1.6) using the finite element method is a function of a very large number \mathcal{N} which makes application of the finite element method quite expensive, especially in the context of repeated evaluation of the solution $u(\mu)$. To address this issue we present the outline for the reduced basis method based on the ideas developed in earlier works [56, 54, 49]. Instead of the finite element solution $u(\mu)$ we are going to construct a reduced basis approximation $u_N(\mu)$ as linear combination of N previously precomputed solutions of the same problem. We expect the operation count to obtain $u_N(\mu)$ to be a function N , presumably a very small number compared to \mathcal{N} . This is why we believe that it would be much cheaper to calculate $u_N(\mu)$ compared to $u(\mu)$.

We start by selecting the sample set S_N in parameter domain \mathcal{D}

$$S_N = \{\mu_1, \dots, \mu_N\}, \quad (1.14)$$

where $\mu_i \in \mathcal{D}, i = 1, \dots, N$. We then define our reduced-basis approximation space W_N as

$$W_N = \text{span} \{u(\mu_i) = \zeta_i, i = 1, \dots, N\}. \quad (1.15)$$

In order to obtain a reduced basis approximation for $u(\mu)$ we simply replace the function space Y in the Galerkin projection (1.6) by the function space W_N – we now look for $u_N(\mu) \in W_N$ such that

$$a^A(u_N(\mu), v; \mu) + a^{NA}(u_N(\mu), v; \mu) + a^{NL}(u_N(\mu), v; \mu) = f(v), \quad \forall v \in W_N. \quad (1.16)$$

Thus, $u_N(\mu)$ serves as an approximation to the field solution $u(\mu)$. Since $u_N(\mu) \in W_N$ it can be expressed as

$$u_N(\mu) = \sum_{i=1}^N u_{N_i}(\mu) \zeta_i. \quad (1.17)$$

The approximation to the output $s(\mu)$ is given by

$$s_N(\mu) = f(u_N(\mu)). \quad (1.18)$$

Since the dimension of W_N is N — which we hope to be quite small compared to \mathcal{N} , the dimension of the finite element space — we expect significant computational savings by turning to this reduced basis approximation.

It is also important to mention that the model problem we consider is well-posed, the proof of this fact can be found in [26]; we also provide an outline of this proof in Section 5.4.1.

1.2.2 A Posteriori Error Estimation

It is now time to discuss the reliability of the reduced basis approximation we just constructed, i.e. how well $u_N(\mu)$ approximates $u(\mu)$. We claim that even for a moderate value of N the reduced basis approximation $u_N(\mu)$ will be quite close to the finite element solution $u(\mu)$. Unfortunately, we cannot determine this fact a priori, so we employ a posteriori error estimation: to verify the accuracy of $u_N(\mu)$ with respect to $u(\mu)$ we now describe a procedure to obtain a rigorous error bound $\Delta_N(\mu)$ such that $\Delta_N(\mu) \geq \|u(\mu) - u_N(\mu)\|_Y$ following the ideas developed in earlier works [56, 54, 49]. If our error bound $\Delta_N(\mu)$ is small, this fact will confirm our claim about the accuracy of the reduced basis approximation $u_N(\mu)$.

We start from defining the "error function" $e_N(\mu)$ as $e_N(\mu) = u(\mu) - u_N(\mu)$. We next write down the "error equation" which is obtained directly from (1.6) and (1.16).

$$\underbrace{a^A(e_N(\mu), v; \mu) + a^{NA}(e_N(\mu), v; \mu) + a^{NL}(u(\mu), v; \mu) - a^{NL}(u_N(\mu), v; \mu))}_{a^L(e_N(\mu), v; \mu)} = \underbrace{f(v) - a^A(u_N(\mu), v; \mu) - a^{NA}(u_N(\mu), v; \mu) - a^{NL}(u_N(\mu), v; \mu))}_{R(v; \mu)}. \quad (1.19)$$

The error equation (1.19) holds $\forall v \in Y$. As indicated in (1.19), we denote the residual functional $R(v; \mu) \in Y'$ as

$$R(v; \mu) = f(v) - a^A(u_N(\mu), v; \mu) - a^{NA}(u_N(\mu), v; \mu) - a^{NL}(u_N(\mu), v; \mu). \quad (1.20)$$

We next proceed using the ideas presented in [52, 57]. We start by introducing a bound con-

ditioner [18, 39, 52, 57] – a symmetric, continuous, and coercive bilinear form $\hat{a}(v, v)$ such that $\forall \mu \in \mathcal{D}$

$$\alpha(\mu)(\|v\|_Y)^2 \leq \hat{a}(v, v) \leq a^L(v, v; \mu), \quad \forall v \in Y. \quad (1.21)$$

The more detailed of description of different types of bound conditioners is contained in [52, 57].

We look for the error function $\hat{e}_N(\mu)$ which satisfies the following error equation

$$\hat{a}(\hat{e}_N(\mu), v) = R(v; \mu), \quad \forall v \in Y. \quad (1.22)$$

We next construct $\tilde{\Delta}_N(\mu)$ such that

$$a^L(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_N(\mu) = \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu)) \quad (1.23)$$

It remains to demonstrate that $a^L(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_N(\mu)$.

We then plug $v = e_N(\mu)$ into (1.19). At the moment we take for granted that

$$a^{NL}(u(\mu), e_N(\mu); \mu) - a^{NL}(u_N(\mu), e_N(\mu); \mu) = \int_{\Omega_{ni}} ((u(\mu))^4 - (u_N(\mu))^4)(u(\mu) - u_N(\mu)) \geq 0, \quad (1.24)$$

and hence

$$a^L(e_N(\mu), e_N(\mu); \mu) \leq R(e_N(\mu)). \quad (1.25)$$

Later in Chapter 5 we discuss the conditions under which (1.24) holds.

We plug $e_N(\mu)$ into (1.22) and invoke (1.21) and Cauchy-Schwartz' inequality to obtain:

$$\begin{aligned} a^L(e_N(\mu), e_N(\mu)) &\leq R(e_N(\mu); \mu) = \hat{a}(\hat{e}_N(\mu), e_N(\mu)) \leq \\ \hat{a}(e_N(\mu), e_N(\mu))^{\frac{1}{2}} \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu))^{\frac{1}{2}} &\leq a^L(e_N(\mu), e_N(\mu))^{\frac{1}{2}} \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu))^{\frac{1}{2}} \end{aligned} \quad (1.26)$$

which implies

$$a^L(e_N(\mu), e_N(\mu); \mu) \leq \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu)) = \tilde{\Delta}_N(\mu). \quad (1.27)$$

Finally, invoking (1.21) we obtain the error bound $\Delta_N(\mu)$ such that

$$\|e_N(\mu)\|_Y \leq \Delta_N(\mu) = \sqrt{\frac{\tilde{\Delta}_N(\mu)}{\alpha(\mu)}} = \sqrt{\frac{\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu))}{\alpha(\mu)}}. \quad (1.28)$$

We note that the calculation of $\Delta_N(\mu)$ requires evaluation of $a^L(e_N(\mu), e_N(\mu); \mu)$. It then can easily be shown that the lower and upper bounds for the output $s(\mu)$ are given by

$$s_N(\mu) \leq s(\mu) \leq s_N(\mu) + \tilde{\Delta}_N(\mu); \quad (1.29)$$

for details of the proof see Section 2.3.3.

1.2.3 Off-line/On-line Procedure: Fallacies and Limitations with respect to Non-affine and Non-linear Components

The usual way to treat problems with parametric dependence – as it was described in [49, 52, 42] – is to provide a so-called off-line/on-line decomposition which is a two stage computational effort. During the off-line stage – which we run only once – we compute and store all the necessary matrices and vectors for the next on-line stage. During the on-line stage – which we run repeatedly for each new parameter point μ – we construct the reduced basis approximation $u_N(\mu)$ and the error bound for it in a fast and efficient way. In this Section we show why such an off-line/on-line decomposition fails in the case of the model problem which we introduced in Section 1.1; this fact serves as a motivation for the development of new and more advanced reduced basis methods.

Reduced Basis Approximation

We first look at the off-line/on-line decomposition that would allow us to construct the reduced basis approximation $u_N(\mu)$. The application of the reduced basis methodology to problems with power function (such as $(u(\mu))^4$ in (1.4) has been addressed earlier in [56]. First, let us focus on the last term $a^{NL}(u_N(\mu), v; \mu)$ of (1.16). Due to (1.17) – the representation of $u_N(\mu)$ – and the non-linear nature of $a^{NL}(u_N(\mu), v; \mu)$ we can expand this term as follows:

$$a^{NL}(u_N(\mu), v; \mu) = \sigma \sum_{j,k,l,m=1}^N u_{N_j}(\mu) u_{N_k}(\mu) u_{N_l}(\mu) u_{N_m}(\mu) \int_{\Omega_{nl}} \zeta_j \zeta_k \zeta_l \zeta_m v. \quad (1.30)$$

Clearly, some of the terms in (1.30) repeat themselves. It is easy to demonstrate that the number of unique terms in the expansion (1.30) is equal to the cardinality of the set of uniquely ordered 4-tuples of integers $\underline{j} \in \{1, \dots, N\} = (j, j', j'', j''')$, such that $j \leq j' \leq j'' \leq j'''$. We denote this set of 4-tuples as \mathcal{P}_4^N . For the general case the cardinality of \mathcal{P}_κ^N is equal to¹

¹The proof of this fact is provided in Appendix A.

$$T(N, \kappa) = \frac{(N-1+\kappa)!}{(N-1)!\kappa!}. \quad (1.31)$$

In our particular case, $\kappa = 4$ and $T(N, \kappa) = \frac{(N+3)!}{(N-1)!\kappa!} = \frac{(N+3)(N+2)(N+1)N}{24}$. As we can see, $T(N, 4) \sim \frac{N^4}{24}$ for large values of N .

We now continue by explaining how the reduced basis approximation $u_N(\mu)$ is obtained from the numerical point of view. Inserting the representation (1.17) for $u_N(\mu)$ into (1.16) and invoking (1.30) we obtain a system of non-linear equations where $u_{N_i}, i = 1, \dots, N$ are the unknowns

$$\begin{aligned} \sum_{j=1}^N u_{N_j}(\mu) a^A(\zeta_j, \zeta_i; \mu) + \sum_{j=1}^N u_{N_j}(\mu) a^{NA}(\zeta_j, \zeta_i; \mu) + a^{NL} \left(\sum_{j=1}^N u_{N_j} \zeta_j, \zeta_i \right) - f(\zeta_i) = \\ \sum_{j=1}^N u_{N_j}(\mu) a^A(\zeta_j, \zeta_i; \mu) + \sum_{j=1}^N u_{N_j}(\mu) a^{NA}(\zeta_j, \zeta_i; \mu) + \\ \sigma \sum_{\underline{j} \in \mathcal{P}_4^N} u_{N_{j_1}}(\mu) u_{N_{j_2}}(\mu) u_{N_{j_3}}(\mu) u_{N_{j_4}}(\mu) \int_{\Omega_{nl}} \zeta_j \zeta_k \zeta_l \zeta_m \zeta_i = 0. \end{aligned} \quad (1.32)$$

We solve (1.32) using Newton method. The construction of the Jacobian matrix for (1.32) requires (i) the assembly of $a^A(\zeta_j, \zeta_i; \mu), i, j = 1, \dots, N$ – which admits off-line/on-line decomposition similar to the one used in the affine problems [49, 52, 42], the on-line operation count for this assembly scales as $O(QN^2)$, Q denotes the affine complexity of the problem, it is equal to 6 for the case of the model problem that we introduced in Section 1.1; (ii) the assembly of $a^{NA}(\zeta_j, \zeta_i; \mu), i, j = 1, \dots, N$ which requires $n_{na}N^2$ operations; to explain this operation count we note that the "local" nature of the problem suggests that out of all finite element test functions ϕ_1, \dots, ϕ_N there are only n_{na} functions, that is, $\tilde{\phi}_1, \dots, \tilde{\phi}_{n_{na}}$, the region of support of which has non-zero intersection with Ω_{na} ; hence, in evaluation of $a^{NA}(\zeta_j, \zeta_i; \mu)$ we only need to consider $\tilde{\phi}_l, l = 1, \dots, n_{na}$ and for each pair i, j this evaluation requires $O(n_{na})$ operations which, in turn, results from the properties of the finite element discretization; and (iii) the assembly of $a^{NL}(\zeta_j, \zeta_i; \mu), i, j = 1, \dots, N$ which requires $O(T(N, 4)N) = O(N^5)$ operations; this operation count results from the polynomial expansion (1.30). The construction of the f term requires $O(N)$ operations. Solving for the updated Newton direction requires $O(N^3)$ iterations. Therefore, the total number of operations per Newton iteration is $O(QN^2) + O(n_{na}N^2) + O(N^5) + O(N^3)$.

There are three important observations we need to make at this time. First we we note that it

is crucial for us to have n_{na} relatively small compared to \mathcal{N} , so that we still be able to maintain computational gains over the finite element method. Second, the number of terms in the non-linear polynomial expansion scales as N^5 . This is a very disturbing issue – even though we expect N to be small — because it might be quite problematic to obtain the computational advantage. Finally, with the current approach we are only able to treat power function non-linearities which imposes a significant limitation on the versatility and applicability of our methods.

Error Bound

We now discuss the off-line/on-line decomposition for the error bound $\Delta_N(\mu)$. As (1.28) indicates, in order to construct the error bound $\Delta_N(\mu)$ during the on-line stage we need to evaluate $\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu))$. The main idea behind the off-line/on-line decomposition is to represent $R(v; \mu)$ as

$$R(v; \mu) = \sum_{j=1}^K \tau_j(\mu) \Upsilon_j(v), \quad (1.33)$$

where $\Upsilon_j(v), j = 1, \dots, K$ are parameter independent functionals and K is the total number of terms in (1.20). We see the motivation for (1.33) if we invoke the representation (1.17) and bilinearity for the terms which comprise (1.33). We then can express $\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu))$ and hence $\Delta_N(\mu)$ as

$$\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu)) = \sum_{j,j'=1}^K \tau_j(\mu) \tau_{j'}(\mu) \Lambda_{jj'}, \quad (1.34)$$

$$\hat{a}(z_j, v) = \Upsilon_j(v), \quad (1.35)$$

$$\Lambda_{jj'} = \hat{a}(z_j, z_{j'}), \quad (1.36)$$

$$\Delta_N(\mu) = \sqrt{\frac{\sum_{j,j'=1}^K \Lambda_{jj'}}{\alpha(\mu)}}. \quad (1.37)$$

This is the approach that was used for error estimation in [37, 42, 49, 52, 41, 57] for the "purely" affine problems (in this case $K = (1 + QN)$). We precompute $\Lambda_{jj'}, j, j' = 1, \dots, K$ during the off-line stage. As we observe from (1.34), the operation count for the evaluation of $\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu))$ – and hence the construction of $\Delta_N(\mu)$ – for the error bound during the on-line stage is equal to K^2 . Apparently, to minimize the operation count during the on-line stage – and hence to enhance the efficiency of our reduced basis method – we would want to keep K as small as possible.

The problems with this approach when we deal with non-affine and non-linear problems now

become clear. First, the non-affine part $a^{NA}(u_N(\mu), v; \mu)$ does not admit a trivial decomposition that fits (1.33). What follows is that we cannot efficiently construct the error bound according to (1.28). The evaluation of $\Delta_N(\mu)$ will then require solution of an $\mathcal{N} \times \mathcal{N}$ linear system of equations. This will clearly destroy all of our efforts to construct an efficient off-line/on-line procedure while retaining the reliability of the method since in order to provide an error bound we would have to solve a system that is equal in size to the original problem.

Second, should we change the boundary condition (1.4) to a function of $u(\mu)$ which does not have a power law structure, say $\exp(u(\mu))$, we immediately face the same kind of problem we mentioned in the last paragraph: $a^{NL}(u_N(\mu), v; \mu)$ also does not admit a decomposition consistent with (1.33) which means that again we cannot efficiently construct the error bound $\Delta_N(\mu)$.

Third, let us go back to the case when the radiation boundary condition (1.4) holds. As it was just shown earlier, there are $O(N^4)$ terms in the residual $R(v; \mu)$ due to the polynomial expansion of $(u_N(\mu))^4$. Since the operation count for the error bound scales as the square of K , the number of terms in the residual, we will have at least an N^8 dependence in the operation count for the error bound (in fact, the operation count would be higher due to the cross-term contribution of the affine and non-affine parts). Though we hope that N remains small it becomes close to impossible to expect the same from N^8 .

We conclude that the current methodology has significant fallacies and limitations in treating locally non-affine and locally non-linear problems both in calculation of the reduced basis approximation and the error bound for it. This fact gives us further motivation to look for alternative ways to address locally non-affine and non-linear problems. We also note that the constraints for the *rigorous* error estimation are much more onerous than those for the efficient reduced basis approximation. For example, this discrepancy in constraints is exploited in [28] where reduced basis methods are applied to globally non-affine and globally non-linear problems by admitting a certain loss of rigor in error estimation.

1.3 Thesis Goals, Scope, Contributions, and Limitations

In this Section we set the goals we are willing to achieve with the new reduced methodology developed in this thesis. We briefly describe the classes of problems we are going to address with our new methods. We then discuss the limitations of our methods with respect to the classes of problems these methods can be applied to, the relation between the reduced basis approximation

and the finite element solution, and the operation counts associated with evaluation of this reduced basis approximation.

1.3.1 Goals

As we just discussed in Section 1.2.3, the current methodology does not allow us to obtain efficient and reliable reduced basis approximations for locally non-affine and non-linear problems. In this thesis we are going to address these issues by developing new efficient computational methods which can treat locally non-affine and locally non-linear problems while providing (i) accurate and fast reduced basis approximations for the field solution and the outputs of interests; (ii) *rigorous* and inexpensive error estimation for the reduced basis approximation; (iii) off-line/on-line framework that would allow us to obtain the reduced basis approximation and respective error bounds.

These goals are consistent with the standards established in the previous works [22, 42, 56, 41, 49, 52, 16]. We also look back at the affine problems and describe application of this new methodology to them.

1.3.2 Classes of Problems Considered

In this thesis we mainly apply our methods to the problems of heat transfer. Nevertheless, we maintain a sufficient level of abstraction which allows us to extend the reduced basis algorithm to other applications. In the most general case we look for the weak solution $u(\mu) \in Y = H^1(\Omega)$ such that

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in Y. \quad (1.38)$$

We require that the problems defined (1.38) are *well-posed*. We now briefly introduce the classes of problems we are going to address.

Affine Coercive Problems

When the underlying strong form of partial differential equations is linear $a(w, v; \mu)$ in (1.38) is a bilinear form. We impose certain regularity conditions on $a(w, v; \mu)$, including coercivity and continuity. We explain these requirements in greater detail in Section 2.1.

Under these regularity conditions a problem is called affine when $a(\cdot, \cdot; \mu)$ admits the following

decomposition

$$a(w, v; \mu) = \sum_{q=1}^Q \Theta_q(\mu) a^q(w, v), \quad \forall w, v \in Y, \quad (1.39)$$

where $a^q(w, v), q = 1, \dots, Q$ do not have any dependence on the parameters.

These problems and application of the reduced-basis method to them were described in [42, 53, 37]. Essentially, when we describe application of the reduced basis method to affine coercive problems we summarize the results obtained in earlier works [24, 25, 41, 57, 49, 52].

Such problems have many applications in engineering and science, for instance a micro-truss example or a multipurpose structure studied in [52, 16] and many more.

Locally Non-affine Coercive Problems

We now broaden the class of affine problems by introducing a new component to the decomposition of $a(w, v; \mu)$; the latter again has to satisfy certain regularity conditions. We now allow more general dependence of $a(w, v; \mu)$ on the parameters:

$$a(w, v; \mu) = a^{NA}(w, v; \mu) + \sum_{q=1}^Q \Theta_q(\mu) a^q(w, v), \quad \forall w, v \in Y, \quad (1.40)$$

where again $a^q(w, v), q = 1, \dots, Q$ are parameter independent bilinear forms but at the same time $a^{NA}(w, v; \mu)$ has an unrestricted general dependence on the parameters. We are going to consider *locally* non-affine problems, for which we have affine parameter dependence almost everywhere in Ω with the exception of a small (compared to Ω) area $\Omega_{na} \subset \Omega$. To be rigorous, we require that $\forall v, w \in Y : v|_{\Omega_{na}} = 0, w|_{\Omega_{na}} = 0$

$$a^{NA}(w, v; \mu) \equiv 0, \quad \forall \mu \in \mathcal{D}. \quad (1.41)$$

Locally Non-linear Problems

We now are going to consider non-linear problems. We assume that $a(w, v; \mu)$ can be decomposed in the following way

$$a(w, v; \mu) = a^A(w, v; \mu) + a^{NL}(w, v; \mu), \quad \forall w, v \in Y, \quad (1.42)$$

where $a^A(\cdot, \cdot; \mu)$ is a coercive, continuous and affine bilinear form (for more detail see Section 2.1) and $a^{NL}(\cdot, \cdot; \mu)$ is an operator that is non-linear in its first and linear in its second argument.

We are going to consider *locally* non linear problems, i.e. those problems for which the non-linearity is only limited to the area $\Omega_{nl} \subset \Omega$ in such a way that $a^{NL}(w, v; \mu)$ that can be expressed as follows

$$a^{NL}(w, v; \mu) = \int_{\Omega_{nl}} g(w; \mu)v, \quad \forall w, v \in Y, \quad (1.43)$$

where $\forall \mu \in \mathcal{D} \quad g(w(\underline{x}); \mu) : \mathbb{R} \times \mathcal{D} \rightarrow \mathbb{R} \in C^\infty(\mathbb{R}), \underline{x} \in \mathbb{R}^d$. Since we only consider well-posed problems, we require that the integral defined in (1.43) is well-defined for all $w, v \in Y$.

We assume that the area of Ω_{nl} is small compared to that of Ω (or that Ω_{nl} is of lower dimension than Ω) and such that the number of nodes in the discretization of Ω_{nl} is small compared to that of Ω . We employ Newton method to obtain the solution $u(\mu)$ of our locally non-linear problem.

The two-dimensional thermal fin problem we considered in Section 1.1 belongs to a mixed locally non-affine - locally non-linear type since it combines the properties that are distinguishing for both types. The radiation boundary conditions produce the non-linear part, whereas the non-affine contribution appears in the weak form because of the varying conductivity $k_5(\underline{x}; \mu)$.

1.3.3 Limitations

Generality of Problems Classes

This thesis shows how the problems mentioned in Section 1.3.2 can be treated in accordance with the goals that were set in Section 1.3.1. At the moment reduced basis methods could be applied to very comprehensive classes of problems and the research work is carried out to extend the versatility of the reduced basis methods. We present a significant extension to the variety of problems to which reduced basis methods can be applied.

However, this thesis does not cover all the problems to which the reduced basis methods are currently applied, such as non-coercive [22, 22, 23], time-dependent parabolic [21, 10, 12, 48], globally non-nonaffine [28], globally non-linear [28, 55, 54] optimization [36], and design [2] problems. At the same time we believe that the new methods described in this thesis can be extended to these problems. We cover this issue when we speak about the future work.

The other important issue is that we only provide the approximation with respect to the finite element solution which, in turn, is an approximation itself with respect to the analytical solution.

When the finite element mesh is fine enough the finite element solution is indeed very close to the analytical one. The convergence of the finite element method is described in [51].

Operation Count for Locally Non-affine and Non-linear Problems

As we stated earlier, our main goal is to achieve computational efficiency and at the same time deliver the certified reliability of our approximations in a mathematically rigorous way. We demonstrate that for the locally non-affine and locally non-linear problems the operation count for the whole algorithm depends on the number of points in the Ω_{na} or Ω_{nl} . We also show that this dependence is linear or slightly higher than linear at most. This fact imposes a limitation on the size of Ω_{na} and Ω_{nl} . In some cases we show that we can eliminate such a debilitating dependence by admitting a slight loss of rigor.

1.4 Approaches

In this Section we list the reduced basis methods – to wit, the classical reduced basis method (CRBM), the partition of unity method (PUM), and the minimax coefficient approximation method (MCAM) – we are going to consider in this thesis and briefly describe the main features of these methods. The PUM and the MCAM and their application parameter-based problems constitutes the main contribution of this thesis to the development of the reduced basis methods. We also point to the fact that these new methods incorporate the key ingredients from prior work [42, 8, 47, 6, 56, 49, 44, 41].

1.4.1 "Classical" Reduced Basis Method

In general, the reduced basis method rely on the ideas which appeared in the scientific literature in the late 1970s - early 1980s [3, 27, 33]; the reduced basis methods and their applications were developed later by Noor [34, 29, 30, 32, 35, 31] and other scientists [8, 39, 6, 47, 9, 38, 13, 14, 40, 15, 4]. We use the reduced basis methods in application to affine problems following the results of [49, 52, 57, 42]. For every given parameter point μ the reduced basis methods rely on the projection of the solution of the underlying affine problem on a low-dimensional space W_N spanned by N previously precomputed solutions; the details of the method are given in the next Chapter. The description of the method is presented to prepare the necessary background for the

new approaches and essentially summarizes the ongoing effort of our research group [49, 52, 42, 56, 41, 55, 54, 43, 11, 53, 57], hence the term "classical".

The classical reduced basis method serves as a foundation for two new approaches: the partition of unity method (PUM) and the minimax coefficient approximation method (MCAM). The "classical" reduced basis method complexity depends on Q , the number of affine parameters, and N , the dimension of the low-dimensional function space mentioned above.

1.4.2 Partition of Unity Method

The partition of unity method presents an enhancement to the "classical" reduced basis which enables us to address locally non-affine problems. The construction of the PUM to a great extent relies on the ideas presented in [20, 19]. The key components of the PUM are – the projection of the "truth" solution onto the low-dimensional function space W_N spanned by N previously pre-computed solutions and the subsequent projection on the space X_S of S functions which constitute a partition of unity over the domain Ω . This second projection allows us to decompose the error bound into a sum of contributions from the regions of support of these "partition-of-unity" functions. Such a decomposition decouples the error estimates from the affine and the non-affine parts of the domain; this last result is used in *rigorous* error estimation. One interesting feature of the partition of unity method is that it can also address affine problems in an essentially different way than the classical reduced basis method. We provide a description of this feature in Chapter 3; we also demonstrate that the PUM complexity depends on Q , N , S , and n_{na} , the number of points in the non-affine region Ω_{na} , presumably small compared to \mathcal{N} .

1.4.3 Minimax Coefficient Approximation Method

The minimax coefficient approximation approach (MCAM) is yet another enhancement of the "classical" reduced basis method; the new method is capable of addressing both locally non-affine and locally non-linear problems. The MCAM also relies on the projection of the parameter-dependent solution onto the low-dimensional function space W_N spanned by N previously pre-computed solutions – which only emphasizes the importance of the "classical" reduced basis method – and the replacement of the underlying non-affine or non-linear part in the weak form with an affine-like approximation based on the ideas presented in [5]. The complexity of the MCAM – as we show in Chapters 4 and 5 – depends on Q , N , M , the number of terms in this affine-like approximation,

and $n_{na}(n_{nl})$. It is important to mention that the MCAM could also be applied to *globally* non-affine and *globally* non-linear problems [28] with a certain loss of rigor in error estimation which in a sense reflects our previous observation about the comparison between the constraints imposed by rigorous error estimation and efficient reduced basis approximation. In this thesis we provide rigorous bounds for all of the reduced basis approximations.

1.4.4 Key Ingredients from Prior Work

Fast and Accurate Reduced Basis Approximation

Solving a problem that belongs to one of the classes we listed in Section 1.3.2 can become overly computationally expensive especially when repeated evaluation of the output or field solution is required. Instead of obtaining the “truth” solution we come up with a reduced approximation $u_N(\mu)$ which requires very moderate computational effort compared to the regular finite element method. The reduced basis approximation relies on one of the methods of our choice — depending on to which class the problem belongs — the classical reduced basis, the partition of unity or the minimax coefficient approximation method. The reduced basis approximation $u_N(\mu)$ is obtained through the projection of the “truth” solution $u(\mu)$ onto N -dimensional space W_N spanned by previously precomputed solutions [42, 49, 52].

Rigorous and Reliable Error Estimation

Once the reduced basis approximation $u_N(\mu)$ is obtained, we need to verify its accuracy and reliability with respect to the “truth” solution $u(\mu)$. We give a description of a computationally efficient procedure which allows us to provide error bounds for $\|u(\mu) - u_N(\mu)\|_Y$ or $|l(u(\mu)) - l(u_N(\mu))|$ where $l(\cdot)$ is a bounded linear functional in Y' . There are several common ideas [8, 47, 6, 42, 56, 49, 52] which are central to this error estimation procedure for all three (CRBM, PUM, MCAM) reduced basis methods we consider, however the particular implementation depends on the class of the problem and our method of choice; the details are given in Sections 2.3.3, 3.2.3, 4.1.5, and 5.3.2.

Framework

All the three (CRBM, PUM, MCAM) methods we mentioned are developed within the standards of a computational framework which consists of two computational stages which we refer to as the “off-line” stage and the “on-line” stage. The off-line stage is a one-time computational “investment”

during which we precompute and store all the necessary components for the on-line stage. Once the off-line stage is completed, we are ready to obtain numerical results with certified accuracy using the reduced basis method of our choice. For each new value of the parameter point μ we obtain (i) a reduced basis approximation as an element of the low dimensional reduced basis space and (ii) the error bound for our reduced basis approximation. Steps (i) and (ii) complete the on-line stage which is executed repeatedly for every new parameter point. This computational framework is one of the earlier contributions [56, 49, 44, 41, 56] of our research group to the development of the reduced basis methods.

1.5 Thesis Outline

In this thesis we focus on applying three different reduced basis methods – namely, CRBM, PUM, and MCAM – to affine, locally non-affine and locally non-linear problems. All of our model problems are heat transfer problems, but the methods we describe could be easily applied to other areas of science as we maintain a sufficient level of abstraction. In Chapter 2 we present the mathematical definition of affine coercive problems and review [42, 41, 49, 52] application of the reduced basis methods to them. In Chapter 3 we present the mathematical definition of locally non-affine problems and introduce the partition of unity method. We describe application of this method to affine and locally non-affine problems and provide concrete numerical results for three model problems which reflect different aspects of the method. In Chapter 4 we describe the minimax coefficient approximation method and its application to locally non-affine problems. In Chapter 5 we define locally non-linear problems in a mathematically rigorous way and describe application of the minimax coefficient approximation method to them. In each of the Chapters 4 and 5 we formulate a new relevant model problem; for each model problem numerical results are provided. In Chapter 6 we conclude the thesis and discuss the directions of the future work.

Chapter 2

Affine Coercive Problems

In this Chapter we provide the description of the classical reduced basis method (CRBM) and its application to coercive affine problems. Both locally non-affine and locally non-linear problems result from an extension of the class of affine problems and thus still contain an affine component which makes it essential to understand the definition of affine problems; these details are provided in Section 2.1. We revisit the methodology which was developed in earlier works [49, 52, 57, 42]. The details of the classical reduced basis method are very important components of this thesis since they provide the necessary background for subsequent construction of the partition of unity and minimax coefficient approximation methods. The ideas behind classical reduced basis method remain central to our new methodology and because of that later in this thesis we are going to reference the theoretical results presented in this Chapter. Some of the theoretical results and algorithms of this Chapter, such as for example, the sample selection algorithm (Section 2.3.4) and the error estimation procedure for the linear functional outputs (Section 2.3.3), are the essential components of both the partition of unity (PUM) and the minimax coefficient approximation methods. The description of the CRBM also allows us to better illustrate the enhancements which were developed in our new methods.

At the end of this Chapter we consider a 2D heat conduction problem to which we apply the CRBM. We analyze the performance of the method by studying the convergence and the accuracy of our reduced basis approximations, the sharpness of our error estimates and the computational effort which is necessary to complete the off-line stage and to run the on-line stage. This analysis in a way sets the standard by which we test our subsequent developments.

The dimension reduction idea [27, 49, 52, 42] is a key component for all the reduced basis

methods presented in this thesis. The error estimation of the partition of unity method – as we show in Chapter 3 – relies on the breaking the main problem into a several smaller problems using domain decomposition ideas [18, 19]; many of these smaller sub-problems appear to be affine and we treat them in a similar fashion as we treat affine problems in this Chapter. In Chapter 3 we also provide an alternative approach to address affine coercive problems based on the partition of unity method which we later compare to the CRBM applying the performance analysis similar to the one presented in Section 2.4.4.

In the minimax coefficient approximation method we essentially replace the original locally non-affine or non-linear component with an affine-like approximation [5] which again shows the relevance of the classical reduced basis method since the subsequent treatment of the modified locally non-affine or non-linear problem to a great extent uses the ideas presented in this Chapter.

2.1 General Abstract Problem Statement

We consider a suitably regular (smooth) domain $\Omega \subset \mathbb{R}^d, d = 1, 2, \text{ or } 3$, and associated function space $Y \subset (H^1(\Omega))^d$, where

$$H^1(\Omega) \equiv \left\{ v \mid v \in L^2(\Omega), \nabla v \in (L^2(\Omega))^d \right\}, \quad (2.1)$$

and

$$L^2(\Omega) \equiv \left\{ v \mid \int_{\Omega} v^2 < \infty \right\}. \quad (2.2)$$

The inner product (\cdot, \cdot) in Y is given by

$$(w, v)_{H^1(\Omega)} \equiv \int_{\Omega} \nabla w \nabla v + wv, \forall w, v \in Y. \quad (2.3)$$

The norm in $\|\cdot\|$ in Y is defined as

$$\|v\|_Y = ((v, v)_Y)^{\frac{1}{2}}, \forall v \in Y. \quad (2.4)$$

The corresponding dual space of Y, Y' is then defined as the space of all functionals $g(v)$ such

that the dual norm of $g(v)$, defined as,

$$\|g\|_{Y'} \equiv \sup_{v \in Y} \frac{g(v)}{\|v\|_Y}, \quad (2.5)$$

is bounded. We also introduce the “representation” operator $\mathcal{Y}: Y' \rightarrow Y$ such that, for any $g \in Y'$,

$$(\mathcal{Y}g, v)_Y = g(v), \quad (2.6)$$

then

$$\|g\|_{Y'} = \|\mathcal{Y}g\|_Y, \quad (2.7)$$

which is the direct result of the Riesz representation theorem.

As in the previous chapter, we define a parameter set $\mathcal{D} \subset \mathbb{R}^P$, a particular point in which is denoted μ . Note that Ω is a parameter independent *reference domain*.

We then introduce a bilinear form $a(., .; .) : Y \times Y \times \mathcal{D} \rightarrow \mathbb{R}$ which corresponds to the weak form of the underlying partial differential equations. We describe the procedure for obtaining a particular instantiation of $a(., .; .)$ in Section 2.4. We assume that $a(., .; .)$ is continuous: there exists $\gamma(\mu) > 0$ such that

$$a(w, v; \mu) \leq \gamma(\mu) \|w\|_Y \|v\|_Y \leq \gamma_0 \|w\|_Y \|v\|_Y, \quad \forall \mu \in \mathcal{D}; \quad (2.8)$$

furthermore, we assume that $a(., .; .)$ is coercive: there exists $\alpha(\mu) > 0$ such that

$$0 < \alpha_0 \leq \alpha(\mu) = \inf_{v \in Y} \frac{a(v, v; \mu)}{\|v\|_Y^2}, \quad \forall \mu \in \mathcal{D}, \quad (2.9)$$

and symmetric, $a(w, v; \mu) = a(v, w; \mu)$, $\forall w, v \in Y$, $\forall \mu \in \mathcal{D}$.

The weak solution of the partial differential equations $u(\underline{x}; \mu)$ is a function in Y which satisfies

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in Y. \quad (2.10)$$

We will often omit the dependence on the space coordinate $\underline{x} \in \mathbb{R}^d$ in $u(\underline{x}; \mu)$ and use $u(\mu)$ instead. We refer to (2.10) as the weak form of the partial differential equations. In practice, to obtain a solution of (2.10) we replace Y with an appropriate “truth” finite element approximation space $Y_{\mathcal{N}}$, where \mathcal{N} is the dimension of $Y_{\mathcal{N}}$. We assume that $Y_{\mathcal{N}}$ is sufficiently rich and the analytical solution

$u(\mu)$ and the finite element solution $u_{\mathcal{N}}(\mu)$ are practically indistinguishable from each other. To emphasize this point we will mostly refer to the latter as $u(\mu)$ omitting the subindex \mathcal{N} . We obtain $u(\mu)$ by applying standard Galerkin projection to (2.10). More information on the convergence of the finite element methods could be found in [51].

We now make certain assumptions with respect to the parametric dependence of $a(\cdot, \cdot; \cdot)$. We assume that there exists a decomposition

$$a(w, v; \mu) = \sum_{i=1}^Q \Theta_i(\mu) a^i(w, v), \quad \forall w, v \in Y, \quad (2.11)$$

where Q is the parametric complexity of the problem and $a^q(w, v), q = 1, \dots, Q$ is a parameter independent bilinear form.

2.2 Dimension Reduction Idea

We now present a general heuristic argumentation behind the reduced basis methods. The arguments we use apply not only to affine problems but also to locally non-affine and locally non-linear problems. For every value of μ the finite element solution $u(\mu)$ belongs to the \mathcal{N} -dimensional space Y , or in other words $u(\mu)$ has \mathcal{N} degrees of freedom. However, we can reasonably argue that with variation of μ over \mathcal{D} $u(\mu)$ does not span the whole space Y . Instead, we make a critical observation that $u(\mu)$ varies continuously with μ , or in the words $u(\mu)$ belongs to a continuous manifold with dimension P much lower than \mathcal{N} ; this is schematically depicted in Figure 2-1. If we consider well-posed problems, the continuity of the manifold could be rigorously proven by considering the sensitivity derivatives of (2.10) with respect to the components of the parameter-vector μ . This idea is fundamental for construction of our methodology, both in this and in the subsequent Chapters of this thesis; this idea was first introduced in [27] and later developed in [34, 29, 30, 32, 35, 31, 57, 42].

Based on the observation that the field solution $u(\mu)$ evolves on a low-dimensional manifold, we make a hypothesis that it is possible to approximate the field solution $u(\mu)$ for any arbitrary $\mu \in \mathcal{D}$ as a linear combination of certain functions $\zeta_i, i = 1, \dots, N$, where N is presumably a small number compared to \mathcal{N} . Unfortunately, we cannot verify the validity of our hypothesis a priori, therefore we employ a posteriori error estimation to ensure reliability of the methods we present. In the ‘‘classical’’ reduced basis formulation for affine problems the functions $\zeta_i, i = 1, \dots, N$ are

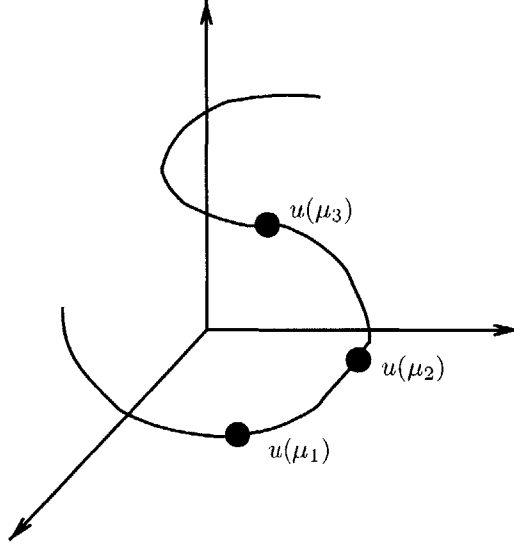


Figure 2-1: Critical observation illustration: a low dimensional manifold.

chosen to be the solutions of the original weak problem (2.10) for N chosen points $\mu_i, i = 1, \dots, N$. This idea with certain extensions remains one of the key components of the partition of unity and the minimax coefficient approximation methods. The extensions arise from the fact that we cannot efficiently solve locally non-affine and non-linear problems using the “classical” reduced basis method.

2.3 “Classical” Reduced Basis Method

In this Section we give an overview of the reduced basis method applied to affine coercive problems.

2.3.1 Reduced Basis Approximation

By the preceding arguments, we see that to approximate $u(\mu)$, and hence $s(\mu)$, we might not need to represent every possible function in Y ; instead, we only need to approximate the functions in the low-dimensional manifold “spanned” by $u(\mu)$. We could, therefore, simply calculate the solution u at several points on the manifold corresponding to different values of μ ; then, for any new parameter value, μ^{new} , we could “interpolate between the points,” that is, approximate $u(\mu^{\text{new}})$ by some linear combination of the known solutions.

Being more specific, we start by introducing a sample in parameter domain \mathcal{D}

$$S_N = \{\mu_1, \dots, \mu_N\}, \tag{2.12}$$

where $\mu_i \in \mathcal{D}, i = 1, \dots, N$. We then define our reduced-basis approximation space W_N as

$$W_N = \text{span}\{\zeta_i \equiv u(\mu_i), i = 1, \dots, N\}, \quad (2.13)$$

where $u(\mu_i) \in Y$ is the solution to (2.10) for $\mu = \mu_i$. For any $\mu \in \mathcal{D}$, we define our reduced-basis approximation for the field solution as $u_N(\mu)$ which is obtained through Galerkin projection of $u(\mu)$ onto W_N

$$a(u_N(\mu), v; \mu) = f(v), \quad \forall v \in W_N. \quad (2.14)$$

The reduced basis approximation for the output $s(\mu) = l(u(\mu))$ is given by

$$s_N(\mu) = l(u_N(\mu)). \quad (2.15)$$

From (2.14) we see that $u_N(\mu)$ can be expressed as a linear combination of basis function of W_N

$$u_N(\mu) = \sum_{j=1}^N u_{Nj}(\mu) \zeta_j. \quad (2.16)$$

We refer to the case when $l(v) = f(v)$ as compliant.

2.3.2 A Priori Convergence Arguments

We now consider the rate at which $u_N(\mu)$ and $s_N(\mu)$ converge to $u(\mu)$ and $s(\mu)$, respectively. To begin, it is standard to demonstrate the optimality of $u_N(\mu)$ in the sense that

$$\|u(\mu) - u_N(\mu)\|_Y \leq \sqrt{\frac{\gamma(\mu)}{\alpha(\mu)}} \inf_{w_N \in W_N} \|u(\mu) - w_N\|_Y \leq \sqrt{\frac{\gamma_0}{\alpha_0}} \inf_{w_N \in W_N} \|u(\mu) - w_N\|_Y. \quad (2.17)$$

where $\|\cdot\|_Y$ is the norm associated with Y , and $\alpha(\mu)$ and $\gamma(\mu)$ are continuity and coercivity constants associated with the bilinear form $a(\cdot, \cdot; \cdot)$. Furthermore, for the compliant output from the orthogonality of the Galerkin projection and the continuity of $a(\cdot, \cdot; \cdot)$ it follows that

$$s(\mu) - s_N(\mu) = a(u(\mu) - u_N(\mu), u(\mu) - u_N(\mu); \mu) \leq \gamma(\mu) (\|u(\mu) - u_N(\mu)\|_Y)^2 \leq \gamma_0 (\|u(\mu) - u_N(\mu)\|_Y)^2. \quad (2.18)$$

As we observe, $s_N(\mu)$ converges to $s(\mu)$ as the square of the error in $u_N(\mu)$.

At this point we have not discussed the algorithm of how to select the sample points, $\mu_i, i = 1, \dots, N$. In Section 2.3.4 we present an efficient “greedy” algorithm that covers this matter.

These theoretical considerations suggest that N may, indeed, be chosen very small. However, we note that both the left and the right hand sides of (2.14) (and therefore $\underline{u}_N(\mu)$ and $s_N(\mu)$) depend on the basis functions, $\zeta_i, i = 1, \dots, N$ and are therefore potentially very expensive to evaluate. Our goal then is to deliver a framework that will allow us to calculate $\underline{u}_N(\mu)$ and $s_N(\mu)$ in a computationally efficient way.

2.3.3 A Posteriori Error Estimation

A Posteriori Error Bound for $\|e_N(\mu)\|_Y$ and Compliant Output

After obtaining the reduced basis approximation $u_N(\mu)$, we proceed to the next step during which we verify the accuracy of this approximation. We start by introducing a linear functional $R(v; \mu) \in Y'$ which we refer to as the residual

$$R(v; \mu) = f(v) - a(u_N(\mu), v; \mu), \forall v \in Y; \quad (2.19)$$

we next define the “error function” as

$$e_N(\mu) = u(\mu) - u_N(\mu). \quad (2.20)$$

Invoking (2.10) and (2.14), we notice that $e_N(\mu)$ satisfies the following “error equation”

$$a(e_N(\mu), v; \mu) = R(v; \mu), \forall v \in Y. \quad (2.21)$$

We next proceed using the ideas presented in [52, 57]. We start by introducing a bound conditioner [18, 39, 52, 57] – a symmetric, continuous, and coercive bilinear form $\hat{a}(v, v; \mu)$ such that $\forall \mu \in \mathcal{D}, \forall v \in Y$

$$\alpha(\mu)(\|v\|_Y)^2 \leq \rho_{min}(\mu)\hat{a}(v, v; \mu) \leq a(v, v; \mu) \leq \rho_{max}(\mu)\hat{a}(v, v; \mu) \leq \gamma(\mu)(\|v\|_Y)^2, \quad (2.22)$$

where

$$1 \leq \rho_{min}(\mu), \quad \rho_{max}(\mu) \leq \rho, \quad \alpha_0 \leq \alpha(\mu), \quad \gamma(\mu) \leq \gamma_0, \quad (2.23)$$

for some (preferably small) constant $\rho \in \mathbb{R}$. The more detailed of description of different types of bound conditioners is contained in [52, 57].

We next provide the error bound for the "compliant" output (when $l(v) = f(v), \forall v \in Y$) as we mentioned in Section (2.3.1). We invoke (2.10) and (2.14) to obtain

$$a(e_N(\mu), e_N(\mu); \mu) = f(u(\mu)) - f(u_N(\mu)) = s(\mu) - s_N(\mu), \quad (2.24)$$

and, hence, based on the cocrcivity of $a(\cdot, \cdot; \cdot)$:

$$s_N(\mu) \leq s(\mu). \quad (2.25)$$

We look for the error function $\hat{e}_N(\mu)$ which satisfies the following error equation

$$\hat{a}(\hat{e}_N(\mu), v; \mu) = R(v; \mu), \forall v \in Y. \quad (2.26)$$

We next construct $\tilde{\Delta}_N(\mu)$ such that

$$a(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_N(\mu) = \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu); \quad (2.27)$$

it remains to demonstrate that $a(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_N(\mu)$. We plug $e_N(\mu)$ into (2.26) and invoke (2.21), (2.22) and Cauchy-Schwartz' inequality:

$$\begin{aligned} a(e_N(\mu), e_N(\mu)) &= R(e_N(\mu); \mu) = \hat{a}(\hat{e}_N(\mu), e_N(\mu); \mu) \leq \\ &(\hat{a}(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} (\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu))^{\frac{1}{2}} \leq \rho_{min}^{-\frac{1}{2}} (a(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} (\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu))^{\frac{1}{2}}, \end{aligned} \quad (2.28)$$

which implies

$$a(e_N(\mu), e_N(\mu); \mu) \leq \rho_{min}^{-1} \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu) \leq \frac{\tilde{\Delta}_N(\mu)}{\rho_{min}}. \quad (2.29)$$

Since $\rho_{min} \geq 1$ we can write that

$$s_N(\mu) \leq s(\mu) \leq s_N(\mu) + \tilde{\Delta}_N(\mu). \quad (2.30)$$

We now define the effectivity $\tilde{\eta}_N(\mu)$ to measure the sharpness of our error prediction for the com-

pliant output such that

$$\tilde{\eta}_N(\mu) = \frac{\tilde{\Delta}_N(\mu)}{s(\mu) - s_N(\mu)} = \frac{\tilde{\Delta}_N(\mu)}{a(e_N(\mu), e_N(\mu); \mu)}. \quad (2.31)$$

We then readily construct the error bound $\Delta_N(\mu)$ for $\|e_N(\mu)\|_Y$ just by invoking the coercivity property from (2.22)

$$\|e_N(\mu)\|_Y \leq \Delta_N(\mu) = \sqrt{\frac{\tilde{\Delta}_N(\mu)}{\alpha(\mu)}}, \quad (2.32)$$

and hence

$$\tilde{\Delta}_N(\mu) = \alpha(\mu)(\Delta_N(\mu))^2. \quad (2.33)$$

We also introduce the effectivity for $\Delta_N(\mu)$

$$\eta_N(\mu) = \frac{\Delta_N(\mu)}{\|e_N(\mu)\|_Y}. \quad (2.34)$$

As we can see from (2.32), (2.27), both effectivities $\tilde{\eta}_N(\mu)$ and $\eta_N(\mu)$ are bounded from below by 1. On the other hand, we would like to keep the effectivity as close to 1 as possible, so that our error bound remains sharp. We next provide a priori effectivity analysis which allows us to obtain a priori bounds for $\tilde{\eta}_N(\mu)$ and $\eta_N(\mu)$.

We note that (2.27) and (2.32) hold for any choice of bound conditioner that satisfies (2.22) and (2.23). If we choose the bound conditioner as $\hat{a}(v, v; \mu) = \alpha(\mu)\|v\|_Y^2$ – which obviously satisfies (2.22) and (2.23) – it is then standard to show that

$$\Delta_N(\mu) = \frac{1}{\alpha(\mu)}\|R(v; \mu)\|_{Y'}, \quad (2.35)$$

and hence using (2.33)

$$\tilde{\Delta}_N(\mu) = \frac{1}{\alpha(\mu)}(\|R(v; \mu)\|_{Y'})^2. \quad (2.36)$$

The calculation of the dual norm $\|R(v; \mu)\|_{Y'}$ becomes an easy matter once we invoke (2.6) and (2.7).

A Priori Effectivity Analysis

We now prove that

$$\rho_{min}(\mu) \leq \tilde{\eta}_N(\mu) \leq \rho_{max}(\mu). \quad (2.37)$$

The proof for the lower bound follows from (2.24) and (2.28); for the upper bound we appeal to (2.21), (2.26) and invoke Cauchy-Schwartz' inequality:

$$\begin{aligned} \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu) &= R(\hat{e}_N(\mu); \mu) = a(\hat{e}_N(\mu), e_N(\mu)) \leq \\ (a(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} (a(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu))^{\frac{1}{2}} &\leq \rho_{\max}^{\frac{1}{2}} (a(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} \hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu), \end{aligned} \quad (2.38)$$

from which the desired result readily follows. Invoking (2.22) and (2.37) we obtain the lower and upper bound for $\eta_N(\mu)$

$$\sqrt{\rho_{\min}(\mu)} \leq \sqrt{\frac{\tilde{\Delta}_N(\mu)}{a(e_N(\mu), e_N(\mu); \mu)}} \leq \sqrt{\frac{\tilde{\Delta}_N(\mu)}{\alpha(\mu)(\|e_N(\mu)\|)^2}} = \sqrt{\frac{\Delta_N(\mu)}{\|e_N(\mu)\|}}, \quad (2.39)$$

$$\sqrt{\frac{\Delta_N(\mu)}{\|e_N(\mu)\|}} = \sqrt{\frac{\tilde{\Delta}_N(\mu)}{\alpha(\mu)(\|e_N(\mu)\|)^2}} \leq \sqrt{\frac{\gamma(\mu)\tilde{\Delta}_N(\mu)}{\alpha(\mu)(\|e_N(\mu)\|)^2}} \leq \sqrt{\frac{\rho_{\max}(\mu)\gamma(\mu)}{\alpha(\mu)}} \leq \sqrt{\frac{\rho_{\max}(\mu)\gamma_0}{\alpha_0}}. \quad (2.40)$$

A Posteriori Error Bound for Linear Outputs

We now consider the case when our output of interest $s(\mu)$ is given by an arbitrary functional $l(v) \in Y'$ such that $l(v) \neq f(v)$. We first observe that for any $l(v) \in Y'$ we can write that

$$|l(e_N(\mu))| = |l(u(\mu)) - l(u_N(\mu))| \leq \|l\|_{Y'} \|e_N\|_Y \leq \|l\|_{Y'} \Delta_N(\mu), \quad (2.41)$$

so once the error bound $\Delta_N(\mu)$ is obtained we can provide an upper and lower bound for the reduced basis approximation $s_N(\mu) = l(u_N(\mu))$ for any linear functional in Y' . The calculation of the dual norm becomes an easy matter if we employ the Riesz representation operator 2.6 and the direct result of the Riesz representation theorem. According to (2.41) $s_N(\mu)$ converges to $s(\mu)$ linearly with $\|e_N(\mu)\|_Y$.

There also exists an alternative ‘‘duality’’ approach [49, 22, 42] for constructing the error bound for the output $s(\mu)$ which enables us to recover the convergence effect similar to the ‘‘square’’ convergence of the compliant output.

2.3.4 Sample Selection Algorithm

We now provide an inexpensive ‘‘greedy’’ algorithm for selecting the sample S_N . Let us assume that we have a big pool – which we call S_{pool} – of N_{pool} points $\{\mu_1^P, \dots, \mu_{N_{\text{pool}}}^P\}$ to pick from. We

need to select $N \ll N_{pool}$ points $\{\mu_1, \dots, \mu_N\}$ from S_{pool} to form S_N .

We start by selecting the first point randomly, or just picking the first point in the list μ_1^P and setting $\mu_1 = \mu_1^P$. We then set $k = 1$, $S_k = \{\mu_1\}$ and build the off-line/on-line framework (the details of this procedure are provided in Section 2.3.5).

We next go over all the remaining points in S_{pool} and select the next point as

$$\mu_{k+1} = \arg \max_{S_{pool}} \Delta_N(\mu), \quad (2.42)$$

or

$$\mu_{k+1} = \arg \max_{S_{pool}} \|e_N(\mu)\|_Y, \quad (2.43)$$

which we add to S_k . We note that both $\Delta_N(\mu_i)$ and obviously $\|e(\mu_i)\|, i = 1, \dots, k$ are equal to zero. We then set $k = 2$. We keep repeating step (2.42) or (2.43) and incrementing k till we reach $k = N$.

The choice of (2.42) or (2.43) depends on how big N_{pool} is. Clearly, (2.42) is cheaper to evaluate since (2.43) requires solving $\mathcal{N} \times \mathcal{N}$ linear system. If our bounds are sharp enough or, in other words, effectivities do not divert from 1 significantly, the first choice is computationally advantageous and produces almost the same outcome as the second one.

When choosing the sample S_{pool} we should pay attention to P , the dimension of the parameter domain \mathcal{D} . The most intuitive choice is to select S_{pool} as a relatively fine uniform mesh over \mathcal{D} ; the problem with this approach is that even when P is relatively small, say 7 or 13, as in the model problems introduced in Sections 1.1 and 2.4, respectively, the computational complexity of the algorithm becomes too huge: the number of points in a uniform grid of \mathcal{D} has an exponential growth with P for a fixed grid diameter.

In many cases the extent to which different parameters influence the field distribution and the outputs of interest is quite inhomogeneous. In this case we would be more interested to have more extensive parameter variation over the range of the "more important" parameters.

We also can just simply choose N_{pool} random parameter points from \mathcal{D} and then proceed with the sample selection algorithm as described in this Section. Such a "random" approach most likely will yield a slightly worse result than the "mesh" approach, but sometimes the former is the only possible choice, since otherwise the computational burden is too big.

We conclude the discussion of the greedy algorithm by stating that our error estimation pro-

cedures remain rigorous in all cases, regardless of the choice of (2.42) versus (2.43) or the way we select S_{pool} .

2.3.5 Off–line/On–line Procedure

We now present the algorithm for the off line/on–line computational procedure. During the off–line stage we make all the necessary preparations for the on–line stage, i.e. we precompute and store all the required vectors and matrices so that during the on–line we could calculate the reduced basis approximation and the required error bound bound in a fast manner.

Reduced Basis Approximation

We use (2.16), the representation of $u_N(\mu)$, to rewrite (2.14) in the matrix-vector form as follows:

$$\underline{A}_N(\mu) \underline{u}_N(\mu) = \underline{F}_N, \quad (2.44)$$

where

$$A_{Nij}(\mu) = a(\zeta_i, \zeta_j; \mu), \quad i, j = 1, \dots, N, \quad (2.45)$$

$$F_{Ni} = f(\zeta_i), \quad i = 1, \dots, N. \quad (2.46)$$

We then invoke (2.11), the affine decomposition of $a(., .; .)$, to obtain

$$\underline{A}_N(\mu) = \sum_{q=1}^Q \Theta_q(\mu) \underline{A}_N^q(\mu), \quad (2.47)$$

where

$$\underline{A}_{Nij}^q = a^q(\zeta_i, \zeta_j), \quad i, j = 1, \dots, N, q = 1, \dots, Q. \quad (2.48)$$

We note that in the decomposition (2.47) \underline{A}_N^q are $N \times N$ matrices and \underline{F}_N is an N -dimensional vector. During the on–line stage it takes $O(QN^2)$ operations to assemble the linear system (2.44) and $O(N^3)$ operations to solve it.

Our output approximation is then given by

$$s_N(\mu) = \underline{u}_N(\mu)^T \underline{L}_N, \quad (2.49)$$

Clearly, for the compliant output $\underline{F}_N = \underline{L}_N$.

Error Bound

We are going to focus on the construction of $\Delta_N(\mu)$ only since the error bounds for the linear outputs of interest can be easily obtained once we have the former using (2.35), (2.24), (2.41) or the duality approach [49] as it was discussed in Section 2.3.3.

We rewrite (2.19), the expression for $R(v; \mu)$, as

$$R(v; \mu) = f(v) - \sum_{q=1}^Q \sum_{i=1}^N \Theta_q(\mu) u_{N_i}(\mu) a^q(\zeta_n, v); \quad (2.50)$$

renumbering (2.50) we obtain

$$R(v; \mu) = \sum_{j=1}^K \tau_j(\mu) \Upsilon_j(v), \quad (2.51)$$

where $K = 1 + QN$ and

$$\tau_1(\mu) = 1, \quad \Upsilon_1(v) = f(v); \quad (2.52)$$

for $j = 2, \dots, 1 + QN$

$$j = 1 + (q - 1)N + l, \tau_j(\mu) = -\Theta_q(\mu) u_{N_l}(\mu), \Upsilon_j(v) = a^q(\zeta_l, v), \quad q = 1, \dots, Q, l = 1, \dots, N. \quad (2.53)$$

We can now write the expression for $\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu)$ from (2.27) as

$$\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu) = \sum_{j,j'=1}^K \tau_j(\mu) \tau_{j'}(\mu) \Lambda_{jj'}, \quad (2.54)$$

where

$$\Lambda_{jj'} = \hat{a}(z_j, z_{j'}; \mu), \quad (2.55)$$

$$\hat{a}(z_j, v; \mu) = f(v), \quad \forall v \in Y. \quad (2.56)$$

The expression for the error bound $\Delta_N(\mu)$ is then obtained from (2.32) as

$$\Delta_N(\mu) = \sqrt{\frac{\sum_{j,j'=1}^K \tau_j(\mu)\tau_{j'}(\mu)\Lambda_{jj'}}{\alpha(\mu)}}. \quad (2.57)$$

We thus state that the off-line stage requires precomputation of $\Lambda_{jj'}$ which involves $K^2 = O(Q^2N^2)$ solutions of $\mathcal{N} \times \mathcal{N}$; in fact, $\Lambda_{jj'}$ is symmetric so the number of solutions can be reduced roughly by a factor of 2. From (2.57) we observe that construction of the error bound $\Delta_N(\mu)$ requires $K^2 = O(Q^2N^2)$ operations provided that $\Lambda_{jj'}$ has been precomputed.

Numerical Algorithm

Off-line

1. Generate S_N as described in Section 2.3.4 and construct the basis for W_N ;
2. Precompute \underline{F}_N as in (2.46) and \underline{A}_N^q as in (2.48), $q = 1, \dots, Q$;
3. Precompute $\Lambda_{jj'}$ as in (2.55), $j, j' = 1, \dots, 1 + QN$.

On-line

1. Form \underline{A}_N as in (2.47) and solve (2.44) for $\underline{u}_N(\mu)$. Cost: $O(QN^2) + O(N^3)$;
2. Construct the error bound $\Delta_N(\mu)$ from (2.57). Cost: $O(Q^2N^2)$.

Total on-line cost: $O(QN^2) + O(N^3) + O(Q^2N^2)$.

2.4 Model Problem: 2D Thermal Fin

2.4.1 Problem Statement

We are going to study a variation of the model problem we introduced in Section 1.1; a similar problem was earlier studied in [49]. We will use this model problem to illustrate application of the classical reduced basis method by providing concrete numerical results.

We consider a 2D thermal fin which is schematically depicted in Figure 2-2; the fin is charac-

terized by 13 parameters ($P=13$) which constitute a 13-dimensional vector,

$$\mu = \left(\underbrace{k_1}_{\mu_1}, \underbrace{k_2}_{\mu_2}, \underbrace{k_3}_{\mu_3}, \underbrace{k_4}_{\mu_4}, \underbrace{\text{Bi}}_{\mu_5}, \underbrace{\beta_1}_{\mu_6}, \underbrace{\beta_2}_{\mu_7}, \underbrace{\beta_3}_{\mu_8}, \underbrace{\beta_4}_{\mu_9}, \underbrace{\alpha_1}_{\mu_{10}}, \underbrace{\alpha_2}_{\mu_{11}}, \underbrace{\alpha_3}_{\mu_{12}}, \underbrace{\alpha_4}_{\mu_{13}} \right),$$

where $k_i, i = 1, \dots, 4$ is the conductivity of the i^{th} sub-fin ($\tilde{\Omega}_i$), Bi is the Biot number, a non-dimensional heat transfer coefficient reflecting convective transport to the air at the fin surfaces, $\beta_i, \alpha_i, i = 1, \dots, 4$ are the length and the thickness of the i^{th} sub-fin, respectively, as shown in Figure 2-2. All the geometric parameters in this model problem are measured relative to the post width. The total height of the fin is equal to $H = 3 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$, since the height of the sections between the sub-fins is fixed at 0.75. For our parameter domain we choose $\mathcal{D} = [0.1, 10.0]^4 \times [0.01, 1.0] \times [2, 3]^4 \times [0.1, 0.5]^4$, that is, $0.1 \leq k_i \leq 10.0, i = 1, \dots, 4$ for the conductivities, $0.01 \leq \text{Bi} \leq 1, 2 \leq \beta_i \leq 3, 0.1 \leq \alpha_i \leq 0.5, i = 1, \dots, 4$ for the geometric parameters. We are primarily interested in the norm of the error $\|e_N(\mu)\|_Y = \|u(\mu) - u_N(\mu)\|_Y$ and the error bound for it since we can express the approximation for the linear outputs through $\|e_N(\mu)\|_Y$ as it was shown in Section 2.3.3.

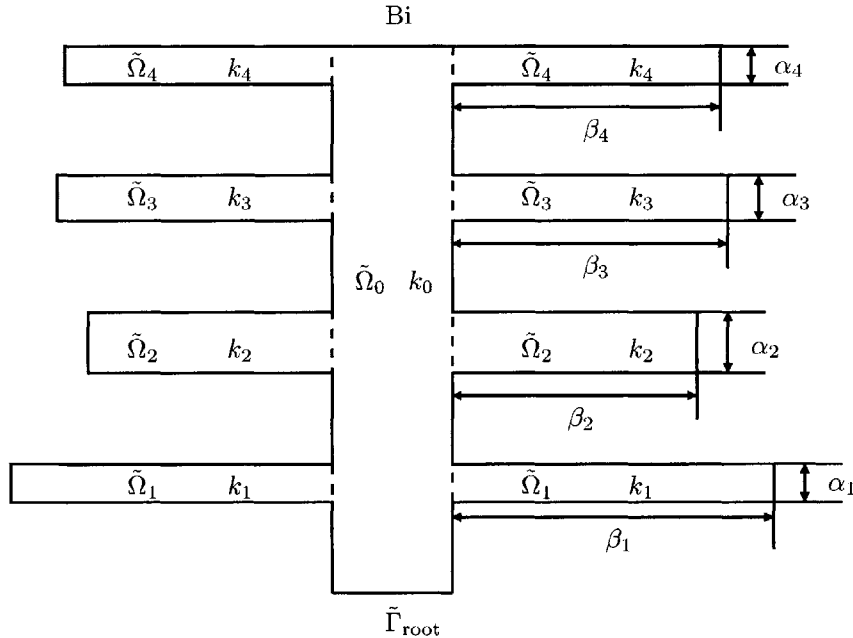


Figure 2-2: 2D affine model problem: a thermal fin with affine shape variation.

2.4.2 Governing Partial Differential Equations

Strong form

The temperature distribution $\tilde{u}(\mu)$ is obtained by solution of the following elliptic partial differential equation:

$$-k_i \nabla^2 \tilde{u}_i(\mu) = 0 \text{ in } \tilde{\Omega}_i, \quad i = 0, \dots, 4, \quad (2.58)$$

where ∇^2 is the Laplacian operator, and $\tilde{u}_i(\mu) \equiv \tilde{u}(\mu)|_{\tilde{\Omega}_i}$ refers to the restriction of $\tilde{u}(\mu)$ to $\tilde{\Omega}_i$. Here $\tilde{\Omega}_i$ is the region of the fin with conductivity k_i , $i = 0, \dots, 4$: $\tilde{\Omega}_0$ is thus the central post, and $\tilde{\Omega}_i$, $i = 1, \dots, 4$, corresponds to the four sub-fins. The use of " \sim " symbol over $u(\mu)$ signifies that the shape and the size of the thermal fin – which area we denote as $\tilde{\Omega}(\mu)$ – depend on μ . In Section 2.4.3 we describe the procedure which allows us to switch to a parameter independent reference domain Ω ; after that the symbol " \sim " will be dropped.

We must also ensure continuity of temperature and heat flux at the conductivity-discontinuity interfaces $\tilde{\Gamma}_i \equiv \partial\tilde{\Omega}_0 \cap \partial\tilde{\Omega}_i$, $i = 1, \dots, 4$, where $\partial\tilde{\Omega}_i$ denotes the boundary of $\tilde{\Omega}_i$:

$$\tilde{u}_0(\mu) = \tilde{u}_i(\mu), \quad (2.59)$$

$$-(\nabla\tilde{u}_0(\mu) \cdot \hat{n}_i) = -k_i(\nabla\tilde{u}_i(\mu) \cdot \hat{n}_i) \text{ on } \tilde{\Gamma}_i, \quad i = 1, \dots, 4; \quad (2.60)$$

here \hat{n}_i is the outward normal on $\partial\tilde{\Omega}_i$. We introduce a Neumann boundary condition on the fin root:

$$-(\nabla\tilde{u}_0(\mu) \cdot \hat{n}_o) = -1 \text{ on } \tilde{\Gamma}_{\text{root}}, \quad (2.61)$$

which models the heat source; and the Robin boundary condition:

$$-k_i(\nabla\tilde{u}_i(\mu) \cdot \hat{n}_i) = \text{Bi } \tilde{u}_i(\mu) \text{ on } \tilde{\Gamma}_{\text{ext } i}, \quad i = 0, \dots, 4, \quad (2.62)$$

which models the convective heat losses.

Weak form

Multiplying (2.58) by an arbitrary function $\tilde{v} \in \tilde{Y} = H^1(\tilde{\Omega})$ and integrating over $\tilde{\Omega}$ we obtain

$$\sum_{i=0}^4 \int_{\tilde{\Omega}_i} -k_i \nabla^2 \tilde{u}_i(\mu) \tilde{v} = 0. \quad (2.63)$$

Integrating by parts and applying the divergence theorem we rewrite (2.63) as

$$\sum_{i=0}^4 \int_{\tilde{\Omega}_i} k_i \nabla \tilde{u}(\mu) \nabla \tilde{v} - \int_{\partial \tilde{\Omega}_i} k_i \frac{\partial \tilde{u}(\mu)}{\partial \hat{n}} \tilde{v} = 0, \forall \tilde{v} \in \tilde{Y}, \quad (2.64)$$

where $\partial \tilde{\Omega}_i, i = 1, \dots, 4$ denotes the boundary of $\tilde{\Omega}_i$.

Using the continuity conditions (2.59) and (2.60) and the boundary conditions (2.61) and (2.62) we arrive at the weak form of the original partial differential equations

$$\sum_{i=0}^4 \int_{\tilde{\Omega}_i} k_i \nabla \tilde{u}(\mu) \nabla \tilde{v} + \text{Bi} \int_{\tilde{\Gamma}_{ext}} \tilde{u}(\mu) \tilde{v} = \int_{\tilde{\Gamma}_{root}} \tilde{v}, \forall \tilde{v} \in \tilde{Y}. \quad (2.65)$$

2.4.3 Reduction to Abstract Form

The formulation (2.65) is now very close to the abstract formulation of Section 2.1. The only complication – as we can observe from Figure 2-2 – is that the size and shape of the thermal fin (and hence $\tilde{\Omega}(\mu)$ and its boundary) depend on $\alpha_i, \beta_i, i = 1, \dots, 4$. For the construction of the classical reduced basis method it is essential to perform all the computations over a *fixed* reference domain Ω .

Affine Mappings

To begin we consider an arbitrary parameter dependent domain $\tilde{\Sigma}(\mu)$ – such as $\tilde{\Omega}_i, i = 1, \dots, 4$ in the model problem of this Chapter. We next introduce the parameter independent reference domain Σ and the affine mapping $\mathcal{G}(\mu)$ such that $\forall \tilde{\underline{x}} \in \tilde{\Sigma}(\mu)$ and $\forall \underline{x} \in \Sigma$

$$\underline{x} = \mathcal{G}(\tilde{\underline{x}}; \mu) = \underline{G}(\mu) \tilde{\underline{x}} + \underline{g}^r(\mu), \quad (2.66)$$

$$\Sigma = \mathcal{G}(\tilde{\Sigma}; \mu), \quad (2.67)$$

where $\underline{G}(\mu)$ is a μ -dependent 2×2 matrix and $\underline{g}^r(\mu)$ is a μ -dependent 2-dimensional vector. The boundary $\partial \tilde{\Sigma}(\mu) = \tilde{\Gamma}(\mu)$ is mapped into $\partial \Sigma = \Gamma$. It is next standard to show that $\forall \tilde{w}(\tilde{\underline{x}}), \tilde{v}(\tilde{\underline{x}}) \in H^1(\tilde{\Sigma}), \forall w(\underline{x}), v(\underline{x}) \in H^1(\Sigma)$

$$\int_{\tilde{\Sigma}(\mu)} \nabla \tilde{w} \nabla \tilde{v} d\tilde{\Sigma} = \int_{\Sigma} \left(\frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} ((G_{11}(\mu))^2 + (G_{12}(\mu))^2) + \right.$$

$$\left(\frac{\partial w}{\partial \tilde{x}_1} \frac{\partial v}{\partial \tilde{x}_2} + \frac{\partial w}{\partial \tilde{x}_2} \frac{\partial v}{\partial \tilde{x}_1} \right) (G_{11}(\mu)G_{21}(\mu) + G_{12}(\mu)G_{22}(\mu)) +$$

$$\frac{\partial w}{\partial \tilde{x}_2} \frac{\partial v}{\partial \tilde{x}_2} ((G_{21}(\mu))^2 + (G_{22}(\mu))^2) \Big|_{\underline{G}(\mu)^{-1}} d\Sigma \quad (2.68)$$

$$\int_{\tilde{\Gamma}(\mu)} \tilde{w} \tilde{v} d\tilde{\Gamma} = \int_{\Gamma} w v \Big|_{\underline{G}^{-1}(\mu)} \hat{e}_t d\Gamma, \quad (2.69)$$

where \underline{e}^t is a unit vector tangent to the boundary $\partial\Sigma$ and

$$|\underline{G}^{-1}(\mu) \underline{e}^t| = \left(\sum_{i=1}^2 (G_{ij} e_j^t)^2 \right)^{1/2} \quad (2.70)$$

Abstract Form

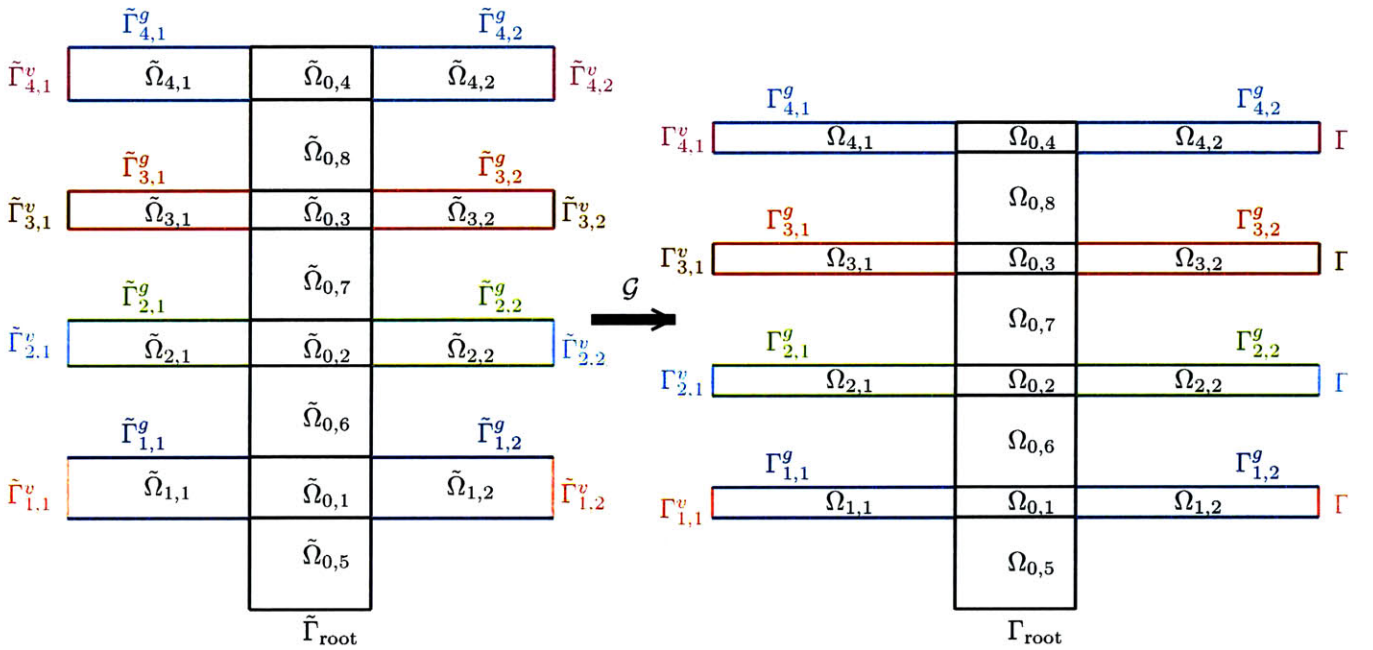


Figure 2-3: 2D affine model problem: the mapping $\mathcal{G}(\mu) : \tilde{\Omega}(\mu) \rightarrow \Omega$ between the varying shape domain $\tilde{\Omega}$ (left) and fixed reference domain Ω (right) and their respective boundaries.

We are now ready to introduce an affine geometric mapping that allows us to reformulate the problem over a parameter independent domain. We introduce a continuous mapping $\mathcal{G}(\mu) : \tilde{\Omega}(\mu) \rightarrow \Omega$ which is schematically depicted in Figure 2-3. Each of the 16 sub-domains of $\tilde{\Omega}(\mu)$: $\tilde{\Omega}_{0,i}, i = 1, \dots, 8, \tilde{\Omega}_{j,k}, j = 1, \dots, 4, k = 1, 2$ is mapped into $\Omega_{0,i}, i = 1, \dots, 8, \Omega_{j,k}, j = 1, \dots, 4, k = 1, 2$, respectively. The height and width of each sub-fin ($\Omega_{j,k}$) in the reference domain Ω is equal to α_{ref}

and β_{ref} .

We next note that \mathcal{G} is comprised of 16 different local sub-mappings \mathcal{G}^i corresponding to the 16 sub-domains as shown in Figure 2-3 where each \mathcal{G}^i admits decomposition (2.66); we also observe that the vector $\underline{g}^r(\mu)$ from (2.66) does not enter into (2.68) and (2.69). However, the correct choice of $\underline{g}^r(\mu)$ is necessary to ensure the continuity of the global mapping \mathcal{G} . In (2.71), (2.72), and (2.73) we give the description of the mapping \mathcal{G} by providing a matrix \underline{G}^k for every $\mathcal{G}^k, k = 1, \dots, 16$

$$\tilde{\Omega}_{0,i} \rightarrow \Omega_{0,i}, i = 1, \dots, 4, k = 1, \dots, 4$$

$$\underline{G}^k(\mu) = \begin{bmatrix} 1 & 0 \\ 0 & \frac{\alpha_{ref}}{\alpha} \end{bmatrix} \quad (2.71)$$

$$\tilde{\Omega}_{0,i} \rightarrow \Omega_{0,i}, i = 5, \dots, 8, k = 5, \dots, 8$$

$$\underline{G}^k(\mu) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.72)$$

$$\tilde{\Omega}_{i,j} \rightarrow \Omega_{i,j}, i = 1, \dots, 4, j = 1, 2, k = 9, \dots, 16$$

$$\underline{G}^k(\mu) = \begin{bmatrix} \frac{\beta_{ref}}{\beta_i} & 0 \\ 0 & \frac{\alpha_{ref}}{\alpha_i} \end{bmatrix} \quad (2.73)$$

$\tilde{\Gamma}(\mu)$, the boundary of $\tilde{\Omega}(\mu)$ is translated into Γ , the boundary of Ω as it is shown in Figure 2-3. Invoking (2.65), (2.68), (2.69), (2.71), (2.72), and (2.73), we recover the affine decomposition for the model problem:

$$a(w, v; \mu) = \int_{\Omega_0} \nabla w \nabla v + \text{Bi} \int_{\Gamma_{ext} \cap \partial\Omega_0 \setminus \Gamma_{root}} wv + \sum_{i=1}^4 \left(\frac{\alpha_i}{\alpha_{ref}} \int_{\Omega_{0,i}} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\alpha_{ref}}{\alpha_i} \int_{\Omega_{0,i}} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} + \right. \\ \left. k_i \frac{\beta_{ref} \alpha_i}{\alpha_{ref} \beta_i} \int_{\Omega_i} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + k_i \frac{\alpha_{ref} \beta_i}{\beta_{ref} \alpha_i} \int_{\Omega_i} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} + \text{Bi} \frac{\beta_i}{\beta_{ref}} \int_{\Gamma_i^g} wv + \text{Bi} \frac{\alpha_i}{\alpha_{ref}} \int_{\Gamma_i^v} wv \right), \quad (2.74)$$

where $\Omega_i = \Omega_{i,1} \cup \Omega_{i,2}, i = 1, \dots, 4$ and Γ_i^h (Γ_i^v) is the horizontal (vertical) part of the exterior boundary of Ω_i , respectively. Thus, Q , the total number of affine parameters, is equal to 26.

2.4.4 Numerical Results

To obtain the “truth” solution $u(\mu)$ of this model problem we consider a finite element mesh that consists of 21802 points. We note that all inputs, i.e. all entries of the parameter point μ (arguably with the exception of $\alpha_i, i = 1, \dots, 4$) exhibit meaningful influence on $u(\mu)$ which is illustrated in Table 2.1. The data in Table 2.1 is organized in the following way: we first choose $\mu^* =$

μ^*	μ^{**}	k_1	k_2	k_3	k_4	Bi	β_1	β_2	β_3	β_4	α_1	α_2	α_3	α_4
6.89	0.69	5.44	5.69	5.85	5.93	0.75	6.22	6.27	6.30	6.31	6.88	6.88	6.88	6.88

Table 2.1: 2D affine model problem: $u_{\text{root}}(\mu)$ variation.

$\{0.1, 0.1, 0.1, 0.1, 0.01, 2, 2, 2, 2, 0.1, 0.1, 0.1, 0.1\}$, $\mu^{**} = \{10, 10, 10, 10, 1, 3, 3, 3, 3, 0.5, 0.5, 0.5, 0.5\}$, $u_{\text{root}}(\mu^*)$, $u_{\text{root}}(\mu^{**})$ are given in the first two columns of Table 2.1. We next construct P parameter points $\tilde{\mu}_i, i = 1, \dots, P$ such that all the entries of $\tilde{\mu}_i$ are equal to the entries of μ^* except for the i^{th} entry which, in turn, is equal to the i^{th} entry of μ^{**} . The values for all $u_{\text{root}}(\tilde{\mu}_i), i = 1, \dots, P$ are presented in the subsequent columns of Table 2.1. We observe that the smallest output variation corresponds to $\alpha_i, i = 1, \dots, 4$ whereas for all other parameters we are able to achieve at least a 10% change in the output.

In Figure 2-4 we plot the field solution $u(\mu)$ for four different values of μ .

We then randomly generate a pool of points $S_{\text{pool}}, N_{\text{pool}} = 100000$, from which we choose N points to form S_N as described in Section 2.3.4. After that we build the off-line/on-line framework. We next test the convergence and the accuracy of our reduced basis approach.

Convergence

As described in Section 2.3.3, we introduce a bound conditioner $\hat{a}(\cdot, \cdot; u)$ that is equal to

$$\hat{a}(w, v) = \sum_{q=1}^Q \bar{\Theta}_q a^q(w, v) =$$

$$\int_{\Omega_0} \nabla w \nabla v + 0.01 \int_{\Gamma_{\text{ext}} \cap \partial \Omega_0 \setminus \Gamma_{\text{root}}} wv + 0.4 \int_{\bigcup_{i=1}^4 \Omega_{0,i}} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + 0.5 \int_{\bigcup_{i=1}^4 \Omega_{0,i}} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} +$$

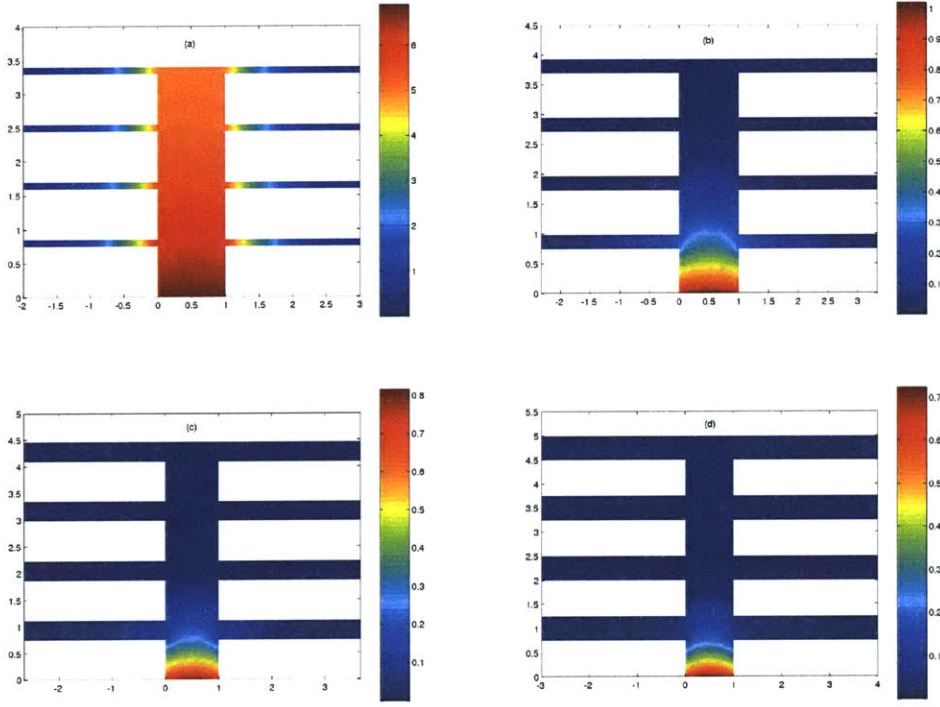


Figure 2-4: 2D affine model problem: the field solution $u(\mu)$ for
(a) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10\}$,
(b) $\mu = \{3.40, 3.40, 3.40, 3.40, 0.34, 2.33, 0.23, 2.33, 0.23, 2.33, 0.23, 2.33, 0.23\}$,
(c) $\mu = \{6.70, 6.70, 6.70, 6.70, 0.67, 2.67, 0.37, 2.67, 0.37, 2.67, 0.37, 2.67, 0.37\}$,
(d) $\mu = \{10.00, 10.00, 10.00, 10.00, 1.00, 3.00, 0.50, 3.00, 0.50, 3.00, 0.50, 3.00, 0.50\}$.

$$0.0267 \int_{\bigcup_{i=1}^4 \Omega_i} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + 0.05 \int_{\bigcup_{i=1}^4 \Omega_i} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} + 0.01 \int_{\bigcup_{i=1}^4 \Gamma_i^g} wv + 0.04 \int_{\bigcup_{i=1}^4 \Gamma_i^v} wv, \quad (2.75)$$

where $\bar{\Theta}_q = \min_{\mu \in \mathcal{D}} \Theta^q(\mu)$, $q = 1, \dots, Q$. We note that this bound conditioner is independent of μ , the description of the more sophisticated bound conditioners could be found in [52, 57]. From (2.74) we can easily see that $\hat{a}(\cdot, \cdot)$ satisfies (2.22) and hence is suitable for our purposes. For this choice of bound conditioner the constants – as given in (2.22) – assume the following values: $\alpha_0 = 0.0231$, $\rho_{min} = 1$, $\rho_{max} = 750$, $\gamma_0 = 37.4554$.

All the numerical results – unless specifically indicated otherwise – are averaged over a sample set $S_{test} = \{\mu_1^t, \dots, \mu_{test}^t\}$ of cardinality $N_{test} = 100000$ which provides some level of statistical significance for our test and because of that we drop the dependence in μ for all quantities we consider. In Figure 2-5 we present the dependencies of $\|e_N\|$ and Δ_N on N , whereas in Figure 2-6 we provide the same dependencies for $a(e_N, e_N)$ and $\tilde{\Delta}_N$. As we observe from Figures 2-5 and 2-6,

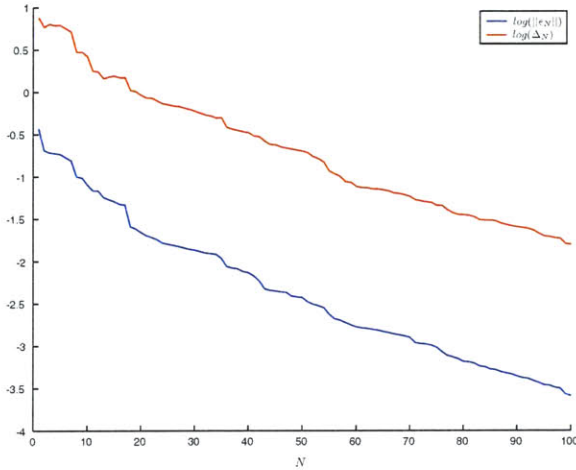


Figure 2-5: Classical reduced basis method, 2D affine model problem: $\|e_N\|_Y$ and Δ_N as functions of N .

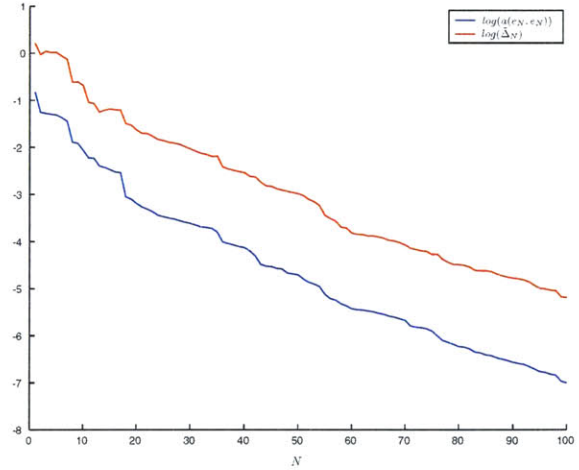


Figure 2-6: Classical reduced basis method, 2D affine model problem: $a(e_N, e_N)$ and $\tilde{\Delta}_N$ as functions of N .

all of the following quantities: $\|e_N\|$, Δ_N , $a(e_N, e_N)$, and $\tilde{\Delta}_N$ decrease with N . In Table 2.2 we provide the rate of relative convergence, i.e. $\frac{\Delta_N}{\|u\|_Y}$, $\frac{\|e_N\|_Y}{\|u\|_Y}$ and $\frac{a(e_N, e_N)}{a(u, u)}$, $\frac{\tilde{\Delta}_N}{a(u, u)}$ with respect to N . At $N = 36$ $\frac{\|e_N\|_Y}{\|u\|_Y}$ reaches the 1% accuracy which is already a good approximation in the engineering practice. At the same time the best level of relative accuracy in Δ_N we achieve is equal to 1.54%. To reach better accuracy we need to further increase N . As we stated earlier, this problem depends in an essential manner on all the inputs – thus the rate of convergence is worse than for a similar problem as described in [49] since our problem has a "richer" parameter dependence.

Effectivity

We are also interested in measuring the sharpness of our error prediction. We first provide the bounds for the effectivities $\eta_N(\mu)$. As given by (2.39) and (2.40) $\eta_N(\mu)$ is bounded from below by the value of 1 and from above by $\sqrt{\frac{\rho_{max} \gamma_0}{\alpha_0}} = 1101.8298$.

In Figure 2-7 we present the plot for effectivities η_N , $\tilde{\eta}_N$ as functions of N . We note that both η_N and $\tilde{\eta}_N$ do not display any significant dependence on N and stays in the range of [25,70] which is significantly lower than the theoretical upper bound for the effectivity; this ensures that our error bounds are indeed reasonably sharp. In general, it is possible to improve the effectivity even further by choosing more sophisticated bound conditioners [52, 57].

N	$\frac{\ e_N\ _Y}{\ u\ _Y}$	$\frac{\Delta_N}{\ u\ _Y}$	$\frac{a(e_N, e_N)}{a(u, u)}$	$\frac{\tilde{\Delta}_N}{a(u, u)}$
5	1.60e-01	5.72e+00	4.38e-02	9.89e-01
10	7.40e-02	2.45e+00	8.36e-03	1.97e-01
15	4.75e-02	1.48e+00	3.31e-03	6.65e-02
20	2.07e-02	8.70e-01	6.30e-04	2.32e-02
25	1.53e-02	6.72e-01	3.37e-04	1.36e-02
30	1.31e-02	5.74e-01	2.47e-04	9.44e-03
35	1.05e-02	4.83e-01	1.64e-04	6.71e-03
40	7.07e-03	3.21e-01	7.54e-05	2.98e-03
45	4.39e-03	2.30e-01	3.07e-05	1.51e-03
50	3.70e-03	1.96e-01	2.09e-05	1.08e-03
55	2.33e-03	1.14e-01	8.03e-06	3.77e-04
60	1.64e-03	7.32e-02	3.81e-06	1.50e-04
65	1.44e-03	6.68e-02	3.06e-06	1.24e-04
70	1.25e-03	5.64e-02	2.25e-06	8.75e-05
75	9.60e-04	4.48e-02	1.35e-06	5.56e-05
80	6.37e-04	3.41e-02	5.99e-07	3.25e-05
85	5.21e-04	2.98e-02	4.06e-07	2.50e-05
90	4.30e-04	2.49e-02	2.81e-07	1.74e-05
95	3.38e-04	1.94e-02	1.77e-07	1.06e-05
100	2.49e-04	1.54e-02	1.01e-07	6.67e-06

Table 2.2: Classical reduced basis method, 2D affine model problem: relative convergence of $\|e_N\|$, Δ_N and $a(e_N, e_N)$, $\tilde{\Delta}_N$ with respect to N .

Computational Costs

During the off-line stage the calculation of $\Lambda_{jj'}$ as given in (2.55) presents the biggest computational challenge as we need to perform roughly $Q^2 N^2$ of solves of $\mathcal{N} \times \mathcal{N}$ linear systems of equations. The time required to run the off-line stage depends mostly on the "raw" power of the computer; it requires several weeks to complete this stage on a modern PC. In the next Chapter we propose a new partition of unity method which allows us to reduce the off-line computational time significantly by decomposing the error estimate into a sum of the local contributions from smaller subdomains $\mathcal{P}_i \subset \Omega, i = 1, \dots, S$. With this approach there is no longer a need to solve the $\mathcal{N} \times \mathcal{N}$ system – we only need to solve systems which are equal in size to the number of nodes in these subdomains \mathcal{P}_i .

We now turn to the issue of the actual computational efficiency of the classical reduced basis method during the on-line stage. In Table 2.3 we present the dependencies of t_{u_N} , the time required

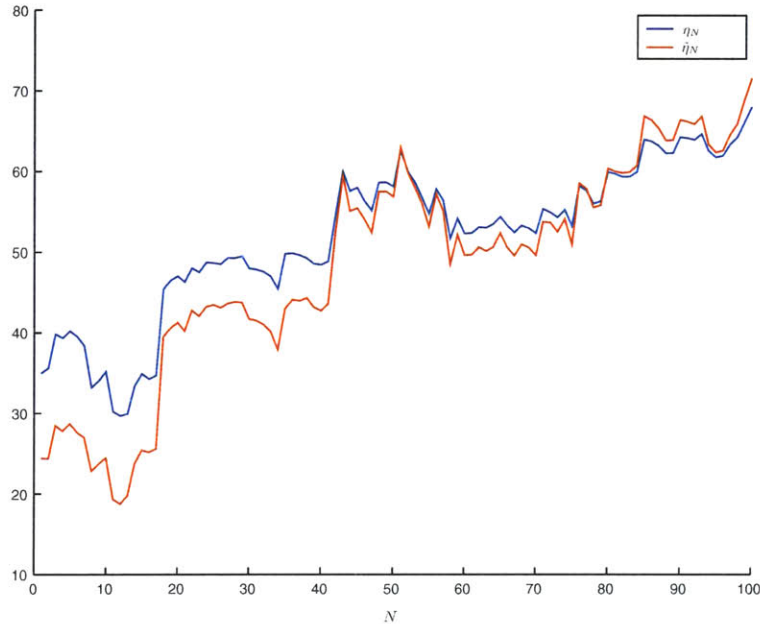


Figure 2-7: Classical reduced basis method, 2D affine model problem: effectivities η_N and $\tilde{\eta}_N$ as functions of N .

to obtain \underline{u}_N , t_{Δ_N} , the time required to obtain Δ_N , and $t_{tot} = t_{u_N} + t_{\Delta_N}$, the total time for the on-line stage as functions of N . In the rightmost column of Table 2.3 we give the value of t_{tr} , the time that is necessary to obtain the truth solution $u(\mu)$ which is obviously independent of N .

As we observe from Table 2.3, both t_{u_N} , t_{Δ_N} increase with N which is consistent with their operation counts, $O(QN^2) + O(N^3)$ and $O(Q^2N^2)$, respectively. We also note that t_{Δ_N} dominates t_{u_N} and hence the former provides the main contribution to t_{tot} ; this fact is partly explained by a relatively large value of Q ; it also relates to the complexity of precomputing $\Lambda_{jj'}$ during the off-line stage.

In Chapter 3 we present a version of the partition of unity method which can be applied to affine problems. With this new method we attempt to reduce this Q^2N^2 dependence of the operation count both for the off-line and the on-line stage.

In general, even for the highest value of $N = N_{max}$ ($N_{max} = 100$) the reduced basis method is still approximately 25 times more efficient than the finite element method (we measure the efficiency by considering the ratio $\frac{t_{tr}}{t_{tot}}$). All the computations are performed in Matlab where matrix-vector operations are performed much faster than the user defined functions; this works against the computational efficiency of the classical reduced basis method. If the user chooses a different programming environment – such as for example C/C++ – we expect computational

N	t_{u_N}, s	t_{Δ_N}, s	t_{tot}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tr}, s
5	1.86e-03	3.68e-04	2.23e-03	1582.45	3.53
10	2.14e-03	5.78e-03	7.92e-03	445.26	3.53
15	2.16e-03	3.11e-03	5.27e-03	669.48	3.53
20	2.58e-03	5.42e-03	8.00e-03	440.79	3.53
25	3.12e-03	7.81e-03	1.09e-02	322.73	3.53
30	3.39e-03	1.10e-02	1.44e-02	245.21	3.53
35	3.88e-03	1.43e-02	1.82e-02	193.61	3.53
40	4.62e-03	1.82e-02	2.28e-02	154.68	3.53
45	5.11e-03	2.31e-02	2.82e-02	124.86	3.53
50	5.99e-03	2.86e-02	3.46e-02	101.91	3.53
55	7.47e-03	3.37e-02	4.12e-02	85.61	3.53
60	8.17e-03	4.14e-02	4.95e-02	71.19	3.53
65	1.13e-02	4.82e-02	5.95e-02	59.25	3.53
70	1.26e-02	5.53e-02	6.79e-02	51.90	3.53
75	1.47e-02	6.32e-02	7.79e-02	45.23	3.53
80	1.76e-02	6.98e-02	8.74e-02	40.35	3.53
85	1.96e-02	7.95e-02	9.91e-02	35.57	3.53
90	2.37e-02	9.06e-02	1.14e-01	30.84	3.53
95	2.83e-02	1.00e-01	1.28e-01	27.48	3.53
100	3.24e-02	1.11e-01	1.43e-01	24.57	3.53

Table 2.3: Classical reduced basis method, 2D affine model problem: on-line computational times.

savings to be even more substantial.

Conclusions

Summarizing the the theoretical and numerical results we presented in this Chapter, we can state that even for the problem of relatively high affine complexity the classical reduced basis method fulfills the goals set in Section 1.3.1; it provides a precise approximation for the finite element solution with certified accuracy; the error bound is sharp and at the same time we achieve reasonable computational savings. We expect the computational savings to be even more significant for different choices of programming environment.

Chapter 3

Locally Non-affine Coercive Problems: Partition of Unity Method

3.1 General Abstract Problem Statement

As in Chapter 2, we consider a suitably regular (smooth) parameter-independent domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, or 3 , and associated function space $Y \subset (H^1(\Omega))^p$ with inner product $(\cdot, \cdot)_Y$, norm $\|\cdot\|_Y = (\cdot, \cdot)_Y^{1/2}$, and dual space Y' ; as before, we define a parameter set $\mathcal{D} \subset \mathbb{R}^P$, a particular point in which will be denoted μ .

We now consider the bilinear form $a(\cdot, \cdot; \mu) : Y^2 \rightarrow \mathbb{R}$ corresponding to the underlying second-order partial differential equations. We assume that $a(\cdot, \cdot; \cdot)$ is continuous: there exists $\gamma(\mu) > 0$ such that

$$a(w, v; \mu) \leq \gamma(\mu) \|w\|_Y \|v\|_Y \leq \gamma_0 \|w\|_Y \|v\|_Y, \quad \forall \mu \in \mathcal{D}; \quad (3.1)$$

furthermore, we assume that $a(\cdot, \cdot; \cdot)$ is coercive: there exists $\alpha(\mu) > 0$ such that

$$0 < \alpha_0 \leq \alpha(\mu) = \inf_{v \in Y} \frac{a(v, v; \mu)}{\|v\|_Y^2}, \quad \forall \mu \in \mathcal{D}, \quad (3.2)$$

and symmetric, $a(w, v; \mu) = a(v, w; \mu)$, $\forall w, v \in Y$, $\forall \mu \in \mathcal{D}$; we also introduce the bounded linear functionals $f(\cdot) \in Y'$ and $l(\cdot) \in Y'$. We shall again make certain assumptions on the parametric dependence of $a(\cdot, \cdot; \mu)$, $f(\cdot)$, and $l(\cdot)$. In particular, we shall suppose that $f(\cdot)$ and $l(\cdot)$ are parameter

independent and $a(\cdot, \cdot; \mu)$ can be expressed in the following way

$$a(w, v; \mu) = a^A(w, v; \mu) + a^{NA}(w, v; \mu), \quad \forall w, v \in Y, \quad \forall \mu \in \mathcal{D}, \quad (3.3)$$

where $a^A(\cdot, \cdot; \mu)$ is a continuous and affine bilinear form as previously described in Section 2.1. Consistent with the definition (2.11) of the previous Chapter, $a^A(\cdot, \cdot; \mu)$ may be expressed as

$$a^A(w, v; \mu) = \sum_{q=1}^Q \Theta_q(\mu) a^q(w, v) \quad \forall w, v \in Y, \quad \forall \mu \in \mathcal{D}, \quad (3.4)$$

where the $\Theta^q(\mu): \mathcal{D} \rightarrow \mathbb{R}$ is a function and $a^q(\cdot, \cdot): Y^2 \rightarrow \mathbb{R}$ is a bilinear form without any parametric dependence.

For $a^{NA}(\cdot, \cdot; \cdot)$ we allow general (unrestricted, non-affine) parametric dependence. The only limitation we impose is that this kind of general dependence can only appear in the area $\Omega_{na} \subset \Omega$ and the size of the former is small compared to the size of the latter. More specifically, $\forall v, w \in Y : v|_{\Omega_{na}} = 0, w|_{\Omega_{na}} = 0$

$$a^{NA}(w, v; \mu) \equiv 0, \quad \forall \mu \in \mathcal{D}. \quad (3.5)$$

The assumptions of the affine parameter dependence and the “local” nature of the non-affine dependence are crucial for the computational efficiency and rigorous error estimation of the partition of unity method that we are going to present in this Chapter.

Our abstract problem statement is then: find $u(\mu) \in Y$ such that

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in Y. \quad (3.6)$$

The output of interest $s(\mu)$ is given by

$$s(\mu) = l(u(\mu)). \quad (3.7)$$

We refer to the particular case when $l(\cdot) = f(\cdot)$ as “compliant”.

As before, (3.6) is our partial differential equation (in weak form), μ is our parameter, $u(\mu)$ is our field solution, and $s(\mu)$ is our output.

We will refer to the problems that are given by (3.6) and admit the decomposition which is

consistent with (3.3), (3.4), and (3.5) as locally non-affine.

In actual practice, Y is replaced by an appropriate “truth” finite element approximation space $Y_{\mathcal{N}}$ of dimension \mathcal{N} defined on a suitably fine truth mesh. We then approximate $u(\mu)$ and $s(\mu)$ by $u_{\mathcal{N}}(\mu)$ and $s_{\mathcal{N}}(\mu)$, respectively, and assume that $Y_{\mathcal{N}}$ is sufficiently rich such that $u_{\mathcal{N}}(\mu)$ and $s_{\mathcal{N}}(\mu)$ are indistinguishable from $u(\mu)$ and $s(\mu)$, respectively.

The number of basis functions, \mathcal{N} of Y is equal to the number of nodes in the finite element discretization of Ω . We denote the number of nodes in Ω_{na} as n_{na} . Since the size of Ω_{na} is small compared Ω the same applies to \mathcal{N} with respect to n_{na} . A priori we will require n_{na} to be no bigger than $\frac{\mathcal{N}}{10}$. Later we will confirm the validity of this assumption from the viewpoint of the efficiency of our computational procedure.

3.1.1 Failure of Classical Reduced Basis Method

It is now time to demonstrate the pitfalls of the classical reduced basis approach which was described in Chapter 2.

As previously discussed, we start by introducing the sample $S_N = \{\mu_1, \dots, \mu_N\}$ and the reduced basis function space $W_N = \text{span}\{\zeta_1 = u(\mu_1), \dots, u(\mu_N)\}$. We look for the reduced basis approximation $u_N(\mu)$ such that

$$a(u_N(\mu), v; \mu) = f(v), \quad \forall v \in W_N, \quad (3.8)$$

$$u_N(\mu) = \sum_{i=1}^N u_{N_i}(\mu) \zeta_i. \quad (3.9)$$

Using the decomposition (3.9) we rewrite (3.8) in matrix-vector form as follows

$$\underline{A}_N(\mu) \underline{u}_N(\mu) = \underline{F}_N, \quad (3.10)$$

where $F_{N_i} = f(\zeta_i)$, $i = 1, \dots, N$ and

$$\underline{A}_N(\mu) = \underbrace{\sum_{q=1}^Q \Theta_q(\mu) \underline{A}_N^q}_{\text{affine part}} + \underbrace{\underline{A}_N^{na}(\mu)}_{\text{non-affine part}}, \quad (3.11)$$

where $\underline{A}_{N_{ij}}^q = a^q(\zeta_i, \zeta_j)$, $\underline{A}_N^{na} = a^{NA}(\zeta_i, \zeta_j; \mu)$, $i, j = 1, \dots, N$, $q = 1, \dots, Q$.

For now we skip the off-line stage and address the on-line stage directly. Given a new value of

μ we first need to assemble the left hand side of (3.8) which is decomposed as shown in (3.11). The assembly of the affine part of (3.11) requires (QN^2) operations. However, we cannot precompute \underline{A}_N^{na} in the same manner. Instead, we first have to form the finite element matrix \underline{A}^{na} corresponding to Ω_{na} , $\underline{A}_{ij}^{na} = a^{NA}(\phi_i, \phi_j)$, $i, j = 1, \dots, \mathcal{N}$; here $\phi_i, i = 1, \dots, \mathcal{N}$ is a finite element test function of $Y_{\mathcal{N}}$. We only have to choose ϕ_i which have non-zero support in Ω_{na} , hence this assembly requires $O(n_{na})$ operations. We then project $\underline{A}^{na}(\mu)$ onto the reduced basis space W_N to calculate \underline{A}_N^{na} as $\underline{A}_{Nij}^{na} = a^{NA}(\zeta_i, \zeta_j; \mu)$, $i, j = 1, \dots, N$, which requires $O(n_{na}N^2)$ operations. Finally, we obtain $\underline{u}_N(\mu)$ by solving (3.10). This requires $O(N^3)$ operations.

We readily note that the total operation count to obtain the reduced basis approximation $u_N(\mu)$ equals to $O(QN^2) + O(n_{na}N^2) + O(N^3)$ and has linear dependence in n_{na} . This is very close to the maximum price we are willing to pay for the reduced basis approximation because n_{na} is a fraction of \mathcal{N} , the total number of finite element nodes, which, in turn, is typically a very large number. Therefore, if the operation count for the reduced basis approximation scales as n_{na}^β , where β is not in the close vicinity of 1 – say, $\beta = 2$ – we can no longer expect any computational savings and the reduced basis method loses its attractiveness. We are going to return to this issue later in this Section.

We now turn to the evaluation of the error bound $\tilde{\Delta}_N(\mu)$ such that

$$a(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_N(\mu). \quad (3.12)$$

As it was demonstrated in Section (2.3.3), the construction of $\tilde{\Delta}_N(\mu)$ essentially requires evaluation of $\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu)$ where

$$\hat{a}(\hat{e}_N(\mu), v; \mu) = R(v; \mu), \quad \forall v \in Y, \quad (3.13)$$

where, in turn,

$$R(v; \mu) = f(v) - a(u_N(\mu), v; \mu). \quad (3.14)$$

In Chapter 2 we have shown that if the residual $R(v; \mu)$ admits the decomposition (2.51) of K terms which are products of parameter dependent coefficients $\tau_j(\mu), j = 1, \dots, K$ and parameter independent linear functionals $\Upsilon_j(v), j = 1, \dots, K$ the operation count for evaluation of $\hat{a}(\hat{e}_N(\mu), \hat{e}_N(\mu); \mu)$ requires K^2 operations. Thus, for the purely affine case this operation count is $(1 + QN)^2 = O(Q^2N^2)$.

However, locally non-affine problems do not readily admit a decomposition similar to (2.51) because $a(\cdot, \cdot; \mu)$ is no longer an affine form. In order to address this issue we can replace the non-affine dependence by an affine element-wise dependence in Ω_{na} . In other words, we will treat each element in Ω_{na} as an independent domain for which we extract a specific parameter dependence. Following this “naive” extension of the CRBM we immediately observe that the number of terms in the residual decomposition, $H = 1 + (Q + n_{el})N$; n_{el} is the number of elements in Ω_{na} , $n_{el} = O(n_{na})$. As we just mentioned, this implies that we will need $O((Q + n_{na})^2 N^2)$ operations to obtain the error bound. This is too much of a burden for our reduced basis method to remain efficient. Let us consider an example for which $\mathcal{N} = 10000$ and set $n_{na} = 1000$, which is consistent with our earlier assumption. The operation count will scale at least as 10^6 which is significantly more than it takes to solve the original problem directly.

In general, it takes $\mathcal{C}_{tr} = O(\mathcal{N}^\kappa)$ operations to obtain the “truth” finite element solution $u(\mu)$ where κ depends on the specifics of the problem and the solution technique. We choose Krylov subspace or analogous highly efficient iterative method [50] as our solution technique and use it as a benchmark. Typically κ lies in the range if (1,2]. At the same time the on-line operation count for the “naive” extension of the CRBM is equal to $\mathcal{C}_{rb} = O(QN^2) + O(n_{na}N^2) + O(N^3) + O((Q + n_{na})^2 N^2)$. In order for our method to remain efficient we require that $\mathcal{C}_{rb} \ll \mathcal{C}_{tr}$; this condition is much harder (or even impossible) to satisfy when we have quadratic dependence in n_{na} (recall that n_{na} is proportional to \mathcal{N} and κ in most cases is less than 2). Also the “naive” reduced basis method clearly loses its computational efficiency if there is a need to increase \mathcal{N} – which is often the case – again thanks to the debilitating quadratic dependence in n_{na} ; this effect becomes even more noticeable for lower values of κ . We conclude the critique of the “naive” method by mentioning that its operation count is also highly dependent on Q (note the cross-term $n_{na}Q$) which in turn imposes yet another restriction on the parametric complexity of the underlying problem.

Based on the arguments presented in this Section we arrive at the following conclusion: in order to fulfill the the goals of providing a computationally reliable and efficient reduced basis method we need to augment the classical reduced basis method in a different way than offered by this “naive” extension. Later in this Chapter we introduce a new partition of unity method with which we are going to address this issue.

3.2 Partition of Unity Method

3.2.1 Partition of Unity

As we previously mentioned, we obtain the “truth” solution by using the finite element discretization over Ω . Let us denote the corresponding discretization as $T_{\mathcal{N}}$ where \mathcal{N} stands for the number of nodes. In this thesis we mostly consider two-dimensional model problems and use triangular finite elements and piece-wise linear test functions. The theory we present can be extended to higher-order finite element approximation which, however, lies beyond the scope of this thesis. Typically \mathcal{N} is quite a large number ranging from several thousand to even a million.

We start by introducing a system of S piecewise linear functions $\Phi_i(\underline{x}) \in Y, i = 1, \dots, S$ – which we later refer to as the partition of unity – such that

$$\sum_{i=1}^S \Phi_i(\underline{x}) \equiv 1, \quad \forall \underline{x} \in \Omega, \quad (3.15)$$

and

$$\bar{\Omega} = \bigcup_{i=1}^S \bar{\mathcal{P}}_i, \quad (3.16)$$

where \mathcal{P}_i is the region of support of Φ_i : $\mathcal{P}_i = \text{supp } \Phi_i, i = 1, \dots, S$.

One way to introduce the partition of unity is to consider a very coarse triangulation of Ω which we call T_S . In most cases it is possible to construct T_S in such a way that S , the number of nodes in T_S is equal to $O(10)$. We show the relation between $T_{\mathcal{N}}$ and T_S for some hypothetical domain $\Omega \subset \mathbb{R}^2$ in Figure 3-1.

Furthermore, in order to maintain computational efficiency of our method we need that T_S – with some certain requirements – to be as coarse as possible. We require that $T_{\mathcal{N}}$ is a refinement of T_S . More formally it means that every finite element of the coarse $T_j, j = 1, \dots, S_{el}$ – where S_{el} is the number of elements in T_S – can be represented as a union of finite number of elements of $T_{\mathcal{N}}$.

We denote the finite element function spaces associated with T_S and $T_{\mathcal{N}}$ as Y_S and $Y_{\mathcal{N}}$, respectively. As before, we will omit the \mathcal{N} subindex in $Y_{\mathcal{N}}$ essentially referring to the fact that we make almost no distinction between $Y = H^1(\Omega)$ and $Y_{\mathcal{N}}$.

We next introduce the basis in Y_S of linear test functions $\Phi_i(\underline{x}), i = 1, \dots, S$, such that

$$\Phi_i(\underline{x}_j) = \delta_{ij}, \quad i, j = 1, \dots, S, \quad (3.17)$$

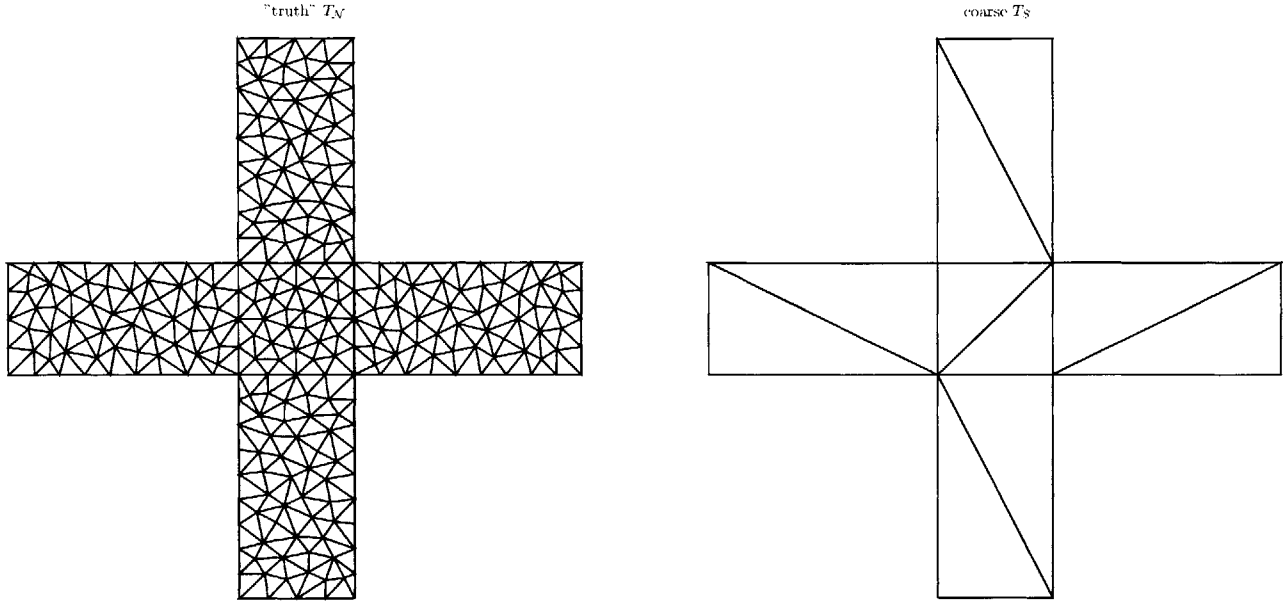


Figure 3-1: Partition of unity method: relation between T_N , the "truth" discretization, and T_S , the coarse discretization.

where \underline{x}_j is the j^{th} node of the triangulation T_S and δ_{ij} is the Kronecker's symbol. It is easy to see that $\Phi_i(\underline{x}), i = 1, \dots, S$ constitute a partition of unity of Ω in the sense of (3.15).

We note that the use of the coarse mesh is not the only way to select a partition of unity. For example, another valid approach would be to choose a set of functions which satisfy (3.15) and vary only with respect to one of the space coordinates, say x_1 .

3.2.2 Reduced Basis Approximation

We are now ready to provide the mathematical description of our new technique which we call the partition of unity method. We start by selecting a sample in parameter domain \mathcal{D}

$$S_N = \{\mu_1, \dots, \mu_N\}, \tag{3.18}$$

where $\mu_i \in \mathcal{D}, i = 1, \dots, N$. We follow one of the two the sample selection algorithms as described in Section 2.3.4.

We then define our reduced-basis approximation space W_N as

$$W_N = \text{span}\{\zeta_i \equiv u(\mu_i), i = 1, \dots, N\}, \tag{3.19}$$

where $u(\mu_i) \in Y$ is the solution to (3.6) for $\mu = \mu_i$. For any $\mu \in \mathcal{D}$, we define the reduced-basis approximation $u_N(\mu)$ which is obtained through Galerkin projection of $u(\mu)$ onto W_N

$$a(u_N(\mu), v) = f(v), \quad \forall v \in W_N. \quad (3.20)$$

From (3.20) we see that $u_N(\mu)$ can be expressed as a linear combination of basis function of W_N

$$u_N(\mu) = \sum_{j=1}^N u_{Nj}(\mu) \zeta_j. \quad (3.21)$$

We next look for $\delta_S(\mu) \in X_S$ given by

$$a(\delta_S(\mu), v; \mu) = f(v) - a(u_N(\mu), v; \mu), \quad \forall v \in X_S, \quad (3.22)$$

$$\delta_S(\mu) = \sum_{k=1}^S \delta_{S_k}(\mu) \Phi_k. \quad (3.23)$$

As we mentioned in Section 3.2.1, we choose S to be $O(10)$, hence assembly and solution of (3.22) is quite inexpensive; we discuss computational complexity in more detail in Section 3.2.4.

We now consider $\tilde{u}_N(\mu) = u_N(\mu) + \delta_S(\mu)$ as the approximation for $u(\mu)$. We next note that $\delta_S(\mu)$ does not make the accuracy of the reduced basis approximation $\tilde{u}_N(\mu)$ significantly worse than that of $u_N(\mu)$ since

$$a(\delta_S(\mu), \delta_S(\mu); \mu) = a(u(\mu) - u_N(\mu), \delta_S(\mu); \mu), \quad (3.24)$$

and hence by application of Cauchy-Schwartz' inequality

$$a(\delta_S(\mu), \delta_S(\mu); \mu) \leq a(u(\mu) - u_N(\mu), u(\mu) - u_N(\mu); \mu). \quad (3.25)$$

3.2.3 A Posteriori Error Estimation

We next present the details of the error estimation procedure which to a great extent uses the principles of the error estimation procedure described in Section 2.3.3 and the ideas presented in [20, 19]. This procedure can be easily extended to the non-compliant outputs using the relations of Section 2.3.3.

A Posteriori Error Bound for $\|e_N(\mu)\|_Y$ and Compliant Output

We start by introducing the error function $e_N(\mu) \in Y$ as

$$e_N(\mu) = u(\mu) - \tilde{u}_N(\mu). \quad (3.26)$$

We then divide all patches $\mathcal{P}_i, i = 1, \dots, S$ into two sets \mathcal{P}^a and \mathcal{P}^{na} ; $\mathcal{P}^a = \{\mathcal{P}_1^a, \dots, \mathcal{P}_J^a\}$ consists of those patches where $a(\cdot, \cdot; \mu)$ only has affine parametric dependence, whereas $\mathcal{P}^{na} = \{\mathcal{P}_1^{na}, \dots, \mathcal{P}_G^{na}\}$ is formed by the patches where $a(\cdot, \cdot; \mu)$ has non-affine parametric dependence, i.e. those $\mathcal{P}_i: \Omega_{na} \cap \mathcal{P}_i \neq \emptyset, i = 1, \dots, S$. We denote the partitions functions related to $\mathcal{P}_i^{a|na}$ as $\Phi_i^{a|na}$, Clearly, $J + G = S$.

We then introduce functional spaces $Y_i^{a|na} = \{v \in H^1(\bar{\mathcal{P}}_i^{a|na})\}$, such that the derivatives of functions v in the regularity constraints for H^1 spaces are evaluated from inside of the domain $\bar{\mathcal{P}}_i^{a|na}$ and the mapping operators $\mathcal{I}_i^{a|na} : Y \rightarrow Y_i^{a|na}$ such that

$$\forall v \in Y \quad \mathcal{I}_i^{a|na} v = \bar{v} \in Y_i^{a|na} : \bar{v} = v|_{\bar{\mathcal{P}}_i^{a|na}}. \quad (3.27)$$

We also define two sets $\{a_{\mathcal{P}_1^a}(\cdot, \cdot; \cdot), \dots, a_{\mathcal{P}_J^a}(\cdot, \cdot; \cdot)\}, \{a_{\mathcal{P}_1^{na}}(\cdot, \cdot; \cdot), \dots, a_{\mathcal{P}_G^{na}}(\cdot, \cdot; \cdot)\}$ of bilinear forms $a_{\mathcal{P}_i^{a|na}}(\cdot, \cdot; \cdot) : (Y_i^{a|na})^2 \rightarrow \mathbb{R}$; and two sets $\{f_{\mathcal{P}_1^a}(\cdot), \dots, f_{\mathcal{P}_J^a}(\cdot)\}, \{f_{\mathcal{P}_1^{na}}(\cdot), \dots, f_{\mathcal{P}_G^{na}}(\cdot)\}$ of linear functionals $f_{\mathcal{P}_i^{a|na}}(\cdot) \in (Y_i^{a|na})'$ such that

$$\forall w, v \in Y : w|_{\Omega \setminus \bar{\mathcal{P}}_i^{a|na}} = 0, v|_{\Omega \setminus \bar{\mathcal{P}}_i^{a|na}} = 0$$

$$a_{\mathcal{P}_i^{a|na}}(\mathcal{I}_i^{a|na} w, \mathcal{I}_i^{a|na} v; \mu) = a(w, v; \mu), \quad (3.28)$$

$$f_{\mathcal{P}_i^{a|na}}(\mathcal{I}_i^{a|na} v) = f(v), \quad (3.29)$$

$$i = 1, \dots, J \text{ for } \mathcal{P}_i^a \text{ and } i = 1, \dots, G \text{ for } \mathcal{P}_i^{na}.$$

We next define a function $c_i(\mu) : \mathcal{D} \rightarrow \mathbb{R}$ and a bilinear form $\hat{a}_i(\cdot, \cdot) : (Y_i^a)^2 \rightarrow \mathbb{R}$ such that $\forall \mu \in \mathcal{D}$

$$c_i(\mu) \hat{a}_i(v, v) \leq a_{\mathcal{P}_i^a}(v, v; \mu) \leq \gamma_i(\mu) \hat{a}_i(v, v) \leq \gamma_{0,i} \hat{a}_i(v, v); \quad (3.30)$$

we note that though we demand the coercivity from $a(\cdot, \cdot; \cdot)$ we do not require the same for each $a_{\mathcal{P}_i}(\cdot, \cdot; \cdot)$. Clearly, $0 \leq a_{\mathcal{P}_i}(v, v; \mu), \forall v \in Y_i^a$, however it may well happen that for some $v \in Y_i^a$

$a_{\mathcal{P}_i^a}(v, v; \mu) = 0$. If this is the case, we need to make sure that $\hat{a}_i(v, v) = 0$ as well, so that (3.30) holds.

We define "local" residual functionals $R_i^{a|na}(v; \mu) \in (Y_i^{a|na})'$ (compare to (2.19)) as follows

$$R_i^{a|na}(v; \mu) = f_{\mathcal{P}_i^{a|na}}(v) - a_{\mathcal{P}_i^{a|na}}(\mathcal{I}_i^{a|na} \tilde{u}_N(\mu), v; \mu), \quad (3.31)$$

$$i = 1, \dots, J \text{ for } \mathcal{P}_i^a \text{ and } i = 1, \dots, G \text{ for } \mathcal{P}_i^{na}.$$

The key difference between (2.19) and (3.31) is in the "local" and "global" nature of $R_i^{a|na}(\cdot)$ and $R(\cdot)$, respectively.

We subsequently look for the "local" error function $\hat{e}_{N,i}^{a|na}(\mu)$ as the solution of the "local" error equation (compare to (2.21)):

$$c_i^a(\mu) \hat{a}_i^a(\hat{e}_{N,i}^a(\mu), v) = R_i^a(\mathcal{I}_i^a \Phi_i^a v; \mu), \quad \forall v \in Y_i^a, i = 1, \dots, J, \quad (3.32)$$

$$a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), v; \mu) = R_i^{na}(\mathcal{I}_i^{na} \Phi_i^{na} v; \mu), \quad \forall v \in Y_i^{na}, i = 1, \dots, G. \quad (3.33)$$

What separates (3.32), (3.33) from (2.21) is that we apply $R_i^{a|na}(\cdot)$ to $\mathcal{I}_i^{a|na} \Phi_i^{a|na} v$ and not just to v as in (2.21) and also that $\mathcal{I}_i^{a|na} \Phi_i^{a|na} v$ is defined over $\bar{\mathcal{P}}_i^{a|na}$ only, a subdomain of Ω whereas v is defined over the whole domain Ω ; this is the place where the partition of unity comes into play.

Here t is the number of nodes in a single finite element of Ω discretization. For the further construction of the method we use the coarse mesh approach to select the partition of unity, $\Phi_i(\underline{x}), i = 1, \dots, S$. For a problem in two dimensions with triangular elements $t = 3$.

It is now time to make a very important observation: it may well happen that for some value(s) of i (3.32) or (3.33) will become singular. If this happens, the main question is if (3.32) or (3.33) is solvable or not. For the case when $a(\cdot, \cdot; \cdot)$ corresponds to a "pure" Laplacian operator – as we are going to show later in this Chapter when we discuss a priori effectivity analysis – we can prove solvability. We proceed by assuming that (3.32) and (3.33) are *either* non-singular and thus have a unique solution $\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^{na}(\mu)$; *or* for some i (3.32) and (3.33) is singular but solvable; in this case we have some flexibility in the choice of $\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^{na}(\mu)$. In the next Section we will demonstrate that in the case of 1D heat conduction model problem this flexibility does not affect the value of the error bound in any way.

We observe from (3.6), (3.31), and the fact that $\Phi_i(\underline{x})e_N(\mu) \equiv 0 \quad \forall \underline{x} \in \Omega \setminus \mathcal{P}_i$ that

$$R_i^{a|na}(\mu)(\mathcal{I}_i^{a|na}[\Phi_i^{a|na}e_N(\mu)]; \mu) = a(e_N(\mu), \Phi_i^{a|na}e_N(\mu); \mu) = a_{\mathcal{P}_i^{a|na}}(\mathcal{I}_i^{a|na}e_N(\mu), \mathcal{I}_i^{a|na}\Phi_i^{a|na}e_N(\mu); \mu). \quad (3.34)$$

If we now perform summation of (3.34) over all $\bar{\mathcal{P}}_i, i = 1, \dots, S$ and invoke (3.15), the partition of unity property, and bilinearity of $a(., .; .)$ we arrive at

$$\begin{aligned} \sum_{i=1}^J R_i^a(\mathcal{I}_i^a[\Phi_i^a e_N(\mu)]; \mu) + \sum_{i=1}^G R_i^{na}(\mathcal{I}_i^{na}[\Phi_i^{na} e_N(\mu)]; \mu) &= \sum_{i=1}^J a(e_N(\mu), \Phi_i^a e_N(\mu); \mu) + \sum_{i=1}^G a(e_N(\mu), \Phi_i^{na} e_N(\mu); \mu) = \\ &= \sum_{i=1}^S a(e_N(\mu), \Phi_i e_N(\mu); \mu) = a(e_N(\mu), e_N(\mu); \mu). \end{aligned} \quad (3.35)$$

We are now ready to demonstrate the following result:

$$\begin{aligned} a(e_N(\mu), e_N(\mu); \mu) &= \left(\sum_{i=1}^J c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \mathcal{I}_i^a e_N(\mu)) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \mathcal{I}_i^{na} e_N(\mu); \mu) \right) \leq \\ &= \left(\sum_{i=1}^J (c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^a(\mu))^{\frac{1}{2}} (c_i(\mu) \hat{a}_i(\mathcal{I}_i^a e_N(\mu), \mathcal{I}_i^a e_N(\mu))^{\frac{1}{2}} + \right. \\ &\quad \left. \sum_{i=1}^G (a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \hat{e}_{N,i}^{na}(\mu); \mu))^{\frac{1}{2}} (a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} e_N(\mu), \mathcal{I}_i^{na} e_N(\mu); \mu))^{\frac{1}{2}} \right) \leq \\ &= \left(\sum_{i=1}^J c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^a(\mu)) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \hat{e}_{N,i}^{na}(\mu); \mu) \right)^{\frac{1}{2}} \times \\ &= \left(\sum_{i=1}^J c_i(\mu) \hat{a}_i(\mathcal{I}_i^a e_N(\mu), \mathcal{I}_i^a e_N(\mu)) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} e_N(\mu), \mathcal{I}_i^{na} e_N(\mu); \mu) \right)^{\frac{1}{2}}, \end{aligned} \quad (3.36)$$

where the first step (the equality) follows from (3.35), (3.32), and (3.33); the second — from application of Cauchy-Schwartz' inequality to the terms $c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \mathcal{I}_i^a e_N(\mu))$ and $a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \mathcal{I}_i^{na} e_N(\mu); \mu)$; the third — from application of Cauchy-Schwartz' inequality to the summations over J and G . We finally invoke (3.30) and the fact that each element of the coarse discretization T_S appears exactly in t patches \mathcal{P}_i to obtain

$$\sum_{i=1}^J c_i(\mu) \hat{a}_i(\mathcal{I}_i^a e_N(\mu), \mathcal{I}_i^a e_N(\mu)) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} e_N(\mu), \mathcal{I}_i^{na} e_N(\mu); \mu) \leq t a(e_N(\mu), e_N(\mu); \mu). \quad (3.37)$$

We are now in the position to construct the error bound $\tilde{\Delta}_{NS}(\mu)$ based on the results obtained in (3.36) and (3.37)

$$a(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_{NS}(\mu) = t \left(\sum_{i=1}^J c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^a(\mu)) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \hat{e}_{N,i}^{na}(\mu); \mu) \right). \quad (3.38)$$

We then invoke the coercivity of $a(\cdot, \cdot; \cdot)$ to obtain Δ_{NS} , the error bound for $\|e_N(\mu)\|_Y$:

$$\|e_N(\mu)\|_Y \leq \Delta_{NS} = \sqrt{\frac{\tilde{\Delta}_{NS}(\mu)}{\alpha(\mu)}}. \quad (3.39)$$

In order to obtain the error bound for the compliant output $\tilde{s}_N(\mu)$ we expand $a(e_N(\mu), e_N(\mu); \mu)$ using (3.6), (3.20), and (3.22).

$$0 \leq a(e_N(\mu), e_N(\mu); \mu) = f(u(\mu)) - f(u_N(\mu) + \delta_S(\mu)) + a(u_N(\mu), \delta_S(\mu); \mu) \leq \tilde{\Delta}_{NS}(\mu). \quad (3.40)$$

as follows

$$\begin{aligned} f(u_N(\mu)) + f(\delta_S(\mu)) - a(u_N(\mu), \delta_S(\mu); \mu) &\leq \underbrace{f(u(\mu))}_{s(\mu)} \leq \\ &\tilde{\Delta}_{NS}(\mu) + f(u_N(\mu)) + f(\delta_S(\mu)) - a(u_N(\mu), \delta_S(\mu); \mu). \end{aligned} \quad (3.41)$$

We note from (3.22) that

$$f(\delta_S(\mu)) = a(\delta_S(\mu), \delta_S(\mu); \mu) + a(u_N(\mu), \delta_S(\mu); \mu); \quad (3.42)$$

and since $a(\delta_S(\mu), \delta_S(\mu); \mu) \geq 0$ we finally obtain $s^+(\mu)$ and $s^-(\mu)$, the upper and the lower bounds for the reduced basis approximation of the compliant output:

$$s^-(\mu) = f(u_N(\mu)) + a(\delta_S(\mu), \delta_S(\mu); \mu) \leq \underbrace{f(u(\mu))}_{s(\mu)} \leq f(u_N(\mu)) + \tilde{\Delta}_{NS}(\mu) = s^+(\mu). \quad (3.43)$$

We introduce two effectivities $\tilde{\eta}_N(\mu)$ and $\eta_N(\mu)$ which measure the sharpness of our error bounds

$$\tilde{\eta}_N(\mu) = \frac{\tilde{\Delta}_{NS}(\mu)}{a(e_N(\mu), e_N(\mu); \mu)}, \quad (3.44)$$

$$\eta_N(\mu) = \frac{\Delta_{NS}(\mu)}{\|e_N(\mu)\|_Y}. \quad (3.45)$$

$$(3.46)$$

The error bounds for non-compliant outputs can be easily obtained using the results of presented in Section 2.3.3 or the duality approach [49, 20].

A Priori Effectivity Analysis

We now prove the bounding properties for $\tilde{\eta}_N(\mu), \eta_N(\mu)$. We start by introducing a constant $\sigma_i^{a|na}$ such that

$$\hat{a}_i(\mathcal{I}_i^a \Phi_i^a v, \mathcal{I}_i^a \Phi_i^a v) \leq \sigma_i^a \hat{a}_i(v, v), \quad \forall v \in Y_i^a, \quad (3.47)$$

$$a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} \Phi_i^{na} v, \mathcal{I}_i^{na} \Phi_i^{na} v; \mu) \leq \sigma_i^{na} a_{\mathcal{P}_i^{na}}(v, v; \mu) \quad \forall v \in Y_i^{na}. \quad (3.48)$$

$$i = 1, \dots, J \text{ for } Y_i^a \text{ and } i = 1, \dots, G \text{ for } Y_i^{na}.$$

Invoking (3.32), (3.33) and the fact that the fact that $\Phi_i(\underline{x})e_N(\mu) \equiv 0 \quad \forall \underline{x} \in \Omega \setminus \mathcal{P}_i$ that

$$c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^a(\mu)) = R_i^a(\mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu); \mu) = a_{\mathcal{P}_i^a}(\mathcal{I}_i^a e_N(\mu), \mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu); \mu), \quad i = 1, \dots, J, \quad (3.49)$$

$$a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \hat{e}_{N,i}^{na}(\mu); \mu) = R_i^{na}(\mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu); \mu) = a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} e_N(\mu), \mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu); \mu), \quad i = 1, \dots, G. \quad (3.50)$$

Using (3.49), (3.50), and (3.38), we rewrite the expression for $\tilde{\Delta}_{NS}(\mu)$ as follows:

$$\begin{aligned} \tilde{\Delta}_{NS}(\mu) &= t \left(\sum_{i=1}^J a_{\mathcal{P}_i^a}(\mathcal{I}_i^a e_N(\mu), \mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu); \mu) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} e_N(\mu), \mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu); \mu) \right) \leq \\ & t \left(\sum_{i=1}^J (a_{\mathcal{P}_i^a}(\mathcal{I}_i^a e_N(\mu), \mathcal{I}_i^a e_N(\mu); \mu))^{\frac{1}{2}} (a_{\mathcal{P}_i^a}(\mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu), \mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu)))^{\frac{1}{2}} + \right. \\ & \left. \sum_{i=1}^G (a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} e_N(\mu), \mathcal{I}_i^{na} e_N(\mu); \mu))^{\frac{1}{2}} (a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu), \mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu)))^{\frac{1}{2}} \right) \leq \end{aligned}$$

$$t \left(\sum_{i=1}^S a_{\mathcal{P}_i}(\mathcal{I}_i e_N(\mu), \mathcal{I}_i e_N(\mu); \mu) \right)^{\frac{1}{2}} \times \\ \left(\sum_{i=1}^J a_{\mathcal{P}_i^a}(\mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu), \mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu); \mu) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu), \mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu); \mu) \right)^{\frac{1}{2}}. \quad (3.51)$$

The second step of (3.51) follows from application the Cauchy-Schwartz' inequality to the terms $a_{\mathcal{P}_i^a}(\mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu), \mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu); \mu)$; the third — from application yet again of the Cauchy-Schwartz' inequality to the summations over J and G . We proceed with (3.51) by invoking (3.34) for the first term in the big parentheses and (3.30) for the terms in the summation over J in the second pair of big parentheses:

$$\tilde{\Delta}_{NS}(\mu) \leq t (ta(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} \times \\ \left(\sum_{i=1}^J \gamma_i(\mu) \hat{a}_i(\mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu), \mathcal{I}_i^a \Phi_i^a \hat{e}_{N,i}^a(\mu); \mu) + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu), \mathcal{I}_i^{na} \Phi_i^{na} \hat{e}_{N,i}^{na}(\mu)) \right)^{\frac{1}{2}}; \quad (3.52)$$

we continue our argument by applying (3.49) and (3.50) to the terms in the summations over J and G , respectively,

$$\tilde{\Delta}_{NS}(\mu) \leq t (a(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} \times \\ \left(t \sum_{i=1}^J \gamma_i(\mu) \sigma_i^a(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^a(\mu)) + t \sum_{i=1}^G \sigma_i^{na}(\mu) a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \hat{e}_{N,i}^{na}(\mu); \mu) \right)^{\frac{1}{2}}; \quad (3.53)$$

and conclude with the following result

$$\tilde{\Delta}_{NS}(\mu) \leq t(a(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}} (\Xi \tilde{\Delta}_{NS}(\mu))^{\frac{1}{2}}, \quad (3.54)$$

where

$$\Xi = \max \left[\max_{i=1, \dots, J} \frac{\gamma_i \sigma_i^a}{c_i(\mu)}, \max_{i=1, \dots, G} \sigma_i^{na} \right]. \quad (3.55)$$

Finally, from (3.38) and (3.54) it follows that

$$1 \leq \tilde{\eta}_N(\mu) \leq \Xi t^2. \quad (3.56)$$

Invoking the continuity of $a(\cdot, \cdot; \cdot)$ and (3.39), we next provide an a priori bound for the effectivity $\eta_N(\mu)$:

$$1 \leq \eta_N(\mu) = \frac{\Delta_{NS}(\mu)}{\|e_N(\mu)\|_Y} = \sqrt{\frac{\tilde{\Delta}_{NS}(\mu)}{\alpha(\mu)(\|e_N(\mu)\|_Y)^2}} \leq \sqrt{\frac{\gamma(\mu)\tilde{\Delta}_{NS}(\mu)}{\alpha(\mu)a(e_N(\mu), e_N(\mu); \mu)}} \leq t\sqrt{\frac{\Xi\gamma(\mu)}{\alpha(\mu)}} \leq t\sqrt{\frac{\Xi\gamma_0}{\alpha_0}}. \quad (3.57)$$

A Priori Effectivity Analysis for 1D Heat Conduction Problem

In this Section we present the effectivity analysis for a simple one-dimensional case. Let us consider a one-dimensional segment $\Omega = \bigcup_{i=1}^{S-1} \Omega_i$ divided into $S - 1$ separate regions Ω_i as depicted in Figure 3-2. The points $x_i, i = 2, \dots, S - 1$ separate the regions with different parametric dependence.

The problem is characterized by $S - 2$ parameters which are conductivities of the material $k_i, i = 2, \dots, S - 2$, where k_i corresponds to Ω_i (all the conductivities are normalized with respect to $k_1 \equiv 1$, the conductivity of Ω_1). For simplicity of our analysis we require that the length of each Ω_i is equal to L . The parametric dependence in this problem is purely affine, however, the analysis we present in this Section directly applies to the non-affine case as well.

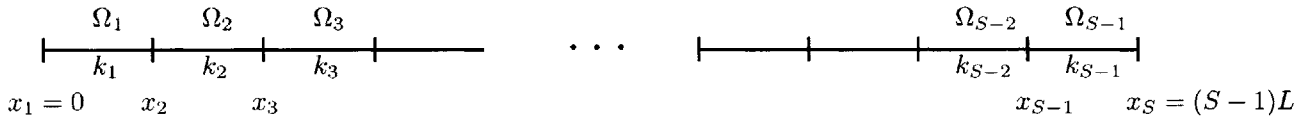


Figure 3-2: Partition of unity method, 1D model problem: domain configuration.

The temperature distribution satisfies,

$$-k_i \frac{\partial u_i(\mu)}{\partial x^2} = q', \quad i = 1, \dots, S - 1, \quad (3.58)$$

where $u_i(\mu)$ refers to the restriction of $u_i(\mu)$ on $\Omega_i(\mu)$ and q' is the uniform heat flux from outside.

We set $q' \equiv 1$.

We require the continuity of the temperature and heat-flux at the conductivity-discontinuity interfaces

$$\left. \begin{aligned} u_i(x_i; \mu) &= u_i(x_i; \mu), \\ -k_{i-1} \frac{\partial u_{i-1}(x_i; \mu)}{\partial x} &= -k_i \frac{\partial u_i(x_i; \mu)}{\partial x} \end{aligned} \right\} i = 2, \dots, S - 1;$$

Finally, we impose zero Dirichlet boundary conditions at $x_1 = 0$, and $x_S = 0$:

$$u_1(0; \mu) = 0, u_{S-1}((S-1)L; \mu) = 0,$$

The weak formulation of the problem is given by as follows: we look for $u(\mu) \in Y = H_0^1(\Omega) = \{v \in H^1(\Omega) | v(0) = 0, v(L) = 0\}$

$$\sum_{i=1}^S \int_{\Omega_i} k_i \frac{\partial u(\mu)}{\partial x} \frac{\partial v}{\partial x} = \int_{\Omega} v, \quad \forall v \in Y. \quad (3.59)$$

Consistently with the approach described in Section 3.2.1, we introduce a partition of unity $\Phi_i(x), i = 1, \dots, S$ as shown in Figure 3-3; $\mathcal{P}_i = \text{supp } \Phi_i$. We then define the function spaces $Y_1 = H_0^1(\bar{\mathcal{P}}_1) = \{v \in H_1(\bar{\mathcal{P}}_1) | v(0) = 0\}, Y_i = H^1(\bar{\mathcal{P}}_i), Y_S = H_0^1(\bar{\mathcal{P}}_S) = \{v \in H_1(\bar{\mathcal{P}}_S) | v(L) = 0\}$

$$\mathcal{P}_1 = \Omega_1, \mathcal{P}_S = \Omega_{S-1}, \mathcal{P}_i = \Omega_i \cup \Omega_{i-1}, i = 2, \dots, S-1;$$

where all the derivatives for the closed sets in the regularity conditions for H^1 are evaluated from inside the set. We also introduce the mapping operator $\mathcal{I}_i : Y \rightarrow Y_i$

$$\forall v \in Y \quad \mathcal{I}_i v = \bar{v} \in Y_i : \bar{v} = v|_{\mathcal{P}_i}, \quad i = 1, \dots, S. \quad (3.60)$$

We then proceed according to the algorithm described earlier in this Chapter. We first obtain

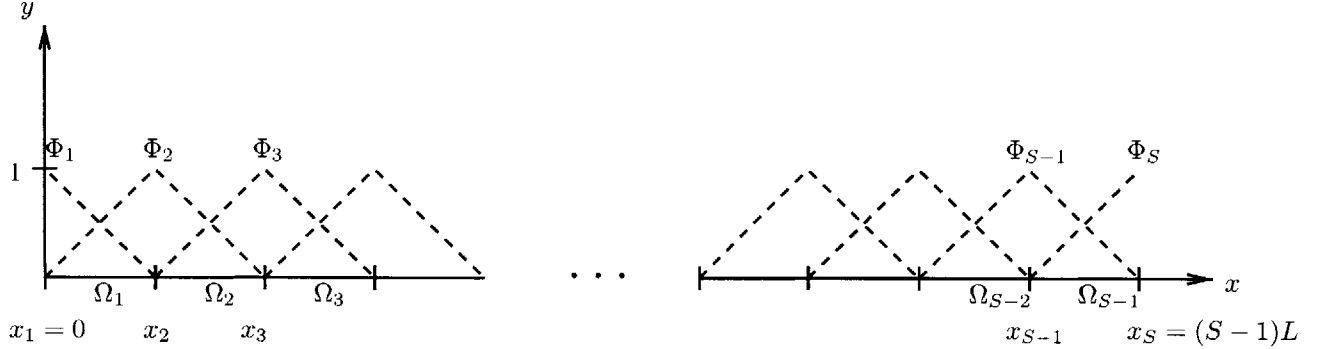


Figure 3-3: 1D model problem: $\Phi_i(x), i = 1, \dots, S$, partition of unity.

$u_N(\mu)$ and $\delta_S(\mu)$ as given by (3.20) and (3.22), respectively. The global error function is equal to $e_N(\mu) = u(\mu) - u_N(\mu) - \delta_S(\mu)$. We next define functions $c_i(\mu) : \mathcal{D} \rightarrow \mathbb{R}$, bilinear forms

$\hat{a}_i(\cdot, \cdot) : (Y_i)^2 \rightarrow \mathbb{R}, i = 1, \dots, S$ such that

$$c_1(\mu)\hat{a}_1(w, v) = k_1 \int_{\mathcal{P}_1} \nabla w \nabla v, \forall w, v \in Y_1, \quad (3.61)$$

$$c_i(\mu)\hat{a}_i(w, v) = \min[k_{i-1}, k_i] \int_{\mathcal{P}_i} \nabla w \nabla v, \forall w, v \in Y_i, i = 2, \dots, S-1, \quad (3.62)$$

$$c_S(\mu)\hat{a}_i(w, v) = k_{S-1} \int_{\mathcal{P}_S} \nabla w \nabla v, \forall w, v \in Y_S. \quad (3.63)$$

We next define the local residuals $R_i(v; \mu)$ as in (3.31)

$$R_1(v; \mu) = \int_{\mathcal{P}_1} v - k_1 \int_{\Omega_1} \mathcal{I}_1 \nabla \tilde{u}_N(\mu) \nabla v, \quad (3.64)$$

$$R_i(v; \mu) = \int_{\mathcal{P}_i} v - k_{i-1} \int_{\Omega_{i-1}} \mathcal{I}_i \nabla \tilde{u}_N(\mu) \nabla v, -k_i \int_{\Omega_i} \mathcal{I}_i \nabla \tilde{u}_N(\mu) \nabla v, i = 2, \dots, S-1 \quad (3.65)$$

$$R_S(v; \mu) = \int_{\mathcal{P}_S} v - k_{S-1} \int_{\Omega_{S-1}} \mathcal{I}_S \nabla \tilde{u}_N(\mu) \nabla v. \quad (3.66)$$

We then look for $\hat{e}_{N,i}(\mu) \in Y_i$ from the local error equations (3.32)

$$c_i(\mu)\hat{a}_i(\hat{e}_{N,i}(\mu), v) = R_i(\mathcal{I}_i \Phi v; \mu), v \in Y_i. \quad (3.67)$$

We note that for $i = 2, \dots, S-1$ (3.67) is singular but still solvable. To prove solvability we plug $v \equiv 1 \in Y_i$ into (3.67). We note that both $\hat{a}_i(\hat{e}_{N,i}(\mu), v) = 0$ because the gradient of v is 0, at the same time $R_i(\mathcal{I}_i \Phi v; \mu) = 0$, which is proven by plugging $v = \Phi_i$ into (3.22). Hence, if we switch to matrix-vector form and rewrite (3.67)

$$c_i(\mu) \hat{A}_i \hat{e}_{N,i}(\mu) = \underline{R}_i(\mu), \quad (3.68)$$

we can state that $\underline{R}_i(\mu)$ is orthogonal to the null space of \hat{A}_i and hence belongs to its column space. The general solution of (3.67) for $i = 3, \dots, S-2$ can be represented as $\hat{e}_{N,i}(\mu) + c$ where $\hat{e}_{N,i}(\mu) \neq 0$ which satisfies (3.67) and $c \in Y_i$ is a constant function. From (3.62) it is easy to see that

$$\hat{a}_i(\hat{e}_{N,i}(\mu) + c, \hat{e}_{N,i}(\mu) + c) = \hat{a}_i(\hat{e}_{N,i}(\mu), \hat{e}_{N,i}(\mu)), i = 2, \dots, S-1 \quad (3.69)$$

for any choice of c . We thus can choose c such that $\hat{e}_{N,i}(\mu) + c$ is of zero mean over \mathcal{P}_i . Keeping that in mind, we subsequently construct $\Delta_{NS}(\mu)$ according to (3.38):

$$\Delta_{NS}(\mu) = 2 \sum_{i=1}^S c_i(\mu) \hat{a}_i(\hat{e}_{N,i}(\mu), \hat{e}_{N,i}(\mu)). \quad (3.70)$$

We next proceed by expanding (3.70) in the following way:

$$\begin{aligned} \Delta_{NS}(\mu) &= 2 \sum_{i=1}^S R_i(\mathcal{I}_i \Phi_i \hat{e}_{N,i}(\mu); \mu) = 2k_1 \int_{\Omega_1} \nabla(\mathcal{I}_1 e_N(\mu)) \nabla(\mathcal{I}_1 \Phi_1 \hat{e}_{N,1}(\mu)) + \\ &2 \sum_{i=2}^{S-1} \left(k_{i-1} \int_{\Omega_{i-1}} \nabla(\mathcal{I}_i e_N(\mu)) \nabla(\mathcal{I}_i \Phi_i \hat{e}_{N,i}(\mu)) + k_i \int_{\Omega_i} (\nabla \mathcal{I}_i e_N(\mu)) \nabla(\mathcal{I}_i \Phi_i \hat{e}_{N,i}(\mu)) \right) + \\ &2k_{S-1} \int_{\Omega_{S-1}} \nabla(\mathcal{I}_S e_N(\mu)) \nabla(\mathcal{I}_S \Phi_S \hat{e}_{N,S}(\mu)) \Big). \end{aligned} \quad (3.71)$$

We observe that there are S terms in (3.71) in total, each of them corresponds to a partition $\Phi_i, i = 1, \dots, S$. We next formulate two Lemmas which are based on the Poincare-Friedrichs inequality for the 1D case.

Lemma 3.1. For any segment $\Sigma = [a, b] \subset \mathbb{R}, \forall v \in Y_{\Sigma}^1 = \{v \in H^1(\Sigma) | v(a) = 0, v'(b) = 0\}$

$$\int_{\Sigma} v^2 \leq \frac{4(b-a)^2}{\pi^2} \int_{\Sigma} (\nabla v)^2. \quad (3.72)$$

The same statement holds if we replace Y_{Σ}^1 in Lemma 3.1 by $\{v \in H^1(\Sigma) | v(b) = 0, v'(a) = 0\}$.

The proof of Lemma 3.1 follows from considering the Rayleigh quotient $\max_{v \in Y_{\Sigma}^1} \frac{\int_{\Sigma} (\nabla v)^2}{\int_{\Sigma} v^2}$ which is equivalent to the eigenproblem of this Lemma.

Lemma 3.2. For any segment $\Sigma = [a, b] \subset \mathbb{R}, \forall v \in Y_{\Sigma}^2 = \{v \in H^1(\Sigma) | v'(a) = 0, v'(b) = 0, \int_{\Sigma} v = 0\}$

$$\int_{\Sigma} v^2 \leq \frac{(b-a)^2}{\pi^2} \int_{\Sigma} (\nabla v)^2. \quad (3.73)$$

The proof of Lemma 3.1 follows from considering the Rayleigh quotient $\max_{v \in Y_{\Sigma}^2} \frac{\int_{\Sigma} (\nabla v)^2}{\int_{\Sigma} v^2}$ which is equivalent to the eigenproblem of this Lemma.

We now look at the individual terms in (3.71) and note that for $\Phi_i, i = 3, \dots, S-2$ (note that for the sake of brevity of our formulas we dropped the dependence of $e_N(\mu)$ and $e_{N,i}(\mu)$ on μ though we mean it implicitly)

$$\begin{aligned}
& k_{i-1} \int_{\Omega_{i-1}} \nabla(\mathcal{I}_i e_N) \nabla(\mathcal{I}_i \Phi_i \hat{e}_{N,i}) + k_i \int_{\Omega_i} (\nabla \mathcal{I}_i e_N) \nabla(\mathcal{I}_i \Phi_i \hat{e}_{N,i}) \leq \\
& \left(k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N)^2 \right)^{\frac{1}{2}} \left(k_{i-1} \int_{\Omega_{i-1}} (\nabla(\Phi_i \hat{e}_{N,i}))^2 \right)^{\frac{1}{2}} + \left(k_i \int_{\Omega_i} (\nabla e_N)^2 \right)^{\frac{1}{2}} \left(k_i \int_{\Omega_i} (\nabla(\Phi_i \hat{e}_{N,i}))^2 \right)^{\frac{1}{2}} \leq \\
& \left(k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N)^2 + k_i \int_{\Omega_i} (\nabla e_N)^2 \right)^{\frac{1}{2}} \left(\max[k_{i-1}, k_i] \int_{\mathcal{P}_i} (\nabla(\Phi_i \hat{e}_{N,i}))^2 \right)^{\frac{1}{2}} \leq \\
& \left(k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N)^2 + k_i \int_{\Omega_i} (\nabla e_N)^2 \right)^{\frac{1}{2}} \left(\max[k_{i-1}, k_i] \int_{\mathcal{P}_i} (\nabla \Phi_i)^2 (\hat{e}_{N,i})^2 + 2|\nabla \Phi_i \hat{e}_{N,i} \Phi_i \nabla \hat{e}_{N,i}| + (\Phi_i)^2 (\nabla \hat{e}_{N,i})^2 \right)^{\frac{1}{2}} \leq \\
& \left(k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N)^2 + k_i \int_{\Omega_i} (\nabla e_N)^2 \right)^{\frac{1}{2}} \times \\
& \max[k_{i-1}, k_i]^{\frac{1}{2}} \left(\int_{\mathcal{P}_i} \frac{1}{L^2} (\hat{e}_{N,i})^2 + \frac{2}{L} \left\{ \int_{\mathcal{P}_i} (\hat{e}_{N,i})^2 \right\}^{\frac{1}{2}} \left\{ \int_{\mathcal{P}_i} (\nabla \hat{e}_{N,i})^2 \right\}^{\frac{1}{2}} + \int_{\mathcal{P}_i} (\nabla \hat{e}_{N,i})^2 \right)^{\frac{1}{2}}. \quad (3.74)
\end{aligned}$$

The first step of (3.74) follows from Cauchy-Schwartz' inequality applied to both terms of in the first line of (3.74), the second step – from Cauchy-Schwartz' inequality applied to the two term summation in the second line, the third step – from the expansion of the integrand of the second term in the product in third line, and the fourth step – from Cauchy-Schwartz' inequality applied to the cross-term in the expansion of the second term of the product in the fourth line. We now apply Lemma 3.2 to the first and second integrals in the second pair of big parenthesis of (3.74) (which cancels the $\frac{1}{L}$ multipliers since the length of $\mathcal{P}_i = 2L, i = 3, \dots, S-2$) to obtain

$$\left(k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N)^2 + k_i \int_{\Omega_i} (\nabla e_N)^2 \right)^{\frac{1}{2}} \left(\max[k_{i-1}, k_i] \left(\frac{4}{\pi^2} + \frac{4}{\pi} + 1 \right) \int_{\mathcal{P}_i} (\nabla \hat{e}_{N,i})^2 \right)^{\frac{1}{2}} =$$

$$\left(1 + \frac{2}{\pi}\right) \left(k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N)^2 + k_i \int_{\Omega_i} (\nabla e_N)^2 \right)^{\frac{1}{2}} \left(\max[k_{i-1}, k_i] \int_{\mathcal{P}_i} (\nabla \hat{e}_{N,i})^2 \right)^{\frac{1}{2}} \quad (3.75)$$

we denote $\rho_i = \max[\frac{k_{i-1}}{k_i}, \frac{k_i}{k_{i-1}}]$ and invoke (3.62) to rewrite the right hand side of (3.75) as

$$\left(1 + \frac{2}{\pi}\right) \sqrt{\rho_i} \left(k_{i-1} \int_{\Omega_{i-1}} [\nabla e_N(\mu)]^2 + k_i \int_{\Omega_i} [\nabla e_N(\mu)]^2 \right)^{\frac{1}{2}} (c_i(\mu) \hat{a}_i(\hat{e}_{N,i}(\mu), \hat{e}_{N,i}(\mu)))^{\frac{1}{2}}, \quad (3.76)$$

We readily provide similar results for $\Phi_{1|S}, i = 1|S$ (because of Dirichlet boundary conditions hence Lemma 3.1 – and the fact that the length of $\mathcal{P}_{1|S}$ is equal to L):

$$k_{1|S} \int_{\mathcal{P}_{1|S}} \nabla(\mathcal{I}_{1|S} e_N(\mu)) \nabla(\mathcal{I}_{1|S} \Phi_{1|S} \hat{e}_{N,1|S}(\mu)) \leq \left(1 + \frac{2}{\pi}\right) \sqrt{\rho_{1|S}} (c_{1|S}(\mu) \hat{a}_{1|S}(\hat{e}_{N,i|S}(\mu), \hat{e}_{N,i|S}(\mu)))^{\frac{1}{2}} \left[k_{1|S-1} \int_{\Omega_{1|S-1}} (\nabla e_N(\mu))^2 \right]^{\frac{1}{2}}, \quad (3.77)$$

where $\rho_{1|S} = 1$; and finally for $\Phi_i, i = 2, S-1$ (again thanks to Dirichlet boundary conditions and the fact that the length of $\mathcal{P}_{2|S-1}$ is equal to $2L$):

$$k_{i-1} \int_{\Omega_i} \nabla(\mathcal{I}_{i-1} e_N(\mu)) \nabla(\mathcal{I}_{i-1} \Phi_{i-1} \hat{e}_{N,i-1}(\mu)) + k_i \int_{\Omega_i} (\nabla \mathcal{I}_i e_N(\mu)) \nabla(\mathcal{I}_i \Phi_i \hat{e}_{N,i}(\mu)) \leq \left(1 + \frac{4}{\pi}\right) \sqrt{\rho_i} (c_i(\mu) \hat{a}_i(\hat{e}_{N,i}(\mu), \hat{e}_{N,i}(\mu)))^{\frac{1}{2}} \left[k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N(\mu))^2 + k_i \int_{\Omega_i} (\nabla e_N(\mu))^2 \right]^{\frac{1}{2}}, \quad (3.78)$$

where $\rho_i = \max[\frac{k_{i-1}}{k_i}, \frac{k_i}{k_{i-1}}]$.

We conclude our argument by invoking (3.71), (3.76), (3.77), and (3.78) to arrive at

$$\Delta_{NS}(\mu) \leq 2 \left(1 + \frac{4}{\pi}\right) \sum_{i=1}^S \sqrt{\rho_i} (c_i(\mu) \hat{a}_i(\hat{e}_{N,i}(\mu), \hat{e}_{N,i}(\mu)))^{\frac{1}{2}} \left[k_{i-1} \int_{\Omega_{i-1}} (\nabla e_N(\mu))^2 + k_i \int_{\Omega_i} (\nabla e_N(\mu))^2 \right]^{\frac{1}{2}} \leq 2 \left(1 + \frac{4}{\pi}\right) \sqrt{\rho_{max}} \left(\frac{1}{2} \tilde{\Delta}_{NS}(\mu)\right)^{\frac{1}{2}} (2a(e_N(\mu), e_N(\mu); \mu))^{\frac{1}{2}}, \quad (3.79)$$

where $\rho_{max} = \max_{i=1,\dots,S} \rho_i$. We conclude that

$$1 \leq \tilde{\eta}_N(\mu) = \frac{\tilde{\Delta}_{NS}(\mu)}{a(e_N(\mu), e_N(\mu); \mu)} \leq 4 \left(1 + \frac{4}{\pi}\right)^2 \rho_{max}. \quad (3.80)$$

We invoke the coercivity and continuity of $a(\cdot, \cdot; \cdot)$, and (3.45) to provide a similar result for the effectivity $\eta_N(\mu)$ given by (3.45).

$$1 \leq \eta_N(\mu) = \frac{\Delta_{NS}(\mu)}{\|e_N\|} \leq 2 \left(1 + \frac{4}{\pi}\right) \sqrt{\frac{\rho_{max} \gamma_0}{\alpha_0}}. \quad (3.81)$$

We now provide concrete numerical results for the case when $S = 11$. We select our parameter domain \mathcal{D} as $\mathcal{D} = [0.10, 10]^9$, that is the conductivity satisfies $0.10 \leq k_i \leq 10$, $i = 2, \dots, 10$. We select the sample S_N out of the bigger sample S_{pool} of randomly chosen $N_{pool} = 100000$ points according to the algorithm described in Section 2.3.4. In Table 3.1 we present the results for the values of effectivities $\tilde{\eta}_N$, η_N as functions of N averaged over S_{test} , a sample of $N_{test} = 100000$ randomly chosen points which provides some level of statistical significance for our test. In the fourth and fifth columns we present a priori upper bounds for $\tilde{\eta}_N, \eta_N$ according to (3.80), (3.81), respectively. We observe that the upper bound for $\tilde{\eta}_N$ is more pessimistic than the similar bound for η_N which is explained by the fact that the former depends linearly on ρ_{max} , whereas the latter only scales as $\sqrt{\rho_{max}}$ where $\rho_{max} \in [1, 100]$;

We need to say that the problem we considered is, in fact, very simple and the solution of it is a quadratic function of x in each Ω_i , $i = 1, \dots, 10$ so when N becomes big enough the reduced basis approximation $u_N(\mu)$ is equal to the truth solution $u(\mu)$. We can avoid this situation by introducing a term $\beta_i u_i(\mu)$, $i = 1, \dots, 10$ into (3.58). In this case the solution $u(\mu)$ would have exponential components in it and the exact match between $u_N(\mu)$ and $u(\mu)$ will no longer be possible. However, even in this case we would have to consider the "solvability" issues for the cases when β_i is equal or close to 0.

3.2.4 Off-line/On-line Procedure

Reduced Basis Approximation

We rewrite (3.20) in the matrix vector form to obtain an $N \times N$ linear system of equations

$$\underline{A}_N(\mu) \underline{u}_N(\mu) = \underline{F}_N, \quad (3.82)$$

N	$\tilde{\eta}_N(\mu)$	$\eta_N(\mu)$	$4\left(1 + \frac{4}{\pi}\right)^2 \rho_{max}$	$2\left(1 + \frac{4}{\pi}\right) \sqrt{\frac{\rho_{max}\gamma_0}{\alpha_0}}$
1	2.12	5.73		
2	2.04	5.68		
3	2.67	7.62		
4	2.88	8.97		
5	3.01	9.18		
6	3.15	9.50		
7	2.89	8.46		
8	2.85	8.20	286.17	55.65
9	2.82	8.40		
10	2.61	8.20		
11	2.86	9.14		
12	2.64	8.50		
13	2.51	8.32		
14	2.41	7.71		
15	2.49	7.60		

Table 3.1: Partition of unity method, 1D model problem: a priori and a posteriori effectivity analysis.

where $\underline{A}_N(\mu)$ admits decomposition consistent with (3.3)

$$\underline{A}_N(\mu) = \sum_{q=1}^Q \Theta_q(\mu) \underline{A}_N^q + \underline{A}_N^{na}(\mu), \quad (3.83)$$

where

$$\underline{A}_{N_{ij}}^q = a^q(\zeta_i, \zeta_j), \underline{A}_{N_{ij}}^{na} = a^{NA}(\zeta_i, \zeta_j; \mu), \underline{F}_{N_i} = f(\zeta_i), q = 1, \dots, Q, i, j = 1, \dots, N. \quad (3.84)$$

We now turn to the computation of $\underline{\delta}_S(\mu)$ from (3.22); this requires solution of an $S \times S$ linear system of equations

$$\underline{A}_S(\mu) \underline{\delta}_S(\mu) = \underline{F}_S(\mu) - \underline{A}_{SN}(\mu) \underline{u}_N(\mu), \quad (3.85)$$

where

$$\underline{F}_{S_i} = f(\Phi_i), i = 1, \dots, S; \quad (3.86)$$

and matrices $\underline{A}_S(\mu), \underline{A}_{SN}(\mu)$ admit the following decomposition

$$\underline{A}_S(\mu) = \sum_{i=1}^Q \Theta_q(\mu) \underline{A}_S^q + \underline{A}_S^{na}(\mu), \quad (3.87)$$

$$\underline{A}_{SN}(\mu) = \sum_{i=1}^Q \Theta_q(\mu) \underline{A}_{SN}^q + \underline{A}_{SN}^{na}(\mu), \quad (3.88)$$

where, in turn, for $q = 1, \dots, Q$

$$\underline{A}_{Sij}^q = a^q(\Phi_i, \Phi_j), \underline{A}_{Nij}^{na} = a^{NA}(\Phi_i, \Phi_j; \mu), \quad i, j = 1, \dots, N, \quad (3.89)$$

$$\underline{A}_{SNij}^q = a^q(\Phi_i, \zeta_j), \underline{A}_{SNij}^{na} = a^{NA}(\Phi_i, \zeta_j; \mu), \quad i = 1, \dots, S, j = 1, \dots, N. \quad (3.90)$$

The vector $\underline{\delta}_S(\mu)$ is obtained by solution of (3.85). We note that the system (3.85) is not only low-dimensional – since S is typically quite small – but also sparse because it corresponds to the finite element discretization T_S . These two facts suggest that (3.85) is very cheap to solve: depending on the nature of the bilinear form $a(\cdot, \cdot; \cdot)$ the operation count for solution of (3.85) is expected to be $O(S^\kappa)$ where $\kappa \in (1, 2]$.

Error Bound

For each of the affine partitions $\Phi_i^a, i = 1, \dots, J$ we denote a set $\{a_{\mathcal{P}_i^a}^1(\cdot, \cdot), \dots, a_{\mathcal{P}_i^a}^{Q_i}(\cdot, \cdot)\}$ of bilinear forms for which there exists $v_j \in Y_i^a$ such that $v_j|_{\mathcal{P}_i^a} \neq 0, a_{\mathcal{P}_i^a}^j(v_j, v_j) \neq 0, j = 1, \dots, Q_i$, the set of affine coefficients corresponding to $a_{\mathcal{P}_i^a}^j(\cdot, \cdot)$ is then denoted as $\{\Theta_1^{\mathcal{P}_i^a}, \dots, \Theta_{Q_i}^{\mathcal{P}_i^a}\}$. Clearly, $Q_i \leq Q, \forall i = 1, \dots, J$. We then define $\{\Phi_1^{\mathcal{P}_i^a}, \dots, \Phi_{S_i}^{\mathcal{P}_i^a}\}$, the set of partitions the support of which has a non-zero intersection with \mathcal{P}_i^a . We denote the projection operators and the coefficients in (3.23) corresponding to $\Phi_j^{\mathcal{P}_i^a}$ as $\mathcal{I}_j^{\mathcal{P}_i^a}, \delta_j^{\mathcal{P}_i^a}, j = 1, \dots, S_i$.

We next note that for each patch \mathcal{P}_i^a (3.31) the expression for $R_i^a(\mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v; \mu)$ can be rewritten as

$$R_i^a(\mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v; \mu) = f_{\mathcal{P}_i^a}(\mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v) - \sum_{q=1}^{Q_i} \Theta_q^{\mathcal{P}_i^a} \left(\sum_{l=1}^N u_{N_l} a_{\mathcal{P}_i^a}^q(\mathcal{I}_l \zeta_l, \mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v) + \sum_{k=1}^{S_i} \delta_{S_k}^{\mathcal{P}_i^a} a_{\mathcal{P}_i^a}^q(\mathcal{I}_k^{\mathcal{P}_i^a} \Phi_k^{\mathcal{P}_i^a}, \mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v) \right), \quad (3.91)$$

$$\forall v \in Y_i^a, \quad i = 1, \dots, J.$$

We next renumber (3.91) and rewrite it in the following form (compare to (2.51)):

$$\sum_{j=1}^{K_i} \tau_j^i(\mu) \Upsilon_j^i(v), \quad (3.92)$$

where $K_i = 1 + Q_i(N + S_i)$ and

$$\tau_1^i(\mu) = 1, \quad \Upsilon_1^i(v) = f_{\mathcal{P}_i^a}(\mathcal{I}_1^{\mathcal{P}_i^a} \Phi_i^a v); \quad (3.93)$$

for $j = 2, \dots, 1 + Q_i N : j = 1 + (q-1)l, q = 1, \dots, Q_i, l = 1, \dots, N$

$$\tau_j^i(\mu) = \Theta_q^{\mathcal{P}_i^a}(\mu) u_{N_l}(\mu), \quad \Upsilon_j^i(v) = a_{\mathcal{P}_i^a}^q(\mathcal{I}_l \zeta_l, \mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v); \quad (3.94)$$

For $j = 2 + Q_i N, \dots, 1 + Q_i(N + S_i) : j = 1 + Q_i N + (q-1)k, q = 1, \dots, Q_i, k = 1, \dots, S_i$

$$\tau_j^i(\mu) = \Theta_q^{\mathcal{P}_i^a}(\mu) \delta_{S_k}^{\mathcal{P}_i^a}, \quad \Upsilon_j^i(v) = a_{\mathcal{P}_i^a}^q(\mathcal{I}_k^{\mathcal{P}_i^a} \Phi_k^{\mathcal{P}_i^a}, \mathcal{I}_i^{\mathcal{P}_i^a} \Phi_i^a v). \quad (3.95)$$

Using (3.93), (3.94), (3.95), (3.38) and (3.91) it is a simple matter to demonstrate that

$$t \sum_{i=1}^J c_i(\mu) \hat{a}_i(\hat{e}_{N,i}^a(\mu), \hat{e}_{N,i}^a(\mu); \mu) = t \sum_{i=1}^J \frac{1}{c_i(\mu)} \sum_{j=1}^{K_i} \sum_{j'=1}^{K_i} \tau_j^i(\mu) \tau_{j'}^i(\mu) \underbrace{\hat{a}_i(z_j^i, z_{j'}^i)}_{\Lambda_{jj'}^i}, \quad (3.96)$$

where $z_j^i, i = 1, \dots, J; j = 1, \dots, K_i = 1 + Q_i(N + S_i)$ is obtained as the solution of

$$\hat{a}_i(z_j^i, v) = \Upsilon_j^i(v), \quad \forall v \in Y_i^a, \quad (3.97)$$

$$\Lambda_{jj'}^i = \hat{a}_i(z_j^i, z_{j'}^i). \quad (3.98)$$

Therefore, (3.38) can be rewritten as

$$\Delta_{NS}(\mu) = t \left(\sum_{i=1}^J \frac{1}{c_i(\mu)} \sum_{j=1}^{K_i} \sum_{j'=1}^{K_i} \tau_j^i(\mu) \tau_{j'}^i(\mu) \Lambda_{jj'}^i + \sum_{i=1}^G a_{\mathcal{P}_i^{na}}(\hat{e}_{N,i}^{na}(\mu), \hat{e}_{N,i}^{na}(\mu); \mu) \right). \quad (3.99)$$

We observe that in (3.99) evaluation of the second sum requires solution of G linear systems (3.33) for $\hat{e}_{N,i}^{na}(\mu), i = 1, \dots, G$. In some cases it might be computationally advantageous to replace all of the partitions corresponding to Φ_i^{na} by one partition that will be equal to the sum of those

”non-affine” partitions. This trick allows us to reduce G to 1.

We note that we can obtain the term $a(\delta_S(\mu), \delta_S(\mu); \mu)$ from (3.43) by reusing the equation (3.85). Clearly, evaluation of $a(\delta_S(\mu), \delta_S(\mu); \mu)$ requires $O(S)$ operations.

Numerical Algorithm

Off-line

1. Generate S_N as described in Section 2.3.4 and construct the basis for W_N ;
2. Precompute $\underline{F}_N, \underline{A}_N^q$ as in (3.84);
3. Precompute \underline{F}_S as in (3.86) and $\underline{A}_S^q, \underline{A}_{SN}^q$ as in (3.89), (3.90), $q = 1, \dots, Q$;
4. Precompute $\Lambda_{jj'}^i$ as in (3.98), $i = 1, \dots, J$; $j, j' = 1, \dots, 1 + Q_i(N + S_i)$.

On-line

1. Form \underline{A}_N as in (3.83) and solve (3.82) for $\underline{u}_N(\mu)$. Cost: $O(QN^2) + O(n_{na}N^2) + O(N^3)$;
2. Form \underline{A}_S as in (3.87), \underline{A}_{SN} as in (3.88) and solve (3.85) for $\underline{\delta}_S(\mu)$. Cost: $O(QS) + O(n_{na}S^2) + O(QSN) + O(n_{na}SN) + O(S^\kappa)$, $\kappa \in (1, 2]$ as discussed in Section 3.2.4.
3. Calculate $\Delta_{NS}(\mu)$ as in (3.99). Cost: $O(\sum_{i=1}^J Q_i^2(S_i + N)^2) + O(n_{na}SN) + O(Gn_{na}^\kappa)$. The last term accounts for G matrix assemblies in (3.33) and subsequent solution of $\tilde{N} \times \tilde{N}$ sparse linear system, where $\tilde{N} \approx n_{na}^1$. Recall that G could be made equal to 1 for any configuration of partitions at the affine/non-affine interface.

$$\text{Total on-line cost: } O(QN^2) + O(n_{na}N^2) + O(N^3) + O(QS) + O(n_{na}S^2) + O(QSN) + O(n_{na}SN) + O(S^\kappa) + O(\sum_{i=1}^J Q_i^2(S_i + N)^2) + O(Gn_{na}^\kappa).$$

3.3 Model Problem: 2D Heat Conduction Example

In this Section we consider a locally non-affine model problem which is a variation of the model problems introduced in Sections 1.1 and 2.4. We provide this model problem to illustrate application of the partition of unity reduced basis method. We consider a two-dimensional fin which consists of a vertical central “post” and four horizontal “sub-fins” on each side of the post; the fin conducts

¹ \tilde{N} can be slightly bigger or smaller than n_{na} because of geometric issues, e.g. the inclusion of some “affine” areas into some members of \mathcal{P}^{na} .

heat from a prescribed uniform flux “source” at the root, Γ_{root} , through the large-surface-area subfins to surrounding flowing air by convection. The thermal fin is characterized by 14 parameters which constitute a 14-dimensional vector,

$$\mu = \left(\underbrace{k_1}_{\mu_1}, \underbrace{k_2}_{\mu_2}, \underbrace{k_3}_{\mu_3}, \underbrace{k_4}_{\mu_4}, \underbrace{\text{Bi}}_{\mu_5}, \underbrace{\beta_1}_{\mu_6}, \underbrace{\beta_2}_{\mu_7}, \underbrace{\beta_3}_{\mu_8}, \underbrace{\beta_4}_{\mu_9}, \underbrace{\alpha_1}_{\mu_{10}}, \underbrace{\alpha_2}_{\mu_{11}}, \underbrace{\alpha_3}_{\mu_{12}}, \underbrace{\alpha_4}_{\mu_{13}}, \underbrace{\tau}_{\mu_{14}} \right),$$

where $k_i, i = 1, \dots, 4$ is the conductivity of the i^{th} subfin ($\tilde{\Omega}_i$), Bi is the Biot number, a non-dimensional heat transfer coefficient reflecting convective transport to the air at the fin surfaces, $\beta_i, \alpha_i, i = 1, \dots, 4$ are the length and the thickness of the i^{th} subfin, respectively, as shown in the Figure 3-4. We have reduced the area of $\tilde{\Omega}_1$ to the subfin located on the left of the root. The non-affine component is introduced into the problem by the 5th subfin ($\tilde{\Omega}_5 = \tilde{\Omega}_{na}$) conductivity, $\tilde{k}_5(\tilde{x}, \mu)$, which is no longer a constant as it was in the affine case, but a function of τ and $\tilde{x} \in \tilde{\Omega}_5 = \tilde{\Omega}_{na}$:

$$\tilde{k}_5(\tilde{x}; \mu) = \exp \left(\tau \sqrt{(\tilde{x}_1 - x_1^*)^2 + (\tilde{x}_2 - x_2^*)^2} \right), \quad (3.100)$$

where \tilde{x}^* is the corner points of $\tilde{\Omega}_5$ as depicted in Figure 3-4. The length and thickness of $\tilde{\Omega}_5 = \tilde{\Omega}_{na}$ are equal to β_1 and α_1 , respectively.

The vertical distance between the subfins is fixed at .75 (measured relative to the post width). Thus, the total variable height of the fin $H = 3 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$. For our parameter domain we choose $\mathcal{D} = [0.1, 10.0]^4 \times [0.01, 1.0] \times [2, 3]^4 \times [0.1, 0.5]^4 \times [-0.7571, 0.7571]$, that is, $0.1 \leq k_i \leq 10.0, i = 2, \dots, 4$ for the conductivities, $0.01 \leq \text{Bi} \leq 1, 2 \leq \beta_i \leq 3, 0.1 \leq \alpha_i \leq 0.5, i = 1, \dots, 4$ for the geometric parameters and $-0.7571 \leq \tau \leq 0.7571$, for the parameter τ that defines the non-affine dependence. It is standard to demonstrate [46] that this problem well-posed.

We are primarily interested in the norm of the error $\|e_N(\mu)\|_Y = \|u(\mu) - u_N(\mu)\|_Y$ and the error bound for it; the approximation for the linear outputs can be obtained as it was shown in Section 2.3.3.

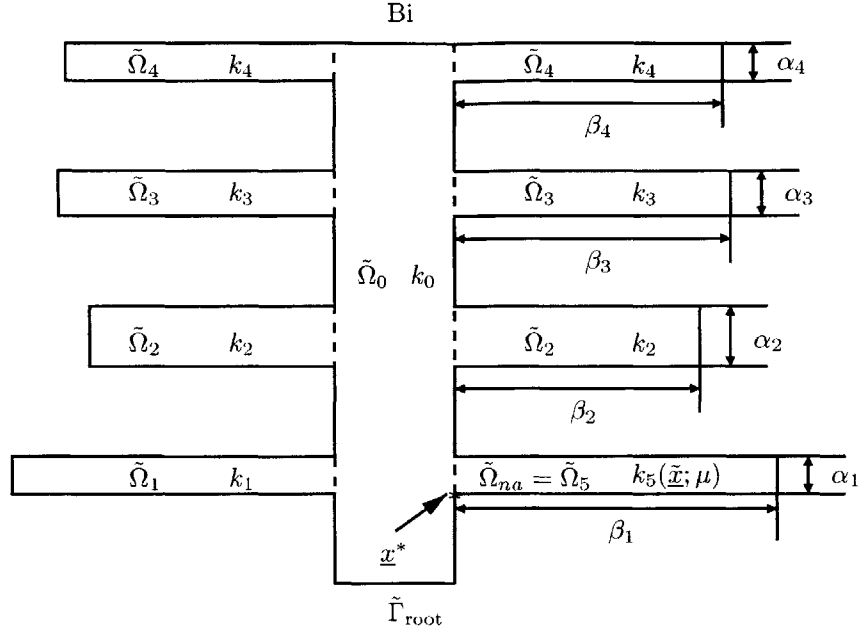


Figure 3-4: 2D locally non-affine model problem: a thermal fin with affine shape and non-affine conductivity variation.

3.3.1 Governing Partial Differential Equations

Strong Form

The temperature distribution $u(\mu)$, is obtained by solution of the following elliptic partial differential equations:

$$-k_i \nabla^2 \tilde{u}_i(\mu) = 0 \text{ in } \tilde{\Omega}_i, \quad i = 0, \dots, 4, \quad (3.101)$$

$$-\nabla(\tilde{k}_5(\underline{x}; \mu) \nabla \tilde{u}_5(\mu)) = 0 \text{ in } \tilde{\Omega}_5 = \tilde{\Omega}_{na}, \quad (3.102)$$

where ∇^2 is the Laplacian operator, and $\tilde{u}_i(\mu) \equiv u(\mu)|_{\tilde{\Omega}_i}$ refers to the restriction of $u(\mu)$ to $\tilde{\Omega}_i$. Here $\tilde{\Omega}_i$ is the region of the fin with conductivity k_i , $i = 0, \dots, 5$: $\tilde{\Omega}_0$ is thus the central post, and $\tilde{\Omega}_i$, $i = 1, \dots, 5$, corresponds to the four subfins. The use of " \sim " symbol over $u(\mu)$ signifies that the shape and size of the thermal fin – which area we denote as $\tilde{\Omega}(\mu)$ – depend on μ . After we switch to a parameter independent reference domain Ω later in this Chapter the " \sim " symbol will be dropped. Again we note that $\tilde{k}_5(\underline{x}; \mu)$ unlike k_0, k_1, k_2, k_3, k_4 is a function of \underline{x} .

We must also ensure continuity of temperature and heat flux at the conductivity-discontinuity

interfaces $\tilde{\Gamma}_i \equiv \partial\tilde{\Omega}_0 \cap \partial\tilde{\Omega}_i$, $i = 1, \dots, 5$, where $\partial\tilde{\Omega}_i$ denotes the boundary of $\tilde{\Omega}_i$:

$$\tilde{u}_0(\mu) = \tilde{u}_i(\mu), \quad (3.103)$$

$$-(\nabla\tilde{u}_0(\mu) \cdot \tilde{n}_i) = -k_i(\nabla\tilde{u}_i(\mu) \cdot \tilde{n}_i) \text{ on } \tilde{\Gamma}_i, \quad i = 1, \dots, 5; \quad (3.104)$$

here \tilde{n}_i is the outward normal on $\partial\tilde{\Omega}_i$. We introduce a Neumann boundary condition on the fin root:

$$-(\nabla\tilde{u}_0(\mu) \cdot \tilde{n}_o) = -1 \text{ on } \tilde{\Gamma}_{\text{root}}, \quad (3.105)$$

which models the heat source; and the Robin boundary condition:

$$-k_i(\nabla\tilde{u}_i(\mu) \cdot \tilde{n}_i) = \text{Bi } \tilde{u}_i(\mu) \text{ on } \tilde{\Gamma}_{\text{ext } i}, \quad i = 0, \dots, 5, \quad (3.106)$$

which models the convective heat losses.

Weak Form

Following the exact same approach as it was presented in Section 2.4.2 we recover the weak formulation of the original problem as

$$\sum_{i=0}^4 \int_{\tilde{\Omega}_i} k_i \nabla\tilde{u}(\mu) \nabla v + \text{Bi} \int_{\tilde{\Gamma}_{\text{ext}}} \tilde{u}_i(\mu) v + \int_{\tilde{\Omega}_5} \tilde{k}_5(\tilde{\mathcal{X}}; \mu) \nabla\tilde{u}(\mu) \nabla v = \int_{\tilde{\Gamma}_{\text{root}}} v, \quad (3.107)$$

3.3.2 Reduction to Abstract Form

The formulation (3.107) is now very close to the abstract formulation of Section 3.1. The only complication – as we can observe from Figure 3-4 – is that the size and shape of the thermal fin (and hence $\tilde{\Omega}(\mu)$ and its boundary) depend on $\alpha_i, \beta_i, i = 1, \dots, 5$.

Applying the geometric mappings identical to those presented in Section 2.4 we recover the abstract formulation (3.3) for the locally non-affine model problem: $\forall w, v \in Y$ we decompose $a(w, v; \mu)$ in the following way

$$a(w, v; \mu) = \int_{\Omega_0} \nabla w \nabla v + \text{Bi} \int_{\Gamma_{\text{ext}} \cap \partial\Omega_0 \setminus \Gamma_{\text{root}}} w v + \sum_{i=1}^4 \left(\frac{\alpha_i}{\alpha_{\text{ref}}} \int_{\Omega_{0,i}} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\alpha_{\text{ref}}}{\alpha_i} \int_{\Omega_{0,i}} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} + \right)$$

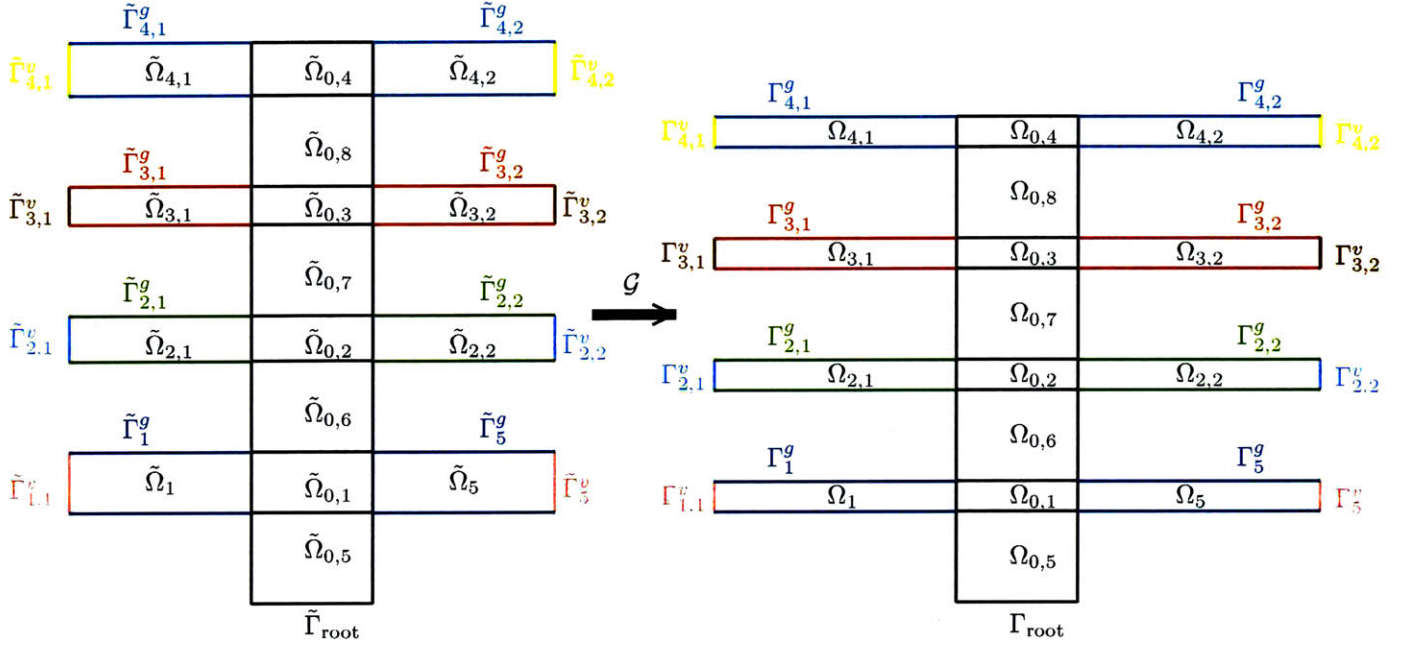


Figure 3-5: 2D locally non-affine model problem: the mapping $\mathcal{G}(\mu) : \tilde{\Omega}(\mu) \rightarrow \Omega$ between the varying shape domain $\tilde{\Omega}$ (left) and fixed reference domain Ω (right) and their respective boundaries.

$$\begin{aligned}
& \left. k_i \frac{\beta_{ref} \alpha_i}{\alpha_{ref} \beta_i} \int_{\Omega_i} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + k_i \frac{\alpha_{ref} \beta_i}{\beta_{ref} \alpha_i} \int_{\Omega_i} \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} + \text{Bi} \frac{\beta_i}{\beta_{ref}} \int_{\Gamma_i^g} wv + \text{Bi} \frac{\alpha_i}{\alpha_{ref}} \int_{\Gamma_i^v} wv \right) + \\
& \underbrace{\frac{\beta_{ref} \alpha_i}{\alpha_{ref} \beta_1} \int_{\Omega_{na}=\Omega_5} k_5(\underline{x}; \mu) \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\alpha_{ref} \beta_1}{\beta_{ref} \alpha_1} \int_{\Omega_{na}=\Omega_5} k_5(\underline{x}; \mu) \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2} + \text{Bi} \frac{\beta_1}{\beta_{ref}} \int_{\Gamma_5^g} wv + \text{Bi} \frac{\alpha_i}{\alpha_{ref}} \int_{\Gamma_5^v} wv}_{a^{NA}(w, x; \mu)}
\end{aligned} \tag{3.108}$$

where $\Omega_i = \Omega_{i,1} \cup \Omega_{i,2}$, $i = 2, \dots, 4$ and Γ_i^h, Γ_i^v are the vertical and the horizontal parts of the exterior boundary of Ω_i , respectively. Thus, the total number of affine parameters Q is equal to 26.

The affine part of the decomposition is almost identical to the one presented for the affine model problem of Chapter 2; the slight dissimilarity occurs from the fact that in the non-affine case the region $\tilde{\Omega}_1$ is defined slightly differently than in Chapter 2; (see Figures 2-2, 2-3 and 3-4, 3-5).

Consistent with (3.3), the locally non-affine contribution in the decomposition (3.108) is given by

$$a^{NA}(w, v; \mu) = \frac{\beta_{ref} \alpha_i}{\alpha_{ref} \beta_1} \int_{\Omega_{na}=\Omega_5} k_5(\underline{x}; \mu) \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} + \frac{\alpha_{ref} \beta_1}{\beta_{ref} \alpha_1} \int_{\Omega_{na}=\Omega_5} k_5(\underline{x}; \mu) \frac{\partial w}{\partial x_2} \frac{\partial v}{\partial x_2}. \tag{3.109}$$

Evaluation of (3.109) requires integration of $k_5(\underline{x}; \mu)$ over the finite element discretization; we perform this integration using 13 point Gaussian quadrature [7] which is exact for polynomials up to degree 7.

3.3.3 Numerical Results

Effect of Non-affine Parameter τ

We now demonstrate that the non-affine parameter τ plays an important role in our model problem. In Figure 3-6 we plot the field solution $u(\mu)$ for four different values of μ . We observe from Figure 3-6 that the temperature distribution in Ω_{na} exhibits significant variation which is explained by the functional dependence (3.100).

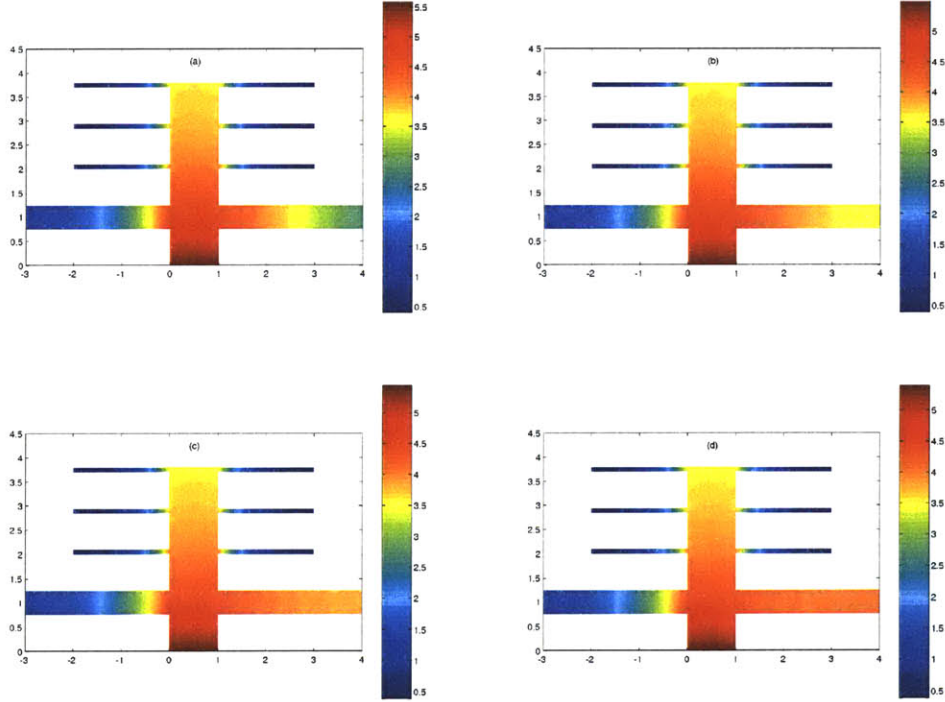


Figure 3-6: 2D affine model problem: the field solution $u(\mu)$ for

- (a) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, -0.7571\}$,
- (b) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, -0.2524\}$,
- (c) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, 0.2524\}$,
- (d) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 3.00, 0.50, 2.00, 0.10, 2.00, 0.10, 2.00, 0.10, 0.7571\}$.

We continue the illustration of the importance of the non-affine parameter τ by presenting the plot of u_{root} , the mean temperature along Γ_{root} , as a function of τ while having all other parameters

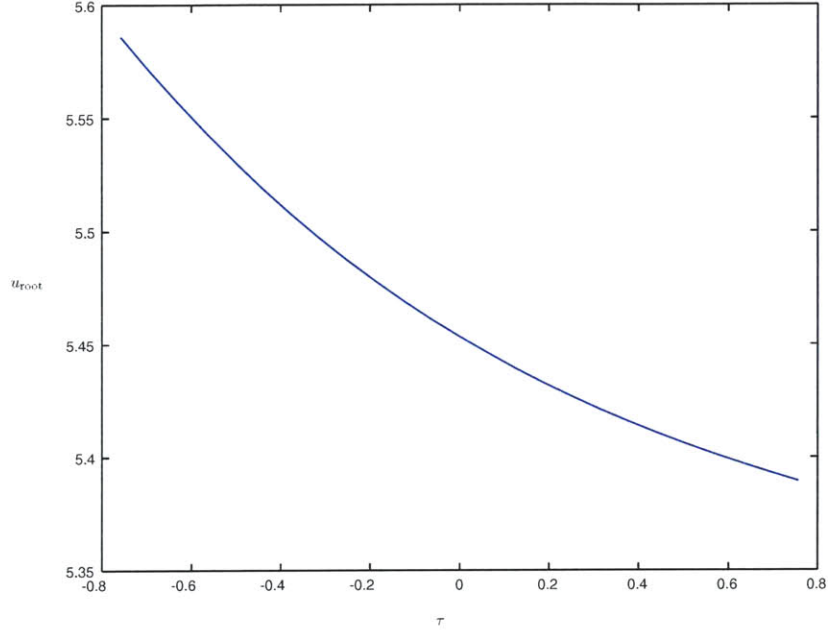


Figure 3-7: 2D locally non-affine model problem: variation of u_{root} as a function of τ , $k_i = 0.10$, $i = 1, \dots, 4$, $\text{Bi} = 0.01$, $\beta_1 = 3.00$, $\alpha_1 = 0.50$, $\beta_i = 2.00$, $\alpha_i = 0.10$, $i = 2, \dots, 4$.

fixed. As we observe from Figure 3-7, just by varying σ we achieve significant output variation.

We next present the actual numerical results obtained by application of the partition of unity reduced basis method to the heat transfer model we just introduced. To obtain the “truth” solution we consider a finite element mesh that consists of 21802 points. We then randomly generate a big pool of points $S_{N_{max}}$, $N_{max} = 100000$ from which we choose N points to form S_N .

For the problem of our choice the field solution displays considerable dependence on all the 14 parameters of μ . In general, the average temperature of the root, $s(\mu)$ will have slightly stronger dependence on k_1 and k_2 rather than k_4 , due to the fact that some of the heat already dissipates to the environment before it reaches the top part of the fin.

After we finish selecting parameter points for S_N we construct the coarse mesh T_S as shown in Figure 3-8. We then build the off-line/on-line framework which allows us to obtain the approximation $u_N(\mu)$ for the field solution $\tilde{u}_N(\mu)$ and the error bound $\Delta_{NS}(\mu)$ for $\|u(\mu) - u_N(\mu)\|$. We do not focus on the linear outputs of interest since the estimates for them can be obtained using the standard procedures as it was demonstrated in Section 2.3.3.

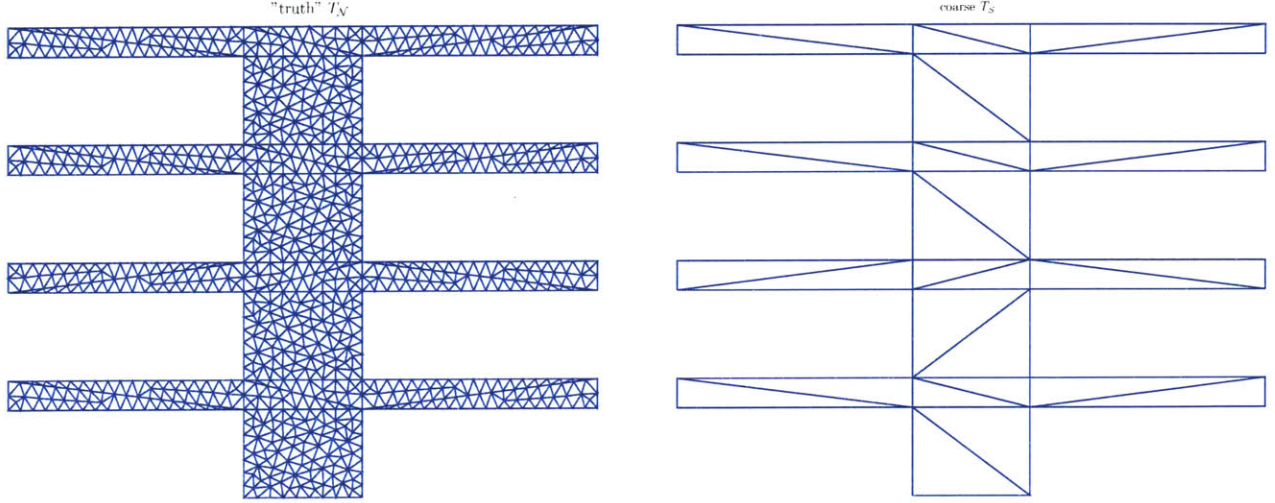


Figure 3-8: 2D locally non-affine model problem: relation between T_N , the "truth" discretization, and T_S , the coarse discretization.

Convergence

As described in (3.30), we introduce a bound conditioner $c_i(\mu)\hat{a}_i(w, v)$ that is equal to

$$\hat{a}_i(w, v) = \sum_{q=1}^{Q_i} \bar{\Theta}_i^{\mathcal{P}_i^a} a^q(w, v), \quad (3.110)$$

where $\bar{\Theta}_q^{\mathcal{P}_i^a} = \min_{\mu \in \mathcal{D}} \Theta_q^{\mathcal{P}_i^a}(\mu)$, $q = 1, \dots, Q_i$, $i = 1, \dots, J$. We note that this bound conditioner is independent of μ , the description of the more sophisticated bound conditioners could be found in [52, 57]. For this choice of bound conditioners the constants – as given in (2.22) – assume the following values: $\alpha_0 = 0.0256$, $\gamma_0 = 37.4555$.

We test convergence by randomly selecting a sample $S_{test} = \{\mu_1^t, \dots, \mu_{N_{test}}^t\}$. Unless specifically stated, we present all the quantities as functions of N averaged over S_{test} , a sample consisting of $N_{test} = 100000$ randomly chosen parameter points, which allows us to omit the parametric dependence. For example, instead of $e_N(\mu)$ we are going to consider e_N - the average "truth" error of the reduced basis approximation averaged over the sample S_{test} . In Figure 3-9 we provide the convergence plots for $a_N(e_N, e_N)$, $\|e_N\|$ and the error bounds for these quantities: $\tilde{\Delta}_{NS}$ and Δ_{NS} , respectively, as functions of N . As we observe from Figure 3-9, $a(e_N, e_N)$, $\tilde{\Delta}_{NS}$, and Δ_{NS} all decrease with N . We see that the accuracy is better for the pair $a(e_N, e_N)$, $\tilde{\Delta}_{NS}$ rather than for the pair $\|e_N\|$, Δ_{NS} which is explained by the quadratic convergence properties of $a_N(e_N, e_N)$, $\tilde{\Delta}_{NS}$.

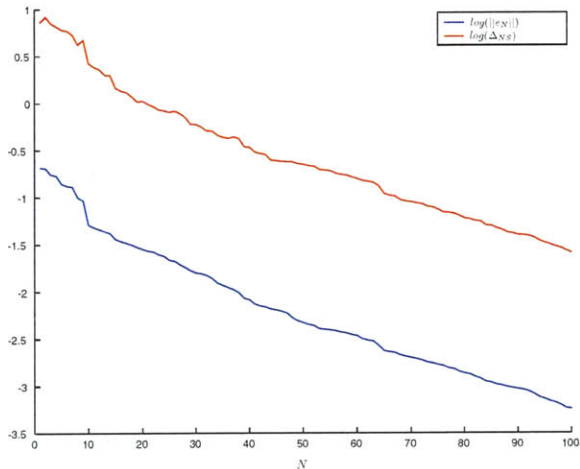


Figure 3-9: Partition of unity method, 2D locally non-affine model problem: $\|e_N\|_Y$ and Δ_N as functions of N .

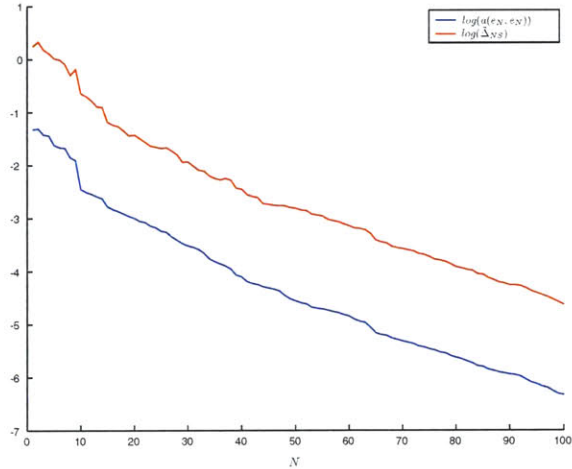


Figure 3-10: Partition of unity method, 2D locally non-affine model problem: $a(e_N, e_N)$ and $\tilde{\Delta}_N$ as functions of N .

To measure the convergence with respect to the field solution we present the results for the relative rate of convergence: in Table 3.2 we provide the values for $\frac{a(e_N, e_N)}{a(u, u)}$, $\frac{\tilde{\Delta}_{NS}}{a(u, u)}$, $\frac{\|e_N\|}{\|u\|}$, and $\frac{\Delta_{NS}}{\|u\|}$.

We note from Figure 3-9 and Table 3.2 that even for moderate values of N we reach good accuracy both for $a_N(e, e)$, $\tilde{\Delta}_{NS}$ and $\|e_N\|$, Δ_N ; this level accuracy is by all means acceptable in the engineering practice.

Effectivity

In order to demonstrate that our error bounds are, indeed, sharp in Figure 3-11 we provide the plots for the effectivities $\tilde{\eta}_N$ and η_N as defined in (3.44) and (3.45). We observe that the values effectivities are slightly higher than in the 1D case which we studied in Section 3.2.3. In the 1D case we had only one operator, that is the Laplacian, in the bilinear form $a(., .; .)$ which allowed us to dynamically choose a multiplier $c_i(\mu)$ in (3.61), (3.62), and (3.63) for each new value of μ . This is no longer the case for the 2D problem where the bilinear form decomposition over each $\mathcal{P}_i, i = 1, \dots, S$ is far more complex. We then use a single point bound conditioner [52, 56] (as we described it on the previous page) and this approach yields slightly less sharp error prediction than the approach we used in the 1D case. One possible way to decrease the effectivity values is to use more sophisticated bound conditioners [52, 56].

Even for the current rather simplistic choice of a bound conditioner the effectivities $\tilde{\eta}_N$ and η_N

N	$\frac{\ e_N\ _Y}{\ u\ _Y}$	$\frac{\Delta_N}{\ u\ _Y}$	$\frac{a(e_N, e_N)}{a(u, u)}$	$\frac{\tilde{\Delta}_N(\mu)}{a(u, u)}$
5	1.38e-01	5.83e+00	2.65e-02	1.07e+00
10	4.77e-02	2.45e+00	3.42e-03	2.10e-01
15	3.27e-02	1.32e+00	1.50e-03	5.82e-02
20	2.64e-02	9.47e-01	9.48e-04	3.15e-02
25	2.05e-02	7.29e-01	5.59e-04	1.84e-02
30	1.54e-02	5.39e-01	3.13e-04	1.00e-02
35	1.14e-02	4.11e-01	1.60e-04	5.45e-03
40	7.86e-03	3.16e-01	7.89e-05	3.30e-03
45	6.15e-03	2.27e-01	4.92e-05	1.71e-03
50	4.42e-03	2.07e-01	2.69e-05	1.43e-03
55	3.70e-03	1.78e-01	1.90e-05	1.03e-03
60	3.23e-03	1.49e-01	1.44e-05	7.04e-04
65	2.19e-03	9.97e-02	6.46e-06	3.46e-04
70	1.84e-03	8.08e-02	4.46e-06	2.33e-04
75	1.55e-03	6.69e-02	3.18e-06	1.62e-04
80	1.25e-03	5.40e-02	2.11e-06	1.04e-04
85	9.86e-04	4.50e-02	1.38e-06	7.33e-05
90	8.43e-04	3.60e-02	9.90e-07	4.68e-05
95	6.52e-04	2.94e-02	6.38e-07	3.25e-05
100	5.14e-04	2.29e-02	4.01e-07	1.94e-05

Table 3.2: Partition of unity method, 2D locally non-affine model problem: relative convergence of $\|e_N\|_Y$, Δ_N and $a(e, e)$, $\tilde{\Delta}_N$ with respect to N .

stay in the range [35,65] which is already a reasonably sharp error bound.

Efficient Partition of Unity Selection

As it was shown in Section 3.2.4, the construction of the error bound $\Delta_{NS}(\mu)$ requires solving G systems of linear equations. The size of each of these systems is equal to the number of "truth" finite element test functions in each of the regions $\mathcal{P}_i^{na}, i = 1, \dots, G$. In order to keep the computational cost as low as possible we want the size of the regions \mathcal{P}_i^{na} to be as small – or, in other words, as close to n_{na} – as possible. This can be achieved by reducing the intersection of \mathcal{P}_i^{na} with the part of Ω where we have affine parametric dependence, albeit at a slight increase in the number of "affine" partitions and hence the total number of partitions. We refine the partitions around the border between the areas with affine and non-affine dependence as it is shown in Figure 3-12. In general, the efficient partition of unity selection is a problem specific task which depends on the operators that define the problem and the geometry of the problem. At the same we a user performs such a

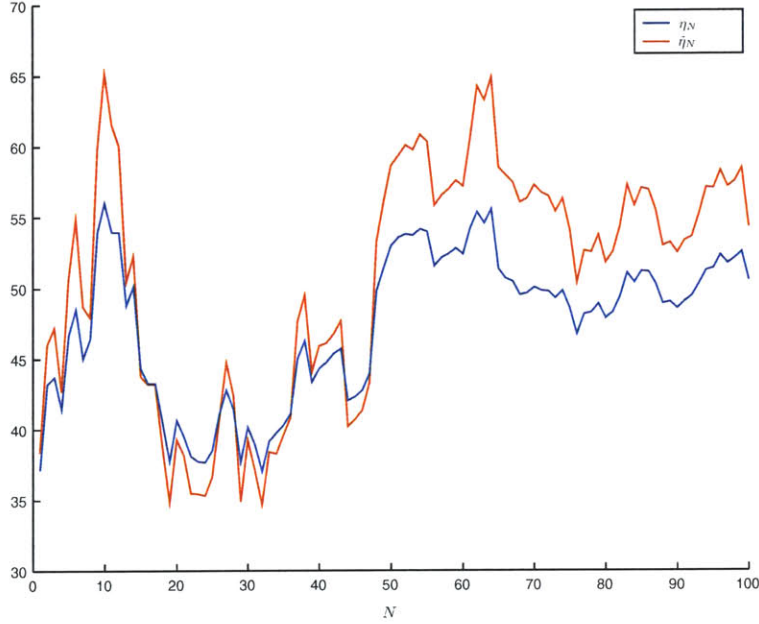


Figure 3-11: Partition of unity method, 2D locally non-affine model problem: effectivities $\tilde{\eta}_N$ and η_N as functions of N .

partition refinement he or she should take into account that the provable upper bound might suffer because of that. This definitely was not an issue in the 1D case (see Section 3.2.3), however in higher dimensions an additional check is required.

In some cases (such as the case of the 2D model problem described in Section 3.3) it is possible to improve the efficiency of the reduced basis method by considering only *one* "non-affine" partition (see Section 3.2.4 for details). We replace all the partitions $\Phi_i^{na}, i = 1, \dots, G$ by their sum $\sum_{i=1}^G \Phi_i$. With this approach in order to provide an error bound we now only have to solve one linear system instead of G (see the numerical algorithm in 3.2.4), however the size of this system is slightly bigger than the size of all linear systems corresponding to \mathcal{P}_i^{na} . The numerical results for the 2D problem are provided for the "non-optimized" partition of unity as shown in Figure 3-8 where all the non-affine partitions are lumped into one as we just discussed.

Computational Costs

In Table 3.3 we provide the dependence on N of t_{u_N} , t_{δ_S} , and Δ_{NS} , the actual computational times required to obtain the reduced basis approximation u_N , the partition of unity contribution δ_S , and the error bound Δ_{NS} , respectively, during the on-line stage. The fifth column of Table 3.3 contains t_{tot} , the total time which is necessary to run the on-line stage for a single parameter value, as a

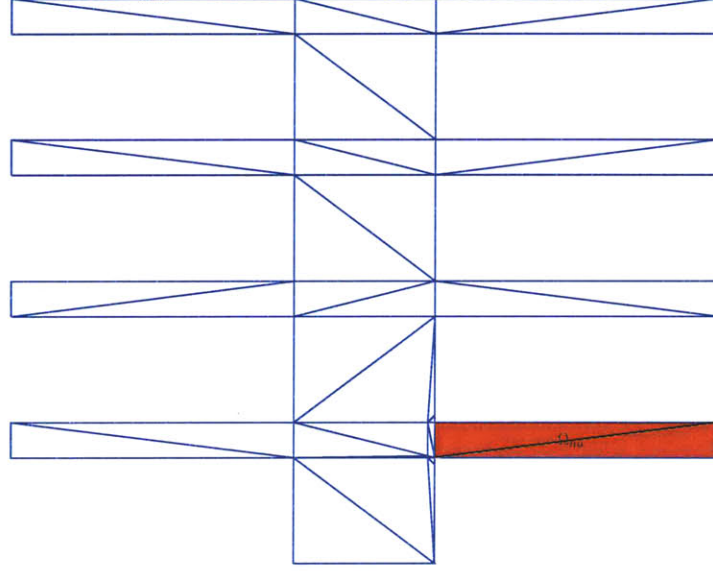


Figure 3-12: Partition of unity method: the idea behind efficient partition of unity selection.

function of N . The value of t_{tr} , the time required to obtain the "truth" finite element solution is given in the rightmost column of the same Table.

Obviously, t_{tr} does not depend on N at all. As we can see from Table 3.3, t_{u_N} , t_{δ_S} , $t_{\Delta_{NS}}$ and hence t_{tot} are all increasing functions of N . The main contribution to t_{tot} is provided by $t_{\Delta_{NS}}$ which is consistent with its operation count of $O(\sum_{i=1}^J Q_i^2(S_i + N)^2) + O(n_{na}SN) + O(Gn_{na}^\kappa)$ (for our case $G = 1$). The operation count has two terms which account for the construction of Δ_{NS} from the affine and non-affine partitions. The fact that $t_{\Delta_{NS}}$ - unlike t_{u_N} and t_{δ_S} - is not close to zero when N is small is explained by the presence of $O(n_{na}SN) + O(Gn_{na}^\kappa)$ term in the operation count for Δ_{NS} (note that this term has only linear dependence on N). The growth of $t_{\Delta_{NS}}$ with N is mainly explained by the affine component of the operation count of Δ_{NS} : $O(\sum_{i=1}^J Q_i^2(S_i + N)^2)$.

We also note that t_{δ_S} displays a somewhat weak dependence on N which is consistent with its operation count of $O(QS) + O(n_{na}S^2) + O(QSN) + O(n_{na}SN) + O(S^\kappa)$ which is linear in N . We observe that t_{u_N} grows with N at approximately the same rate as $t_{\Delta_{NS}}$, this behavior is supported by the operation count to obtain \underline{u}_N which is equal to $O(QN^2) + O(n_{na}N^2) + O(N^3)$.

For this particular configuration even for the largest values of N ($N = 100$) we observe computational savings (measured by the efficiency ratio $\frac{t_{tr}}{t_{tot}}$) of approximately 7.

In general, we have noticed that the efficiency ratio of the reduced reduced basis method is an increasing function of \mathcal{N} , the number of nodes in the "truth" mesh.

N	t_{u_N}, s	t_{δ_S}, s	$t_{\Delta_{NS}}, s$	t_{tot}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tr}, s
5	5.90e-03	2.80e-02	9.02e-02	1.24e-01	28.37	3.52
10	1.02e-02	3.13e-02	9.39e-02	1.35e-01	26.01	3.52
15	1.57e-02	3.17e-02	9.86e-02	1.46e-01	24.11	3.52
20	2.11e-02	3.29e-02	1.05e-01	1.59e-01	22.20	3.52
25	2.70e-02	3.41e-02	1.11e-01	1.72e-01	20.44	3.52
30	3.25e-02	3.52e-02	1.19e-01	1.87e-01	18.82	3.52
35	4.03e-02	3.68e-02	1.28e-01	2.05e-01	17.16	3.52
40	4.71e-02	3.79e-02	1.37e-01	2.22e-01	15.89	3.52
45	5.60e-02	4.01e-02	1.46e-01	2.42e-01	14.54	3.52
50	6.46e-02	4.12e-02	1.56e-01	2.62e-01	13.43	3.52
55	7.30e-02	4.28e-02	1.67e-01	2.83e-01	12.44	3.52
60	8.21e-02	4.44e-02	1.78e-01	3.05e-01	11.54	3.52
65	8.90e-02	4.58e-02	1.89e-01	3.24e-01	10.86	3.52
70	9.92e-02	4.76e-02	2.01e-01	3.48e-01	10.12	3.52
75	1.09e-01	4.86e-02	2.11e-01	3.68e-01	9.56	3.52
80	1.20e-01	4.98e-02	2.23e-01	3.92e-01	8.97	3.52
85	1.31e-01	5.11e-02	2.35e-01	4.17e-01	8.45	3.52
90	1.47e-01	5.24e-02	2.48e-01	4.47e-01	7.87	3.52
95	1.61e-01	5.38e-02	2.63e-01	4.78e-01	7.37	3.52
100	1.73e-01	5.54e-02	2.80e-01	5.08e-01	6.92	3.52

Table 3.3: Partition of unity method, 2D locally non-affine model problem: on-line computational times.

Conclusions

Using this model problem we were able to show that the partition of unity method is capable of providing very accurate results as it was demonstrated in Figure 3-9 and Table 3.2. At the same time our error bounds remain quite sharp which is confirmed by the effectivity plots in Figure 3-11. Both $\tilde{\eta}_N$ and η_N remain within the range of [35,65] which indeed is quite a good result. Even for the highest value of N considered ($N = 100$) the total time for the on-line stage is approximately 5 times less than the time to obtain the truth solution. If we refine the "truth" mesh we can only expect this ratio to increase, thus making the partition of unity method even more attractive for our purposes.

3.4 Application of Partition of Unity Method to Affine Coercive Problems

The partition of unity method was originally developed by us to address locally non-affine problems as described in Section 3.2. At the same time it might be of interest to apply the PUM to the purely affine problems which potentially might yield better results than the classical reduced basis method. We will describe the PUM numerical algorithm for the affine case which in fact is a simplification of the locally non-affine case. To avoid unnecessary repetition we will proceed directly to the numerical algorithm. Later we provide a criterion under which the partition of unity method can provide better performance than the classical reduced basis method.

3.4.1 General Abstract Problem Statement

Just as a reminder, the original problem is given by: find $u(\mu) \in Y$

$$a(u(\mu), v; \mu) = f(v), \forall v \in Y, \quad (3.111)$$

where $a(\cdot, \cdot; \mu)$ admits the following decomposition

$$a(w, v; \mu) = \sum_{i=1}^Q \Theta_q(\mu) a^q(w, v), \quad \forall w, v \in Y, \quad (3.112)$$

where Q is the parametric complexity of the problem and $a^q(w, v), q = 1, \dots, Q$ is a parameter independent bilinear form.

3.4.2 Reduced Basis Approximation

We consider an affine problem that satisfies all the regularity conditions of Chapter 2.

$$S_N = \{\mu_1, \dots, \mu_N\}, \quad (3.113)$$

where $\mu_i \in \mathcal{D}, i = 1, \dots, N$. We then define our reduced-basis approximation space W_N as

$$W_N = \text{span}\{\zeta_i \equiv u(\mu_i), i = 1, \dots, N\}, \quad (3.114)$$

where $u(\mu_i) \in Y$ is the solution of (3.111) for $\mu = \mu_i$. For any $\mu \in \mathcal{D}$, we define our reduced-basis approximation for the field solution as $u_N(\mu)$ which is obtained through Galerkin projection of $u(\mu)$ onto W_N

$$a(u_N(\mu), v) = f(v), \quad \forall v \in W_N, \quad (3.115)$$

$$u_N(\mu) = \sum_{j=1}^N u_{Nj}(\mu) \zeta_j. \quad (3.116)$$

We then look for $\delta_S(\mu) \in X_S$ given by

$$a(u_N(\mu) + \delta_S(\mu), v; \mu) = f(v), \quad \forall v \in X_S, \quad (3.117)$$

$$\delta_S(\mu) = \sum_{k=1}^S \delta_{S_k}(\mu) \Phi_k. \quad (3.118)$$

We consider $\tilde{u}_N(\mu) = u_N(\mu) + \delta_S(\mu)$ as the approximation to $u(\mu)$. The matrix vector form of (3.115) is given by

$$\underline{A}_N(\mu) \underline{u}_N(\mu) = \underline{F}_N, \quad (3.119)$$

where $\underline{A}_N(\mu)$ admits the decomposition consistent with (2.11), (3.112)

$$\underline{A}_N(\mu) = \sum_{q=1}^Q \Theta_q(\mu) \underline{A}_N^q, \quad (3.120)$$

where

$$\underline{A}_{N_{ij}}^q = a^q(\zeta_i, \zeta_j), \underline{F}_{N_i} = f(\zeta_i), q = 1, \dots, Q, i, j = 1, \dots, N. \quad (3.121)$$

We now turn to the computation of $\delta_S(\mu)$ from (3.22); this requires solution of an $S \times S$ linear system of equations

$$\underline{A}_S(\mu) \underline{\delta}_S(\mu) = \underline{F}_S(\mu) - \underline{A}_{SN}(\mu) \underline{u}_N(\mu), \quad (3.122)$$

where

$$\underline{F}_{S_i} = f(\Phi_i), i = 1, \dots, S; \quad (3.123)$$

and where matrices $\underline{A}_S(\mu), \underline{A}_{SN}(\mu)$ admit the following decomposition

$$\underline{A}_S(\mu) = \sum_{i=1}^Q \Theta_q(\mu) \underline{A}_S^q, \quad (3.124)$$

$$\underline{A}_{SN}(\mu) = \sum_{i=1}^Q \Theta_q(\mu) \underline{A}_{SN}^q, \quad (3.125)$$

where, in turn, for $q = 1, \dots, Q$

$$\underline{A}_{S_{ij}}^q = a^q(\Phi_i, \Phi_j), \quad i, j = 1, \dots, N, \quad (3.126)$$

$$\underline{A}_{SN_{ij}}^q = a^q(\Phi_i, \zeta_j), \quad i = 1, \dots, S, j = 1, \dots, N. \quad (3.127)$$

The vector $\underline{d}_S(\mu)$ is obtained by solution of (3.85). We note that the system (3.122) is not only low-dimensional – since S is typically quite small – but also sparse because it corresponds to the finite element discretization T_S . These two facts suggest that (3.122) is very cheap to solve: depending on the nature of the bilinear form $a(\cdot, \cdot; \cdot)$ the operation count for solution of (3.122) is expected to be $O(S^\kappa)$ where κ is equal to or slightly greater than 1.

3.4.3 A Posteriori Error Estimation

The construction of the error bound almost exactly – with the exception of the non-affine component – follows the construction of the error bound for the locally non-affine case. Essentially we provide a simplification of the algorithm presented in Section 3.2.1: in the purely affine case there is no need to introduce two sets of partitions, therefore we can drop the superscripts in $\Phi_i, i = 1, \dots, S$.

We start by introducing the error function $e_N(\mu) \in Y$ as

$$e_N(\mu) = u(\mu) - \tilde{u}_N(\mu). \quad (3.128)$$

We then introduce spaces $Y_i = H^1(\bar{\mathcal{P}}_i)$ such that the derivatives of functions v in the regularity constraints for H^1 spaces are evaluated from inside of the domain $\bar{\mathcal{P}}_i^{a|na}$ and the mapping operators $\mathcal{I}_i : Y \rightarrow Y_i$ such that

$$\forall v \in Y \quad \mathcal{I}_i v = \bar{v} \in Y_i : \bar{v} = v|_{\bar{\mathcal{P}}_i}, \quad i = 1, \dots, S. \quad (3.129)$$

We also define a bilinear form $a_{\mathcal{P}_i}(\cdot, \cdot; \cdot) : (Y_i)^2 \rightarrow \mathbb{R}$ and linear functional $f_{\mathcal{P}_i}(\cdot) \in (Y_i)'$ such

that

$$\forall w, v \in Y : w|_{\Omega \setminus \mathcal{P}_i} = 0, v|_{\Omega \setminus \mathcal{P}_i} = 0$$

$$a_{\mathcal{P}_i}(\mathcal{I}_i w, \mathcal{I}_i v; \mu) = a(w, v; \mu), \quad (3.130)$$

$$f_{\mathcal{P}_i}(\mathcal{I}_i v) = f(v). \quad (3.131)$$

We next define a function $c_i(\mu) : \mathcal{D} \rightarrow \mathbb{R}$ and a bilinear form $\hat{a}_i(.,.) : (Y_i^a)^2 \rightarrow \mathbb{R}$ such that

$$c_i(\mu) \hat{a}_i(v, v) \leq a_{\mathcal{P}_i}(v, v; \mu), \quad \forall \mu \in \mathcal{D}, \forall v \in Y_i; \quad (3.132)$$

we note that though we require coercivity of $a(.,.,.)$ we do not require the same for each $a_{\mathcal{P}_i}(.,.,.)$. Clearly, $0 \leq \hat{a}_i(v, v) \quad \forall v \in Y_i$, however it may well happen that for some $v \in Y_i$ $a_{\mathcal{P}_i}(v, v; \mu) = 0$. If this is the case, we need to make sure that $\hat{a}_i(v, v) = 0$ as well, so that (3.132) holds.

Following the same approach as in Section 3.2.3 we introduce "local" residual functionals $R_i(v; \mu) \in (Y_i)'$ (compare to (2.19)) as follows

$$R_i(v; \mu) = f_{\mathcal{P}_i}(v) - a_{\mathcal{P}_i}(\mathcal{I}_i \tilde{u}_N(\mu), v; \mu). \quad (3.133)$$

We subsequently look for error functions $\hat{e}_i(\mu) \in Y_i$ as the solutions of the following local error equations:

$$c_i(\mu) \hat{a}_i(\hat{e}_i(\mu), v) = R_i(\mathcal{I}_i \Phi_i v; \mu), \quad \forall v \in Y_i; i, \dots, S. \quad (3.134)$$

The error bound $\tilde{\Delta}_{NS}(\mu)$ is then defined as follows

$$a(e_N(\mu), e_N(\mu); \mu) \leq \tilde{\Delta}_{NS}(\mu) = t \sum_{i=1}^S c_i(\mu) \hat{a}_i(\hat{e}_i(\mu), \hat{e}_i(\mu); \mu). \quad (3.135)$$

We are going to omit the proof of the bounding properties of $\Delta_{NS}(\mu)$ since it is almost identical to the proof of Section 3.2.3.

For each of the partitions $\Phi_i, i = 1, \dots, S$ we denote a set $\{a_{\mathcal{P}_i}^1(.,.), \dots, a_{\mathcal{P}_{Q_i}}^{Q_i}(.,.)\}$ of bilinear forms for which there exists a function $v \in Y_i$ such that $v|_{\mathcal{P}_i} \neq 0$, the set of affine coefficients corresponding to $a_{\mathcal{P}_i}^j(.,.), j = 1, \dots, Q_i$, is then denoted as $\{\Theta_1^{\mathcal{P}_i}, \dots, \Theta_{Q_i}^{\mathcal{P}_{Q_i}}\}$. Clearly, $Q_i \leq Q$,

$\forall i = 1, \dots, J$. We then define a set $\{\Phi_1^{\mathcal{P}_i}, \dots, \Phi_{S_i}^{\mathcal{P}_i}\}$ of partitions the support of which has a non-zero intersection with \mathcal{P}_i . We denote the projection operators and the coefficients in (3.118) corresponding to $\Phi_j^{\mathcal{P}_i}$ as $\mathcal{I}_j^{\mathcal{P}_i}, \delta_j^{\mathcal{P}_i}, j = 1, \dots, S_i$.

We next note that for each patch \mathcal{P}_i (3.133), the expression for $R_i(\mathcal{I}_i^{\mathcal{P}_i} \Phi_i v; \mu), \forall v \in Y_i, i = 1, \dots, S$ can be rewritten as

$$R_i(\mathcal{I}_i^{\mathcal{P}_i} \Phi_i v; \mu) = f_{\mathcal{P}_i}(\mathcal{I}_i^{\mathcal{P}_i} \Phi_i v) - \sum_{q=1}^{Q_i} \Theta_q^{\mathcal{P}_i} \left(\sum_{l=1}^N u_{N_l} a_{\mathcal{P}_i}^q(\mathcal{I}_l \zeta_l, \mathcal{I}_i^{\mathcal{P}_i} \Phi_i v) + \sum_{k=1}^{S_i} \delta_{S_k}^{\mathcal{P}_i} a_{\mathcal{P}_i}^q(\mathcal{I}_k^{\mathcal{P}_i} \Phi_k^{\mathcal{P}_i}, \mathcal{I}_i^{\mathcal{P}_i} \Phi_i v) \right). \quad (3.136)$$

We next renumber (3.136) and rewrite it in the following form:

$$\sum_{j=1}^{K_i} \tau_j^i(\mu) \Upsilon_j^i(v), \quad (3.137)$$

where $K_i = 1 + Q_i(N + S_i)$ and

$$\tau_1^i(\mu) = 1, \quad \Upsilon_1^i(v) = f_{\mathcal{P}_i}(\mathcal{I}_1^{\mathcal{P}_i} \Phi_i v); \quad (3.138)$$

for $j = 2, \dots, 1 + Q_i N : j = 1 + (q-1)N + l, q = 1, \dots, Q_i, l = 1, \dots, N$

$$\tau_j^i(\mu) = -\Theta_q^{\mathcal{P}_i}(\mu) u_{N_l}(\mu), \quad \Upsilon_j^i(v) = a_{\mathcal{P}_i}^q(\mathcal{I}_l \zeta_l, \mathcal{I}_i^{\mathcal{P}_i} \Phi_i v); \quad (3.139)$$

for $j = 2 + Q_i N, \dots, 1 + Q_i(N + S_i) : j = 1 + Q_i N + (q-1)S_i + k, q = 1, \dots, Q_i, k = 1, \dots, S_i$

$$\tau_j^i(\mu) = -\Theta_q^{\mathcal{P}_i}(\mu) \delta_{S_k}^{\mathcal{P}_i}, \quad \Upsilon_j^i(v) = a_{\mathcal{P}_i}^q(\mathcal{I}_k^{\mathcal{P}_i} \Phi_k^{\mathcal{P}_i}, \mathcal{I}_i^{\mathcal{P}_i} \Phi_i v). \quad (3.140)$$

Using (3.138), (3.139), (3.140), (3.135), and (3.136) it is a simple matter to demonstrate that

$$\Delta_{NS}(\mu) = \sum_{i=1}^J c_i(\mu) \hat{a}_i(\hat{e}_i(\mu), \hat{e}_i(\mu); \mu) = t \sum_{i=1}^J \frac{1}{c_i(\mu)} \sum_{j=1}^{K_i} \sum_{j'=1}^{K_i} \tau_j^i(\mu) \tau_{j'}^i(\mu) \underbrace{\hat{a}_i(z_i^j, z_i^{j'})}_{\Lambda_{jj'}^i}, \quad (3.141)$$

where $z_j^i, i = 1, \dots, J; j = 1, \dots, K_i = 1 + Q_i(N + S_i)$ is obtained as the solution of

$$\hat{a}_i(z_j^i, v) = \Upsilon_j^i(v), \quad \forall v \in Y_i^a, \quad (3.142)$$

$$\Lambda_{jj'}^i = \hat{a}_i(z_j^i, z_{j'}^i). \quad (3.143)$$

Therefore, (3.135) can be rewritten as

$$\Delta_{NS}(\mu) = t \sum_{i=1}^S \frac{1}{c_i(\mu)} \sum_{j=1}^{K_i} \sum_{j'=1}^{K_i} \tau_j^i(\mu) \tau_{j'}^i(\mu) \Lambda_{jj'}^i. \quad (3.144)$$

Numerical Algorithm

Off-line

1. Generate S_N as described in Section (2.3.4) and construct the basis for W_N ;
2. Precompute \underline{F}_N and \underline{A}_N^q as in (3.121), $q = 1, \dots, Q$;
3. Precompute \underline{F}_S as in (3.123) and $\underline{A}_S^q, \underline{A}_{S_N}^q$ as in (3.126), (3.127), $q = 1, \dots, Q$;
4. Precompute $\Lambda_{mm'}^i$ as in (3.143), $i = 1, \dots, S$; $m, m' = 1, \dots, 1 + Q_i(N + S_i)$.

On-line

1. Form \underline{A}_N as in (3.120) and solve (3.119) for $\underline{u}_N(\mu)$. Cost: $O(QN^2) + O(N^3)$;
2. Form \underline{A}_S as in (3.124), \underline{A}_{S_N} as in (3.125) and solve (3.122) for $\underline{\delta}_S(\mu)$. Cost: $O(QS) + Q(QSN) + O(S^\kappa)$;
3. Calculate $\Delta_{NS}(\mu)$ from (3.141). Cost: $O(\sum_{i=1}^S Q_i^2(S_i + N)^2)$.

$$\text{Total on-line cost: } O(QN^2) + O(N^3) + O(QS^2) + O(QSN) + O(S^\kappa) + \sum_{i=1}^S Q_i^2(N + S_i)^2.$$

As it was in the affine case in Chapter 2, the most dominating contribution is coming from the construction of the error bound. When making a decision which method to use for an affine problem it is important to compare two quantities: the operation count for the error bounds for the classical reduced basis method which is equal to $O(Q^2N^2)$ and the partition of unity methods which is equal to $O(\sum_{i=1}^S (Q_i(S_i + N))^2)$.

We readily notice that of the partition of unity method is advantageous over the classical reduced basis method with respect to the off-line stage: as we mentioned in Chapter 2 the main challenge for the CRBM is to form $\Lambda_{jj'}$ as given by (2.55) which requires solution of Q^2N^2 linear systems of size \mathcal{N} . The partition of unity significantly reduces this computational burden: to form $\Lambda_{mm'}^i$ as given by (3.143) for each $i = 1, \dots, S$ we need to solve $Q_i^2(N + S_i)^2$ linear system of size \mathcal{N}_i , the number of nodes in \mathcal{P}_i , and \mathcal{N}_i typically is $O(10)$ times smaller than \mathcal{N} .

For both methods we expect the calculation of the error bound to be the most computationally

expensive part of the on-line stage. From the operation counts provided in this Section and Section 2.3.5 it is not obvious right away which of the operation counts $O(\sum_{i=1}^S (Q_i(S_i + N)^2))$ or Q^2N^2 is going to be smaller. We revisit this issue in later in this Chapter.

3.4.4 Numerical Results in Comparison to Classical Reduced Basis Method

We now apply the partition of unity method to the problem we described in Section 2.4 and compare the numerical results for both methods. For the partition of unity method we select the partition of unity in the same way geometry-wise as it is shown in Figure 3-8 for the locally non-affine case.

Convergence

In Figure 3-13 we present the convergence plots for the norm of the error and the error bound both for the classical reduced basis method and the partition of unity method. As we can see from this Figure, the methods essentially demonstrate the same rate of convergence with respect to $\|e_N\|$. The error bound Δ_N for the CRBM is slightly smaller than the error bound Δ_{NS} for the PUM.

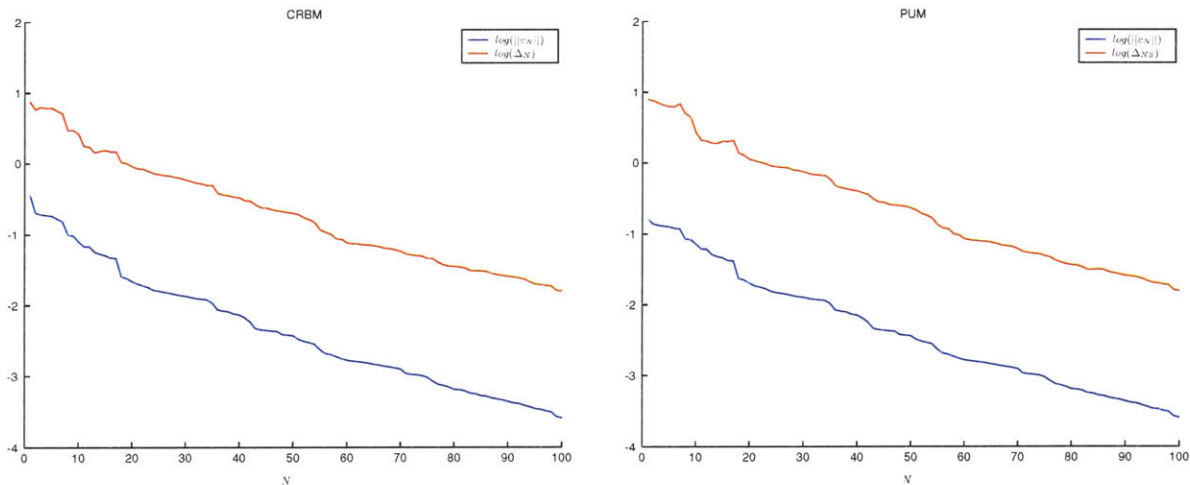


Figure 3-13: Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: $\|e_N\|_Y$ and Δ_N as functions of N .

Effectivities

Comparing the effectivity values for the two methods in Figure 3-14 we arrive at a conclusion that both methods provide sharp error bounds, the effectivities are slightly better (smaller) for the case of the CRBM which is also consistent with how the methods behave in Figure 3-13. This might be

explained by the distortions introduced into the error equations (3.134) by the partition of unity functions $\Phi_i, i = 1, \dots, S$.

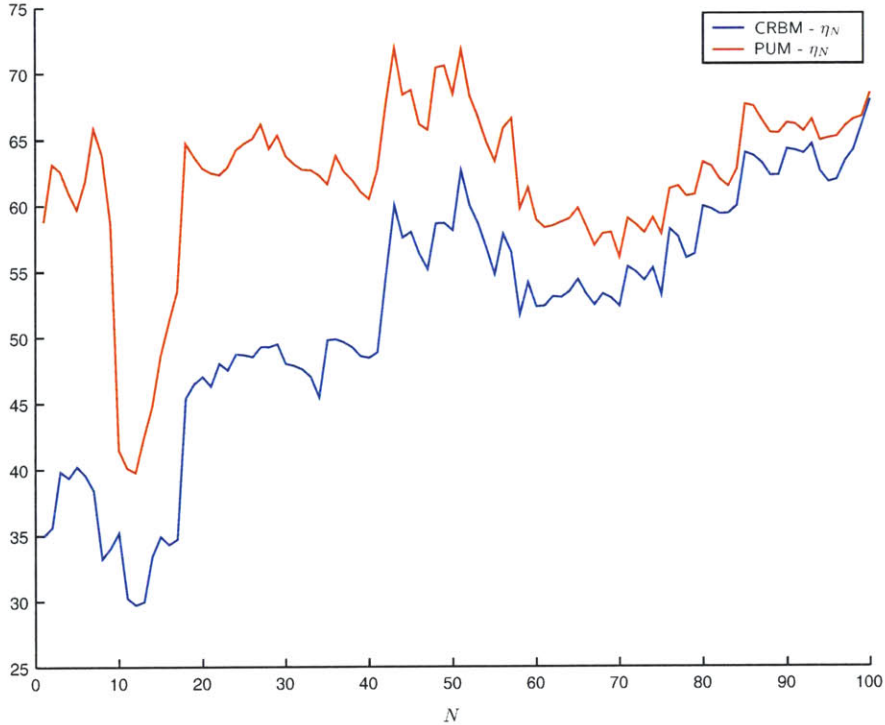


Figure 3-14: Comparison of classical reduced basis (blue) and partition of unity (red) methods, 2D affine model problem: effectivity η_N as a function of N .

Computational Costs

In Table 3.4 we provide the details of the decomposition of the total time required to run the on-line stage (which we denote t_{tot}) into t_{u_N} , t_{δ_S} , and $t_{\Delta_{NS}}$, the times required to obtain \underline{u}_N , $\underline{\delta}_S$, and Δ_{NS} , respectively. As expected, the main contribution to t_{tot} is provided by $t_{\Delta_{NS}}$ which is consistent with $t_{\Delta_{NS}}$ operation count $\sum_{i=1}^S (Q_i(N + S_i))^2$. Very similar to the non-affine case, t_{δ_S} exhibits the weakest dependence on N (out of t_{u_N} , t_{δ_S} , and $t_{\Delta_{NS}}$), whereas the dependence of t_{u_N} on N is quite strong; these two facts are consistent with operation counts for t_{δ_S} and t_{u_N} : $O(QS^2) + O(QSN) + O(S^\kappa)$ and $O(QN^2) + O(N^3)$, respectively.

In Tables 3.5 and 3.6 we provide the total on-line computational time for both methods as functions of $\|e_N\|$ and $\Delta_N(\Delta_{NS})$, respectively. We observe that in both Tables t_{tot} for the PUM is approximately twice as big as for the CRBM (we consider the ranges when the reduced basis approximations are reasonably accurate). For both the CRBM and the PUM methods the main

N	t_{u_N}, s	t_{δ_S}, s	$t_{\Delta_{NS}}, s$	t_{tot}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tr}, s
5	1.63e-03	4.79e-03	6.20e-03	1.26e-02	276.59	3.53
10	1.92e-03	5.20e-03	9.32e-03	1.64e-02	212.29	3.53
15	2.12e-03	5.30e-03	1.33e-02	2.07e-02	168.62	3.53
20	2.43e-03	5.31e-03	1.81e-02	2.59e-02	134.91	3.53
25	2.73e-03	5.63e-03	2.34e-02	3.18e-02	109.84	3.53
30	3.06e-03	5.79e-03	2.96e-02	3.85e-02	90.69	3.53
35	3.54e-03	5.89e-03	3.67e-02	4.62e-02	75.59	3.53
40	4.09e-03	6.14e-03	4.44e-02	5.46e-02	63.88	3.53
45	4.75e-03	6.34e-03	5.41e-02	6.52e-02	53.52	3.53
50	5.36e-03	6.59e-03	6.33e-02	7.53e-02	46.35	3.53
55	5.97e-03	6.67e-03	7.40e-02	8.66e-02	40.28	3.53
60	7.41e-03	7.61e-03	8.52e-02	1.00e-01	34.84	3.53
65	8.17e-03	7.69e-03	9.71e-02	1.13e-01	30.89	3.53
70	9.82e-03	7.85e-03	1.10e-01	1.27e-01	27.39	3.53
75	1.16e-02	8.17e-03	1.23e-01	1.43e-01	24.38	3.53
80	1.52e-02	8.05e-03	1.38e-01	1.61e-01	21.67	3.53
85	1.69e-02	9.44e-03	1.53e-01	1.79e-01	19.49	3.53
90	1.99e-02	8.63e-03	1.69e-01	1.97e-01	17.70	3.53
95	2.38e-02	9.45e-03	1.87e-01	2.20e-01	15.87	3.53
100	2.78e-02	9.48e-03	2.06e-01	2.44e-01	14.32	3.53

Table 3.4: Partition of unity method, 2D affine model problem: on-line computational times.

contributor to t_{tot} is $t_{\Delta_N}(t_{\Delta_{NS}})$, the time required to obtain the error bound.

We then provide the comparison for the error bound operation counts for both the classical reduced basis method and the partition of unity method, since this operation count represents the major numerical challenge during the on-line stage. For our 2D model problem the operation count for the partition of unity method is higher than the operation count for the classical reduced basis method as depicted in Figure 3-15; the on-line computational times reflect this fact.

The off-line stage run for the PUM takes significantly less time – about 40 times less in our case – than that for the CRBM. The off-line stage is a one time computational effort, however it is in our interest to make this computational effort as small as possible; from this point of view the PUM clearly wins over the CRBM. As we mentioned earlier, $Q_i \leq Q, i = 1, \dots, S$. To emphasize the efficiency of the partition of unity method we note that even in the most pessimistic case when $Q_i = Q, i = 1, \dots, S$ the off-line effort for the PUM will still be significantly smaller than that of the CRBM due to the domain decomposition nature of the PUM.

One way to improve the on-line performance of the PUM is to provide a more efficient partition

CRBM			"truth"	PUM		
$\ e_N\ _Y$	t_{tot}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tr}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tot}, s	$\ e_N\ _Y$
1.69e-01	2.47e-03	1415.53	3.53	321.62	1.09e-02	1.66e-01
8.07e-02	7.92e-03	440.77	3.53	225.28	1.55e-02	8.43e-02
5.12e-02	5.27e-03	662.73	3.53	178.53	1.95e-02	5.18e-02
2.44e-02	7.33e-03	476.09	3.53	146.66	2.38e-02	2.37e-02
1.25e-02	1.62e-02	215.00	3.53	87.84	3.97e-02	1.23e-02
8.23e-03	2.11e-02	165.19	3.53	70.51	4.95e-02	8.16e-03
4.51e-03	2.82e-02	123.60	3.53	55.67	6.27e-02	4.47e-03
2.40e-03	4.12e-02	84.75	3.53	40.28	8.66e-02	2.36e-03
1.36e-03	6.29e-02	55.47	3.53	29.45	1.18e-01	1.38e-03
8.65e-04	8.04e-02	43.39	3.53	23.80	1.47e-01	8.52e-04
6.54e-04	9.13e-02	38.23	3.53	21.67	1.61e-01	6.52e-04
4.70e-04	1.10e-01	31.69	3.53	18.33	1.90e-01	4.75e-04
3.46e-04	1.33e-01	26.25	3.53	15.87	2.20e-01	3.45e-04
2.61e-04	1.43e-01	24.33	3.53	14.32	2.44e-01	2.57e-04

Table 3.5: Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: on-line computational times as functions of $\|e_N\|_Y$.

of unity selection which could be done by using an optimization procedure that will take into account the parametric complexity and the geometry of the problem.

However, let us consider a situation when the error bound operation counts for the two methods trade places. Let us revisit the 1D problem studied in Section 3.2.3. For this problem $Q = S - 2$, $Q_i \leq 2, S_i \leq 3$. Thus, the operation count for the error bound (PUM) is going to be less than $\sum_{i=1}^S 4(N+3)^2 = 4S(N+3)^2$. At the same time – if we address the same problem with the CRBM – we would have $(S-2)^2 N^2$ for the operation count of the error bound. It is easy to see that in this case the PUM will perform better than the CRBM time-wise especially for larger values of S, N .

In general, based on the expressions for the operation counts for the PUM and the CRBM we would expect the PUM to be more efficient for problems that allow partition of unity which satisfies the following condition:

$$\sum_{i=1}^S Q_i^2 \leq Q^2. \quad (3.145)$$

Clearly, the 1D problem we just mentioned satisfies (3.145). Furthermore, we would expect the PUM to perform better with respect to the on-line computational times for problems where the parameters are "domain-decomposition" based, i.e. the problems for which we can limit the participating operators for each partition to a minimum. Clearly, this method would win in the cases

CRBM			"truth"	PUM		
Δ_N	t_{tot}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tr}, s	$\frac{t_{tr}}{t_{tot}}$	t_{tot}, s	Δ_{NS}
7.58e+00	3.13e-03	1116.09	3.53	309.78	1.13e-02	7.62e+00
5.15e+00	2.79e-03	1249.78	3.53	237.08	1.47e-02	5.00e+00
1.05e+00	9.32e-03	374.35	3.53	128.99	2.71e-02	1.07e+00
8.55e-01	8.51e-03	410.27	3.53	101.59	3.44e-02	8.55e-01
5.95e-01	1.44e-02	242.74	3.53	75.59	4.62e-02	5.90e-01
3.83e-01	1.93e-02	180.95	3.53	61.48	5.68e-02	3.80e-01
1.91e-01	3.58e-02	97.61	3.53	43.51	8.02e-02	1.93e-01
8.52e-02	4.54e-02	76.79	3.53	34.84	1.00e-01	8.57e-02
6.74e-02	5.77e-02	60.44	3.53	27.86	1.25e-01	6.64e-02
4.04e-02	8.13e-02	42.92	3.53	22.56	1.55e-01	4.00e-02
3.09e-02	9.50e-02	36.72	3.53	19.21	1.82e-01	3.14e-02
2.14e-02	1.24e-01	28.04	3.53	16.25	2.15e-01	2.18e-02
1.59e-02	1.43e-01	24.33	3.53	14.32	2.44e-01	1.58e-02

Table 3.6: Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: on-line computational times as functions of the error bound.

where each partition would have just one associated operator (as in the 1D problem) with or – even better without – parametric dependence. At the same time, if the parametric dependencies are deeply interwoven and cannot be localized the CRBM would clearly have an advantage with respect to the on-line stage.

Turning back to the 2D model problem that was introduced in Section 2.4, we note that if we modify the problem by eliminating the convection term the use of the PUM for this modified would be advantageous compared the use of the CRBM with respect to the effectivity because it will allow us to use the scalable bound conditioners as we had in the 1D case.

Conclusions

To conclude the comparison of the CRBM and PUM in application the affine problems we can state that both methods address the affine coercive problems efficiently, both perform almost equally well in terms of convergence and effectivity. In general, the PUM requires significantly less time to complete the off-line stage. However, there is no clear winner as regards the on-line computational time because for each method there are affine problems for which this method performs better than the other method. For the cases when the PUM has a clear advantage over the CRBM for the off-line stage and at the same time both methods provide computational savings of approximately same magnitude during the on-line stage, the PUM could be used as a viable alternative to the

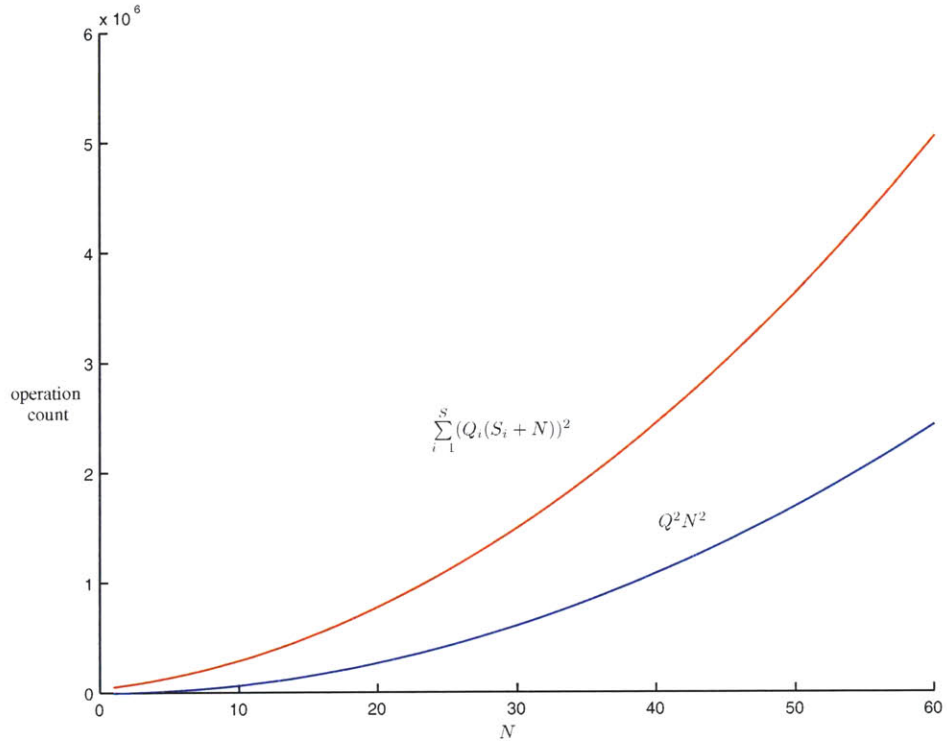


Figure 3-15: Comparison of classical reduced basis (left) and partition of unity (right) methods, 2D affine model problem: the dominating operation count for construction of error bound.

CRBM.

Chapter 4

Locally Non-affine Problems: Minimax Coefficient Approximation Method

In this Chapter we are going to present the minimax coefficient method which provides an alternative way to treat locally non-affine problems described in Section 3.1. The minimax coefficient approximation method (MCAM) relies on fundamentally different principles than the partition of unity method (PUM): in the MCAM we replace the underlying functional non-affine parametric dependence with an affine-like approximation. This idea was originally proposed in [5]. Such a replacement allows us to subsequently address the modified locally non-affine problem within the affine context. We now proceed to the mathematical description of the minimax coefficient approximation method in application to the locally non-affine problems.

4.1 Minimax Coefficient Approximation Method

4.1.1 Parametric Functional Dependency

As it was described in Section 3.1, the general abstract formulation for a problem with locally non-affine parametric dependence is given as: find $u(\mu)$ such that

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in Y, \tag{4.1}$$

where $a(\cdot, \cdot; \mu)$ admits the following decomposition

$$a(w, v; \mu) = a^A(w, v; \mu) + a^{NA}(w, v; \mu), \quad \forall w, v \in Y, \quad \forall \mu \in \mathcal{D}, \quad (4.2)$$

where, in turn, $a^A(\cdot, \cdot; \cdot)$ is an affine bilinear form:

$$a^A(w, v; \mu) = \sum_{q=1}^Q \Theta_q(\mu) a^q(w, v) \quad \forall w, v \in Y, \quad \forall \mu \in \mathcal{D}, \quad (4.3)$$

where the $\Theta^q(\mu): \mathcal{D} \rightarrow \mathbb{R}$ is a function and $a^q(\cdot, \cdot): Y^2 \rightarrow \mathbb{R}$ is a bilinear form without any parametric dependence; and $a^{NA}(\cdot, \cdot; \cdot)$ is a locally non-affine bilinear form.

The equation (4.1) is equivalent to the equation (3.6). It is important to mention that the assumption of the locally non-affine nature of the parametric dependence – as we are going to show later in this Chapter – is crucial for the construction of the *rigorous* error bound.

The parametric dependence in μ in (4.2) is indeed a functional dependence. For the model problem studied in Section 3.3 the non-affine component is introduced into the abstract formulation (4.1) through (3.107). Evaluation of (3.107) requires numerical integration of the function $k_5(\underline{x}; \mu)$ (defined in (3.100)) which we perform using Gaussian quadratures. Clearly, in this case the non-affine functional dependence is given by $k_5(\underline{x}; \mu)$.

We then rewrite (4.1) as

$$a^A(u(\mu), v; \mu) + a^{NA}(u(\mu), v; g(\underline{x}; \mu)) = f(v), \quad \forall v \in Y, \quad (4.4)$$

$$s(\mu) = l(u(\mu)), \quad (4.5)$$

where $g(\underline{x}; \mu)$ represents the functional form of the non-affine dependence and $s(\mu)$ is our output of interest. As before, $f(\cdot), l(\cdot) \in Y'$ are continuous bounded linear functionals. We now impose certain regularity restrictions on the bilinear forms $a(\cdot, \cdot; \cdot)$ and $a^{NA}(\cdot, \cdot; \cdot)$ from (4.1) and (4.4), that is, we assume that $a(\cdot, \cdot; \cdot)$ is bilinear, symmetric, coercive: there exists $\alpha(\mu) > 0$ such that

$$0 < \alpha_0 \leq \alpha(\mu) = \inf_{v \in Y} \frac{a(v, v; \mu)}{\|v\|_Y^2}, \quad \forall \mu \in \mathcal{D}, \quad (4.6)$$

and continuous: there exists $\gamma(\mu) > 0$ such that

$$a(w, v; \mu) \leq \gamma(\mu) \|w\|_Y \|v\|_Y \leq \gamma_0 \|w\|_Y \|v\|_Y, \quad \forall \mu \in \mathcal{D}, \quad (4.7)$$

whereas $a^{NA}(\cdot, \cdot, \cdot)$ is trilinear, affine in all of its arguments, symmetric in its first two arguments and continuous: there exists $\nu > 0$ such that

$$|a^{NA}(w, v; g)| \leq a^{NA}(w, w; g)^{\frac{1}{2}} a^{NA}(v, v; g)^{\frac{1}{2}} \leq \nu \|w\|_Y \|v\|_Y \|g\|_{L^\infty(\Omega_{na})}. \quad (4.8)$$

For $a^{NA}(\cdot, \cdot, \cdot)$ we allow general (unrestricted, non-affine) parametric dependence. The only limitation we impose is that this kind of general dependence can only appear in the area $\Omega_{na} \subset \Omega$ and the size of the former is small compared to the size of the latter. More specifically, $\forall v, w \in Y : v|_{\Omega_{na}} = 0, w|_{\Omega_{na}} = 0$

$$a^{NA}(u(\mu), v; g(\underline{x}; \mu)) \equiv 0, \quad \forall \mu \in \mathcal{D}. \quad (4.9)$$

We solve the problem given by (4.1) using the finite element method on a fine mesh with \mathcal{N} nodes which ensures that the analytical solution $u(\mu)$ and the finite element solution $u_{\mathcal{N}}(\mu)$ are practically indistinguishable from each other. The integration of the non-affine functional dependence introduced by the function $g(\underline{x}; \mu)$ is performed using the Gaussian quadratures for the finite elements of the truth mesh which ensures the exact integration for all polynomials up to a sufficiently high order, the details can be found in [7].

In practice the constant ν (4.8) is determined (often just by inspection) based on the properties of the form $a^{NA}(w, w; z)$ – which allows us to "extract" the functional dependence in z and put it into the $\|g\|_{L^\infty(\Omega_{na})}$ term – and the subsequent eigenproblem analysis. We are going to describe the procedure to obtain ν in the particular case of the model problem in Section 4.2.

4.1.2 S_M^g Sample Selection

We next consider the functional form of the parametric dependence as given in (4.4). In Chapter 3 we presented the partition of unity method which enables us to address the problems with locally non-affine parametric dependence. We now describe the underlying idea [5] for the minimax coefficient approximation method which is based on replacing $g(\underline{x}; \mu)$ with the affine-like function $g_M(\underline{x}; \mu)$. We here provide the mathematical algorithm for such a replacement.

We begin from building a recursive sequence $\mu_M^g, M = 1, \dots$. We start from some random μ_1^g

and set $\xi_1(\underline{x}) = g(\underline{x}; \mu)$. We then generate μ_2^g, μ_3^g, \dots , using the following recursive algorithm

$$\mu_M^g = \arg \max_{\mu \in S_{\mathcal{D}^{NA}}} \epsilon_{M-1}^*(\mu), \quad (4.10)$$

$$\xi_M(\underline{x}) = g(\underline{x}; \mu_M^g), \quad (4.11)$$

where¹ $S_{\mathcal{D}^{NA}}$ is a fine enough sample (mesh) in $\mathcal{D}^{NA} \subset \mathbb{R}^{P_{NA}}$, the area of variation of P_{NA} parameters that define the non-affine functional dependence; furthermore

$$\epsilon_M^*(\mu) = \min_{\gamma \in \mathbb{R}^M} \max_{x \in \Omega_{na}} |g(\underline{x}; \mu) - \sum_{j=1}^M \gamma_j \xi_j(\underline{x})|, \quad (4.12)$$

$$\underline{\gamma}^* = \arg \min_{\gamma \in \mathbb{R}^M} \max_{x \in \Omega_{na}} |g(\underline{x}; \mu) - \sum_{j=1}^M \gamma_j \xi_j(\underline{x})|, \quad (4.13)$$

$$g_M^*(\underline{x}; \mu) = \sum_{j=1}^M \gamma_j^* \xi_j(\underline{x}). \quad (4.14)$$

It is important to note that this situation is different from the case described in Section 2.3.4 where we discussed the selection of the optimal sample S_N . As it was mentioned in Section 2.3.4, when P , the number of the parameters that define the problem, is even moderately big, say $O(10)$, it becomes too computationally expensive to perform a search over a mesh over \mathcal{D} . However, when solving (4.10) we only have to optimize over the set of P_{NA} parameters which define the locally non-affine dependence. Since $P_{NA} \leq P$, we have better chances to be able to apply (4.10) in our sample selection algorithm when we actually choose candidates for μ_M^g from a fine enough mesh in \mathcal{D}^{NA} .

It is possible to have a situation where P_{NA} is too big for us to generate a fine mesh in $\mathcal{D}^{NA} \subset \mathbb{R}^{P_{NA}}$. If such a situation occurs, we would have to utilize the approach similar to the one described in Section 2.3.4 and replace $S_{\mathcal{D}^{NA}}$ in (4.10) with $S_{M_{pool}}$, an appropriately large pool of randomly chosen points:

$$\mu_M^g = \arg \max_{\mu \in S_{M_{pool}}} \epsilon_M^*(\mu), \quad (4.15)$$

the cardinality of $S_{M_{pool}}$ is equal to M_{pool} .

In the model problem introduced in Section 3.3 equals to 1 which allows us to build a fine mesh over the area of variation of the only parameter τ . The recursive nature of equations (4.10) and

¹In practice we replace (4.12) with an appropriate linear programming program defined over a fine mesh in Ω_{na} .

(4.12) guarantees that $\epsilon_{M-1}^*(\mu_M^g)$ is a decreasing non-negative sequence. We repeat steps (4.10) and (4.12) until $\epsilon_{M-1}^*(\mu_M^g)$ gets sufficiently small and set M_{max} equal to the current M . We next define the sample $S_{M_{max}}^g = \{\mu_1^g, \dots, \mu_{M_{max}}^g\}$ and introduce the functional space $V_{M_{max}}^g = \text{span}\{\xi_M(\underline{x}) = g(\underline{x}, \mu_M^g), M = 1, \dots, M_{max}\}$.

In order to demonstrate the validity of the approach we formulate and prove several statements in a similar way how it was done in [5]. We first assume that

$$\epsilon_{M_{max}}^*(\mu_{M_{max}}^g) \geq \epsilon_0 > 0 \quad (4.16)$$

for some given value of ϵ_0 . Using the fact that $V_1^g \subset V_2^g \subset \dots \subset V_{M_{max}}^g$ from (4.12) we see that $\epsilon_1^*(\mu_2^g) \geq \epsilon_2^*(\mu_3^g) \geq \dots \geq \epsilon_{M_{max}-1}^*(\mu_{M_{max}}^g) \geq \epsilon_0$. Let us now state and prove the Lemma 4.1

Lemma 4.1. The dimension of space V_M^g is equal to M , $M = 1, \dots, M_{max}$.

We proceed by induction. The basis of induction is obvious: $\dim(V_1^g) = 1$. Let us assume that $\dim(V_{M-1}^g) = M - 1$, if then $\dim(V_M^g) = M - 1$ that implies that $g(\underline{x}; \mu_M^g) \in W_{M-1}^g$ which contradicts $\epsilon_{M-1}^*(\mu_M^g) \geq 0$.

4.1.3 "Magic Points" Selection

Once we are done with the selection of the sample $S_{M_{max}}^g$ we run another recursive procedure. We now introduce a sequence of "residual" functions $r_M(\underline{x})$, $M = 1, \dots, M_{max}$ and a sequence of physical points \underline{t}_M , $M = 1, \dots, M_{max}$ that are constructed according to the following procedure

$$r_1(\underline{x}) = \xi_1(\underline{x}), \quad (4.17)$$

$$\underline{t}_1 = \arg \max_{\underline{x} \in \Omega_{na}} |r_1(\underline{x})|, \quad (4.18)$$

$$\underline{B}_{11}^1 = \xi_1(\underline{t}_1); \quad (4.19)$$

then, for $M = 2, \dots, M_{max}$

$$\underline{B}^{M-1} \sigma^{M-1} = [\xi_M(\underline{t}_1), \dots, \xi_M(\underline{t}_{M-1})]^T, \quad (4.20)$$

$$r_M(\underline{x}) = \xi_M(\underline{x}) - \sum_{j=1}^{M-1} \sigma_j^{M-1} \xi_j(\underline{x}), \quad (4.21)$$

$$\underline{t}_M = \arg \max_{\underline{x} \in \Omega_{na}} |r_M(\underline{x})|, \quad (4.22)$$

$$\underline{B}_{ij}^M = \xi_j(\underline{t}_i), 1 \leq i, j \leq M. \quad (4.23)$$

We refer to the sequence $\underline{t}_M, M = 1, \dots, M_{max}$ as to the "magic points". We now prove an important lemma that guarantees the existence and uniqueness of \underline{t}_M .

Lemma 4.2. The rank of \underline{B}^M is equal to $M, M = 1, \dots, M_{max}$.

As in Lemma 4.1, we proceed by induction. $\underline{B}^1 = \max_{\Omega_{na}} |\xi_1(\underline{x})| \geq 0$ based on Lemma 4.1. Let us now suppose that the $\text{rank}(\underline{B}^{M-1}) = M - 1$. If we now suppose that $\text{rank}(\underline{B}^M) = M - 1$ and look at (4.22) and $\xi_M(\underline{t}_M)$, we readily see that $r_M(\underline{t}_M) = 0$. What follows is that $\xi_M(\underline{x})$ can be expressed as a linear combination of $\xi_1(\underline{x}), \dots, \xi_{M-1}(\underline{x}) \forall \underline{x} \in \Omega_{na}$ which clearly contradicts Lemma 4.1. By this contradiction we prove that $\text{rank}(\underline{B}^M) = M$ and we indeed have a unique sequence of "magic points".

We are now ready to introduce the affine-like the replacement function $g_M(\underline{x}; \mu)$ as

$$g_M(\underline{x}; \mu) = \sum_{m=1}^M \alpha_m(\mu) \xi_m(\underline{x}), \quad \underline{\alpha}(\mu) \in \mathbb{R}^M, M \leq M_{max}, \quad (4.24)$$

$$\underline{\alpha}(\mu) = (\underline{B}^M)^{-1} [g(\underline{t}_1, \mu) \dots g(\underline{t}_M, \mu)]^T. \quad (4.25)$$

From (4.25) we can see that

$$g_M(\underline{t}_i; \mu) = g(\underline{t}_i; \mu), \quad i = 1, \dots, M. \quad (4.26)$$

We then introduce the interpolation error $\epsilon_M(\mu)$ as

$$\epsilon_M(\mu) = \|g(\underline{x}; \mu) - g_M(\underline{x}; \mu)\|_{L^\infty(\Omega_{na})}. \quad (4.27)$$

We also define the "Lebesgue" constant Λ_M as

$$\Lambda_M = \sup_{\underline{x} \in \Omega_{na}} \sum_{k=1}^M |V_k^M(\underline{x})|, \quad (4.28)$$

where

$$\sum_{j=1}^M \underline{B}_{ji}^M V_j^M(\underline{x}) = \xi_i(\underline{x}), \quad i = 1, \dots, M. \quad (4.29)$$

We are now ready to prove the next Lemma.

Lemma 4.3. $\epsilon_M(\mu) \leq \epsilon_M^*(\mu)(1 + \Lambda_M), \forall \mu \in \mathcal{D}, M \leq M_{max}$.

We first define $e^*(\underline{x}; \mu) = g_M^*(\underline{x}; \mu) - g_M(\underline{x}; \mu)$. We now write

$$\epsilon_M(\mu) = \|g(\underline{x}; \mu) - g_M^*(\underline{x}; \mu) + g_M^*(\underline{x}; \mu) - g_M(\underline{x}; \mu)\|_{L^\infty(\Omega_{na})} \leq \epsilon_M^*(\mu) + \|e^*(\underline{x}; \mu)\|_{L^\infty(\Omega_{na})}.$$

Since $e^*(\underline{x}; \mu) \in V_M^g$ we can write that

$$\begin{aligned} \epsilon_M(\mu) &\leq \epsilon_M^*(\mu) + \left\| \sum_{m=1}^M \delta_m^M(\mu) \xi_m(\underline{x}) \right\|_{L^\infty(\Omega_{na})} = \epsilon_M^*(\mu) + \left\| \sum_{k=1}^M \sum_{m=1}^M \underline{B}_{km}^M \delta_m^M(\mu) V_k^M(\underline{x}) \right\|_{L^\infty(\Omega_{na})} = \\ &\epsilon_M^*(\mu) + \left\| \sum_{i=1}^M e_M^*(\underline{t}_i; \mu) V_i^M(\underline{x}) \right\|_{L^\infty(\Omega_{na})} \leq \epsilon_M^*(\mu)(1 + \Lambda_M), \end{aligned} \quad (4.30)$$

since $\|e_M^*(\underline{x}; \mu)\|_{L^\infty(\Omega_{na})} \leq \epsilon_M^*(\mu)$.

In [5] a proof is presented where an extremely pessimistic (exponential in M) upper bound for Λ_M is provided. Lemma 4.3 ensures – provided that Λ_M is sufficiently small and M is sufficiently big – that $g_M(\cdot; \cdot)$ and $g(\cdot; \cdot)$ are indeed close.

We talk about measuring the error between the two functions in Section 4.2.2 and provide concrete values for Λ_M for the case of a 2D model problem.

We finally formulate the following Corollary.

Corollary 4.1. $\epsilon_{M-1}^*(\mu_M^g) \leq |r_M(\underline{t}_M)| \leq (1 + \Lambda_M) \epsilon_{M-1}^*(\mu_M^g)$.

The proof of the corollary follows directly from (4.12), (4.21), and Lemma 4.3.

Later in this Chapter we will often omit the dependence in space in $g_M(\underline{x}, \mu)$ and $g(\underline{x}, \mu)$ and write $g_M(\mu)$ and $g(\mu)$ instead, respectively.

4.1.4 Reduced Basis Approximation

As in the partition of unity method, we introduce the sample set $S_N = \{\mu_1, \dots, \mu_N\}$ using one of the sample selection algorithms described in Section 2.3.5 and subsequently construct the reduced basis function space as $W_N = \text{span}\{\zeta_i = u(\mu_i), 1 \leq i \leq N\}$. We note that this idea which was first introduced in Section 2.3.1 remains the key component of the minimax coefficient approximation reduced basis method. We now look for the approximation $u_{NM}(\mu) \in W_N$ to $u(\mu)$ from (4.1) as

the solution of

$$a^A(u_{NM}(\mu), v; \mu) + a^{NA}(u_{NM}(\mu), v; g_M(\mu)) = f(v), \forall v \in W^N, \quad (4.31)$$

$$g_M(\mu) = \sum_{m=1}^M \alpha_m(\mu) \xi_m(\underline{x}), \quad \underline{\alpha}(\mu) \in \mathbb{R}^M, \quad (4.32)$$

$$\underline{\alpha}(\mu) = (\underline{B}^M)^{-1} [g(\underline{t}_1; \mu) \dots g(\underline{t}_M; \mu)]^T, \quad (4.33)$$

$$s_{NM}(\mu) = l(u_{NM}(\mu)). \quad (4.34)$$

Invoking (4.32) and the properties of $a^{NA}(\cdot, \cdot; \cdot)$ we expand (4.31) as

$$\sum_{q=1}^Q \Theta_q(\mu) a^q(u_{NM}(\mu), v) + \sum_{k=1}^M \alpha_k(\mu) a^{NA}(u_{NM}(\mu), v; \xi_k) = f(v), \quad \forall v \in Y. \quad (4.35)$$

We notice that the formulation (4.35) essentially has an affine structure comparable to (2.11) since $a^{NA}(w, v; \xi_k)$, $k = 1, \dots, M$ does not depend on μ . Since $u_{NM}(\mu) \in W_N$ it can be decomposed as

$$u_{NM}(\mu) = \sum_{i=1}^N u_{NM_i}(\mu) \zeta_i. \quad (4.36)$$

4.1.5 A Posteriori Error Estimation

We start by introducing a bound conditioner [18, 39, 52, 57] – a symmetric, continuous, and coercive bilinear form $\hat{a}(v, v; \mu)$ such that $\forall \mu \in \mathcal{D}$

$$\alpha(\mu) (\|v\|_Y)^2 \leq \rho_{min}(\mu) \hat{a}(v, v; \mu) \leq a(v, v; \mu) \leq \rho_{max}(\mu) \hat{a}(v, v; \mu) \leq \gamma(\mu) (\|v\|_Y)^2, \quad \forall v \in Y, \quad (4.37)$$

where

$$1 \leq \rho_{min}(\mu), \quad \rho_{max}(\mu) \leq \rho, \quad \alpha_0 \leq \alpha(\mu), \quad \gamma(\mu) \leq \gamma_0, \quad (4.38)$$

for some (preferably small) constant $\rho \in \mathbb{R}$. The more detailed of description of different types of bound conditioners is contained in [52, 57].

We now introduce the error function $e_{NM}(\mu)$ (compare to (2.20), 3.32, 3.33, and (3.128)) as

$$e_{NM}(\mu) = u(\mu) - u_{NM}(\mu). \quad (4.39)$$

Using (4.1) and (4.31) we then notice that $e_{NM}(\mu)$ satisfies the following "error equation" (compare

to (2.21))

$$\underbrace{a(u(\mu) - u_{NM}(\mu), v; \mu)}_{e_{NM}(\mu)} = f(v) - a^A(u_{NM}(\mu), v; \mu) - a^{NA}(u_{NM}(\mu), v; g_M(\mu)) - a^{NA}(u_{NM}(\mu), v; g(\mu) - g_M(\mu)), \quad \forall v \in Y. \quad (4.40)$$

We next introduce linear functionals $R_1(v; \mu)$ and $R_2(v; \mu)$ such that

$$R_1(v; \mu) = f(v) - a^A(u_{NM}(\mu), v; \mu) - a^{NA}(u_{NM}(\mu), v; g_M(\mu)), \quad (4.41)$$

$$R_2(v; \mu) = -a^{NA}(u_{NM}(\mu), v; g(\mu) - g_M(\mu)). \quad (4.42)$$

and rewrite (4.40) as

$$a(e_{NM}(\mu), v; \mu) = R_1(v; \mu) + R_2(v; \mu). \quad (4.43)$$

We next consider two subproblems where we look for $\hat{e}_{NM,1}(\mu)$ and $\hat{e}_{NM,2}(\mu)$ such that

$$\hat{a}(\hat{e}_{NM,1}(\mu), v; \mu) = R_1(v; \mu), \quad (4.44)$$

$$\hat{a}(\hat{e}_{NM,2}(\mu), v; \mu) = R_2(v; \mu). \quad (4.45)$$

We next introduce $\tilde{\Delta}_{NM,1}(\mu)$ as

$$\tilde{\Delta}_{NM,1}(\mu) = \hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu). \quad (4.46)$$

We then invoke (4.8), (4.37), and (4.27) to obtain

$$\hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu) = R_2(\hat{e}_{NM,2}(\mu); \mu) \leq \underbrace{\nu \|g(\mu) - g_M(\mu)\|_{L^\infty(\Omega_{na})}}_{\epsilon_M(\mu)} \|u_{NM}(\mu)\|_Y \|\hat{e}_{NM,2}(\mu)\|_Y \leq$$

$$\frac{\nu \epsilon_M(\mu)}{\alpha} \hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu)^{\frac{1}{2}} \hat{a}(\hat{e}_{NM,2}, \hat{e}_{NM,2})^{\frac{1}{2}}. \quad (4.47)$$

From (4.47) we readily construct the bound $\tilde{\Delta}_{NM,2}(\mu)$ for $\hat{a}(\hat{e}_{NM,2}, \hat{e}_{NM,2})$

$$\hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu) \leq \Delta_{NM,2}(\mu) = \left(\frac{\nu \epsilon_M(\mu)}{\alpha(\mu)} \right)^2 \hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu), \quad (4.48)$$

Plugging $v = e_{NM}(\mu)$ into (4.43) we obtain

$$\begin{aligned}
a(e_{NM}(\mu), e_{NM}(\mu); \mu) &= R_1(e_{NM}(\mu); \mu) + R_2(e_{NM}(\mu); \mu) = \\
&\hat{a}(\hat{e}_{NM,1}(\mu), e_{NM}(\mu); \mu) + \hat{a}(\hat{e}_{NM,2}(\mu), e_{NM}(\mu); \mu) \leq \\
&\left(\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)^{\frac{1}{2}} + \hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu)^{\frac{1}{2}} \right) \hat{a}(e_{NM}(\mu), e_{NM}(\mu); \mu)^{\frac{1}{2}} \leq \\
&((\tilde{\Delta}_{NM,1})^{\frac{1}{2}} + (\tilde{\Delta}_{NM,2})^{\frac{1}{2}}) \hat{a}(e_{NM}(\mu), e_{NM}(\mu); \mu)^{\frac{1}{2}}. \tag{4.49}
\end{aligned}$$

We now explain how we arrived at the final result of (4.49): in the first step we invoked (4.43), in the second – (4.44) and (4.45), in the the third – Cauchy-Schwartz' inequality, and, finally, for the last step – the definitions of $\tilde{\Delta}_{NM,1}$ and $\tilde{\Delta}_{NM,2}$, (4.46) and (4.47), respectively.

From (4.49) invoking (4.37) we readily construct the error bound $\tilde{\Delta}_{NM}(\mu)$ such that

$$\begin{aligned}
a(e_{NM}(\mu), e_{NM}(\mu); \mu) &\leq \tilde{\Delta}_{NM}(\mu) = \left((\tilde{\Delta}_{NM,1})^{\frac{1}{2}} + (\tilde{\Delta}_{NM,2})^{\frac{1}{2}} \right)^2 = \\
&\left(\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)^{\frac{1}{2}} + \frac{\nu \epsilon_M(\mu)}{\alpha(\mu)} \hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu)^{\frac{1}{2}} \right)^2. \tag{4.50}
\end{aligned}$$

Finally invoking (4.37) we construct the error bound $\Delta_{NM}(\mu)$ for $\|e_{NM}(\mu)\|_Y$

$$\begin{aligned}
\|e_{NM}(\mu)\|_Y \leq \Delta_{NM}(\mu) &= \sqrt{\frac{\tilde{\Delta}_{NM}(\mu)}{\alpha(\mu)}} = \sqrt{\frac{\tilde{\Delta}_{NM,1}(\mu)}{\alpha(\mu)}} + \sqrt{\frac{\tilde{\Delta}_{NM,2}(\mu)}{\alpha(\mu)}} = \\
&\sqrt{\frac{\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)}{\alpha(\mu)}} + \frac{\nu \epsilon_M(\mu)}{\alpha(\mu)} \sqrt{\frac{\hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu)}{\alpha(\mu)}}. \tag{4.51}
\end{aligned}$$

As before, we introduce the effectivities $\tilde{\eta}_{NM}(\mu)$ and $\eta_{NM}(\mu)$ which serve as a measure of the sharpness of the error bounds $\Delta_{NM}(\mu)$, $\tilde{\Delta}_{NM}(\mu)$

$$1 \leq \tilde{\eta}_{NM}(\mu) = \frac{\tilde{\Delta}_{NM}(\mu)}{a(e_{NM}(\mu), e_{NM}(\mu); \mu)}, \tag{4.52}$$

$$1 \leq \eta_{NM}(\mu) = \frac{\Delta_{NM}(\mu)}{\|e_{NM}(\mu)\|_Y}. \tag{4.53}$$

As the equation (4.51) indicates, we have two terms that contribute to the error bound $\Delta_{NM}(\mu)$.

Let us denote these terms as

$$\Delta_{NM}(\mu) = \Delta_{NM,1}(\mu) + \Delta_{NM,2}(\mu), \quad (4.54)$$

$$\Delta_{NM,1}(\mu) = \sqrt{\frac{\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)}{\alpha(\mu)}}, \quad (4.55)$$

$$\Delta_{NM,2}(\mu) = \frac{\nu \epsilon_M(\mu)}{\alpha(\mu)} \sqrt{\frac{\hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu)}{\alpha(\mu)}}. \quad (4.56)$$

We are going to discuss the importance and relative contribution of each term later in this Chapter in Section 4.2 when we consider a particular model problem.

Similar to how it was demonstrated in [5] we now make an assumption that

$$g(\underline{x}; \mu) \in V_{M+1}^g; \quad (4.57)$$

under this assumption it is then an easy matter to show from (4.26) that

$$\epsilon_M(\mu) = |g(\underline{t}_{M+1}; \mu) - g_M(\underline{t}_{M+1}; \mu)|. \quad (4.58)$$

It is important to mention that at the moment we cannot rigorously verify the validity of this assumption for an arbitrary non-affine functional dependence. However, we did the numerical tests of (4.58) we found that it approximately holds especially for larger values of M .

Alternatively, if we assume that the sample we choose candidates for μ_M^g in (4.10) is arbitrarily fine we then can bound $\Delta_{NM,2}(\mu)$ based on Lemma 4.3 as follows

$$\Delta_{NM,2} \leq \frac{\nu(1 + \Lambda_M) \epsilon_M^*(\mu_M^g)}{\alpha(\mu)} \sqrt{\frac{\hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu)}{\alpha(\mu)}}. \quad (4.59)$$

There are two important issues regarding this last assumption: first, as we mentioned earlier in this Chapter, in many cases it is computationally impossible to come up with a reasonably fine mesh over D^{NA} ; second, it is again impossible to provide an arbitrarily fine sample (mesh).

Thus, both of the assumptions we just presented can only be used on a heuristic level and are not valid for completely rigorous error estimation. At the same time, both of these two assumptions – as we will show in Section 4.1.6 – can potentially reduce the computational effort for the a posteriori error estimation albeit at a slight loss of rigor.

4.1.6 Off-line/Online Procedure

Reduced Basis Approximation

Invoking the decomposition (4.36) we rewrite the equation (4.31) in the matrix-vector form as

$$\underline{A}_N(\mu)\underline{u}_{NM}(\mu) = \underline{F}_N, \quad (4.60)$$

where $\underline{u}_{NM}(\mu)$ is the unknown vector of size N corresponding to the reduced basis approximation coefficients $u_{NM_1}(\mu), \dots, u_{NM_N}(\mu)$; and the $N \times N$ matrix $\underline{A}_N(\mu)$ admits the following decomposition which is consistent with (4.35)

$$\underline{A}_N(\mu) = \sum_{q=1}^Q \Theta_q(\mu) \underline{A}_N^q + \sum_{i=m}^M \alpha_m(\mu) \tilde{\underline{A}}_N^m, \quad (4.61)$$

where

$$\underline{A}_{N_{jl}}^q = a^q(\zeta_j, \zeta_l), \underline{F}_{N_j} = f(\zeta_j), \quad q = 1, \dots, Q, \quad j, l = 1, \dots, N, \quad (4.62)$$

$$\tilde{\underline{A}}_{N_{jl}}^m = a^{NA}(\zeta_j, \zeta_l; \xi_m), \quad m = 1, \dots, M, \quad j, l = 1, \dots, N, \quad (4.63)$$

and $\underline{\alpha}(\mu)$ is given by (4.33).

Error Bound

We now rewrite (4.41) as

$$\begin{aligned} R_1(v; \mu) &= f(v) - \sum_{q=1}^Q \sum_{l=1}^N u_{NM_l}(\mu) \Theta_q(\mu) a^q(\zeta_l, v) - \sum_{l=1}^N u_{NM_l}(\mu) a^{NA}(\zeta_l, v; \xi_m) = \\ &= f(v) - \sum_{q=1}^Q \sum_{l=1}^N u_{NM_l}(\mu) \Theta_q(\mu) a^q(\zeta_l, v) - \sum_{m=1}^M \alpha_m(\mu) \sum_{l=1}^N u_{NM_l}(\mu) a^{NA}(\zeta_l, v; \xi_m). \end{aligned} \quad (4.64)$$

Renumbering (4.64) we obtain

$$R_1(v; \mu) = \sum_{j=1}^{1+N(Q+M)} \tau_j(\mu) \Upsilon_j(v), \quad (4.65)$$

where

$$\tau_1(\mu) = 1, \Upsilon_1(v) = f(v); \quad (4.66)$$

for $j = 2, \dots, 1 + QN$

$$j = 1 + (q-1)N + l, \tau_j(\mu) = -\Theta_q(\mu)u_{NM_l}(\mu), \Upsilon_j(v) = a^q(\zeta_l, v), \quad q = 1, \dots, Q, l = 1, \dots, N; \quad (4.67)$$

and for $j = 2 + QN, \dots, 1 + (Q + M)N$

$$j = 1 + QN + (m-1)N + l, \tau_j(\mu) = -\alpha_m(\mu)u_{NM_l}(\mu), \Upsilon_j(v) = a^{NA}(\zeta_l, v; \xi_m), \quad m = 1, \dots, M, l = 1, \dots, N. \quad (4.68)$$

Note that $\Upsilon_j(v) \in Y', j = 1, \dots, 1 + (Q + M)N$ is a parameter independent linear functional.

Invoking (4.65), (4.44), (4.66), (4.67), and (4.68) we can rewrite the expression for $\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)$ as

$$\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu) = \sum_{j=1}^{1+N(Q+M)} \sum_{j'=1}^{1+N(Q+M)} \tau_j(\mu)\tau_{j'}(\mu)\Lambda_{jj'}, \quad (4.69)$$

where

$$\Lambda_{jj'} = \hat{a}(z_j, z_{j'}; \mu), \quad j, j' = 1, \dots, 1 + N(Q + M), \quad (4.70)$$

$$\hat{a}(z_j, v; \mu) = \Upsilon_j(v), \quad \forall v \in Y. \quad (4.71)$$

Invoking the bilinearity property of $\hat{a}(\cdot, \cdot; \cdot)$, we next note that $\hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu)$ term in (4.51) can be expressed as

$$\hat{a}(u_{NM}(\mu), u_{NM}(\mu); \mu) = \sum_{l=1}^N \sum_{l'=1}^N u_{NM_l}(\mu)u_{NM_{l'}}(\mu)\hat{a}(\xi_l, \xi_{l'}; \mu)_Y. \quad (4.72)$$

We finally use (4.51), (4.69), and (4.72) to obtain the expression for $\Delta_{NM}(\mu)$ as

$$\Delta_{NM}(\mu) = \frac{1}{\sqrt{\alpha(\mu)}} \sqrt{\sum_{j=1}^{1+N(Q+M)} \sum_{j'=1}^{1+N(Q+M)} \tau_j(\mu)\tau_{j'}(\mu)\Lambda_{jj'} +}$$

$$\frac{\nu \epsilon_M(\mu)}{\alpha(\mu) \sqrt{\alpha(\mu)}} \sqrt{\sum_{l=1}^N \sum_{l'=1}^N u_{NM_l}(\mu) u_{NM_{l'}}(\mu) \hat{a}(\xi_l, \xi_{l'}; \mu)}. \quad (4.73)$$

In order to determine the interpolation error $\epsilon_M(\mu) = \|g(\mu) - g_M(\mu)\|_{L_\infty(\Omega_{na})}$ we have to construct and evaluate the function $g(\mu) - g_M(\mu)$ in n_{na} points and determine its maximum absolute value; clearly this procedure adds an $O(Mn_{na})$ dependence to the operation count. This is why the assumption of locally non-affine dependence is essential from the *rigorous* error estimation point of view. We can reasonably expect that the contribution to the error bound resulting from the second term of (4.73) decreases with M . In fact, if we admit a slight loss of rigor and use one of the assumptions we presented at the end of Section 4.1.5 we can get rid of this $O(Mn_{na})$ dependence in the operation count.

Turning to the decomposition of the error bound $\Delta_{NM}(\mu)$ we introduced in (4.54) we observe that

$$\Delta_{NM,1} = \frac{1}{\sqrt{\alpha(\mu)}} \sqrt{\sum_{i=1}^{1+N(Q+M)} \sum_{i'=1}^{1+N(Q+M)} \tau_i(\mu) \tau_{i'}(\mu) \Lambda_{ii'}}, \quad (4.74)$$

$$\Delta_{NM,2} = \frac{\nu \epsilon_M(\mu)}{\alpha(\mu) \sqrt{\alpha(\mu)}} \sqrt{\sum_{l=1}^N \sum_{l'=1}^N u_{NM_l}(\mu) u_{NM_{l'}}(\mu) \hat{a}(\xi_l, \xi_{l'}; \mu)}. \quad (4.75)$$

Numerical Algorithm

Off-line

1. Generate S_M^g as described in Section 4.1.2 and construct the basis for V_M^g ;
2. Generate \underline{B}_M^{-1} as shown in Section 4.1.3;
3. Generate S_N as described in Section 2.3.4 and construct the basis for W_N ;
4. Precompute $\underline{A}_N^q, \underline{F}_N$ as in (4.62), $\tilde{\underline{A}}_N^m$ as in (4.63), $q = 1, \dots, Q$, $m = 1, \dots, M$;
5. Precompute $\Lambda_{jj'}$ as in (4.70) and $\hat{a}(\xi_l, \xi_{l'}; \mu)$ as in (4.72) $j, j' = 1, \dots, 1+N(Q+M)$, $l, l' = 1, \dots, N$.

On-line

1. Calculate $\underline{\alpha}(\mu)$ from (4.33). Cost: $O(M^2)$;
 2. Form (4.62) and solve (4.60) for $\underline{u}_{NM}(\mu)$. Cost: $O((Q+M)N^2) + O(N^3)$;
 3. Construct the error bound $\Delta_{NM}(\mu)$ from (4.73). Cost: $O((Q+M)^2N^2) + O(Mn_{na})$.
- Total on-line cost: $O(M^2) + O((Q+M)N^2) + O(N^3) + O((Q+M)^2N^2) + O(Mn_{na})$.

Note that in the on-line operation count we have a $O((Q + M)^2 N^2)$ term which might present a significant (if not the main) contribution to this operation count based on the results of Chapter 2 where the contribution of $O(Q^2 N^2)$ was the most expensive part of the on-line operation count. Thus, the $O((Q + M)^2 N^2)$ term might impose certain restrictions on the affine complexity of the problem; we are going to return to this issue in the next Section.

4.2 Model Problem: 2D Heat Conduction Example

To illustrate the application of the minimax coefficient approximation method we consider a slightly simplified version of the model problem introduced in Section 3.3. We now fix the geometric parameters $\alpha_i, \beta_i, i = 1, \dots, 4$ at the values of $\alpha_{ref}, \beta_{ref}$, respectively. The thermal fin is then characterized by a 6-dimensional vector

$$\mu = \{k_1, k_2, k_3, k_4, \text{Bi}, \tau\},$$

and the non-affine parametric functional dependence is given by

$$k_5(\underline{x}; \mu) = \exp\left(\tau \sqrt{(x_1 - x_1^*)^2 + (x_2 - x_2^*)^2}\right), \quad (4.76)$$

where \underline{x}^* is left bottom corner point of Ω_{na} as shown in Figure 3-4. We choose the parametric domain \mathcal{D} as $\mathcal{D} = [0.1, 10.0]^4 \times [0.01, 1.0] \times [-1.1424, 1.1424]$. This problem is well-posed [46] for the same reasons as the problem in Section 3.3. We are only interested in the norm of the error $\|e_{NM}(\mu)\|_Y$ and the respective error bound for it since the error bounds for the linear outputs – as discussed in Chapter 2 – can be easily expressed through $\|e_{NM}(\mu)\|_Y$. Compared to the model problem studied in Section 3.3, this model problem has fewer affine parameters.

4.2.1 Weak Form

We next provide the formulation of this simplified model problem which is consistent with (4.1), (4.2), and (4.4). Based on (3.108) for the case of fixed geometry $a(w, v; \mu)$ admits the following decomposition $\forall w, v \in Y$

$$a(w, v; \mu) = \sum_{i=0}^4 k_i \int_{\Omega_i} \nabla w \nabla v + \text{Bi} \int_{\Gamma_{ext} \setminus \Gamma_{root}} wv + \int_{\Omega_5 = \Omega_{na}} k_5(\underline{x}, \mu) \nabla w \nabla v, \quad (4.77)$$

$$f(v) = \int_{\Gamma_{\text{root}}} v. \quad (4.78)$$

Thus, in this case the affine part $a^A(.,.,.)$ is given by

$$a^A(w, v; \mu) = \sum_{i=0}^4 k_i \int_{\Omega_i} \nabla w \nabla v + \text{Bi} \int_{\Gamma_{\text{ext}} \setminus \Gamma_{\text{root}}} wv; \quad (4.79)$$

whereas the non-affine part $a^{NA}(.,.,.)$ is given by

$$a^{NA}(w, v; g) = \int_{\Omega_5 = \Omega_{na}} g \nabla w \nabla v, \quad (4.80)$$

and hence the constant ν as in (4.8) is derived in the following way

$$\begin{aligned} |a^{NA}(w, v; z)| &\leq \|g\|_{L^\infty(\Omega_{na})} \int_{\Omega_{na}} \nabla w \nabla v \leq \|g\|_{L^\infty(\Omega_{na})} \left(\int_{\Omega_{na}} (\nabla w)^2 \right)^{\frac{1}{2}} \left(\int_{\Omega_{na}} (\nabla v)^2 \right)^{\frac{1}{2}} \leq \\ &\sup_{r \in Y} \frac{\int_{\Omega_{na}} (\nabla r)^2}{(\|r\|_Y)^2} \|g\|_{L^\infty(\Omega_{na})} \|w\|_Y \|v\|_Y. \end{aligned} \quad (4.81)$$

Here, for our model problem

$$\nu = \sup_{r \in Y} \frac{\int_{\Omega_{na}} (\nabla r)^2}{(\|r\|_Y)^2}, \quad (4.82)$$

and is obtained as the solution of a generalized eigenvalue problem. For our problem ν assumes the value of 0.999957. As we mentioned earlier, for the general case in the derivation of ν from (4.8) we would rely on some integral majorizing theorem (since in most cases the functional dependence in g would appear in an integral because of the weak formulation) and subsequent eigenvalue analysis.

4.2.2 Numerical Results

To obtain the “truth” solution we consider a finite element mesh that consists of 21802 points.

Approximation of Non-affine Functional Dependency

We now present the numerical results for the approximation of the conductivity function described by the equation (4.76). We first generate the sample S_M^g as it was discussed in Section 4.1.2. In

Figure 4-1 we present the plots of $\epsilon_M^*(\mu_{M+1}^g)$. The plot of $\epsilon_M^*(\mu_{M+1}^g)$ (in red), $|r_{M+1}(\underline{t}_{M+1})|$ (in blue), and ϵ_M (in green) as functions of M ; ϵ_M is averaged over $S_{test} = \{\mu_1^t, \dots, \mu_{N_{test}}^t\}$, a pool of $N_{test} = 10000$ randomly chosen points which provides some level of statistical significance for our test. Further in this Chapter all the numerical results – unless noted otherwise – are averaged over S_{test} .

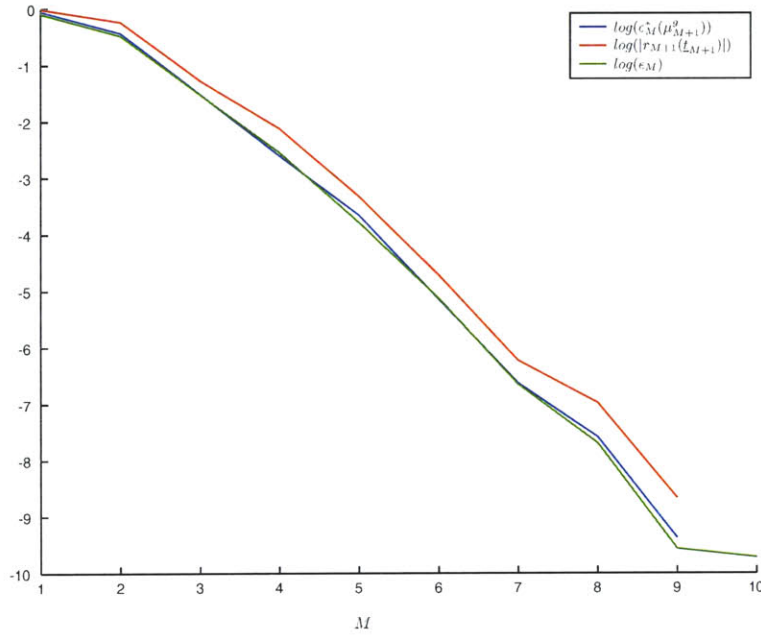


Figure 4-1: Minimax coefficient approximation method, 2D locally non-affine problem: functional parametric dependence approximation.

We note that $\epsilon_M^*(\mu_{M+1}^g) \leq |r_{M+1}(\underline{t}_{M+1})|$ which is supported by Corollary 4.1. As we can see from Figure 4-1, ϵ_M appears to be very close to the value of $\epsilon_M^*(\mu_{M+1}^g)$. There are two reasons for that. First, we choose μ_{M+1}^g which maximizes the error ϵ_M^* in (4.10), so for most points $\mu_i^t, i = 1, \dots, N_{test}$ in S_{test} $\epsilon_M^*(\mu_i^t) \leq \epsilon_M^*(\mu_{M+1}^g)$. Second, for each μ_i^t $\epsilon_M(\mu_i^t) \geq \epsilon_M^*(\mu_i^t)$ as follows from (4.10) and (4.21). These two effects counterbalance each other to produce the result we observe in Figure 4-1.

From the observation of Figure 4-1 we see that our approximation g_M becomes closer to g as M increases. We can conclude that we would want M to be big enough so that it approximates the desired functional dependence well. On the other hand, we do not want M to be too big since the on-line operation count for the error bound as it was shown Section 4.1.6 scales as $N^2(Q + M)^2$. If M becomes too large it might jeopardize the computational efficiency of our reduced basis method and our ability to fulfill the goals set in Section 1.3.1.

We provide the dependence of the condition number of \underline{B}^M and Λ_M as functions of M in Table

4.1. From Table 4.1 we also note that as M increases, the condition number of \underline{B}^M becomes rather high, at the same time Λ_M remains quite small which guarantees that $\epsilon_M^*(\mu)$ and $\epsilon_M(\mu)$ do not divert from each other too much. The reason why the condition number grows with M is explained by the fact that the vectors the columns of \underline{B}^M become more and more linear dependent since all of them are described by the same functional dependence. The alternative "orthogonalization" approach to construct \underline{B}^M is described in [5]. This approach allows to keep the condition number of \underline{B}^M low, but produces some additional complications with respect to numerical integration; more details are provided in [28, 10].

M	$\text{cond}(\underline{B}^M)$	Λ_M
1	1.00e+00	1.00
2	1.01e+01	1.00
3	2.82e+01	1.19
4	3.52e+02	2.32
5	5.50e+03	1.82
6	7.86e+04	2.54
7	2.25e+06	3.14
8	8.36e+07	5.05
9	1.33e+09	4.41
10	4.02e+10	2.83

Table 4.1: Minimax coefficient approximation method, 2D locally non-affine problem: $\text{cond}(B^M)$ and Λ_M as functions of M .

Convergence

As described in Section 2.3.3, we introduce a bound conditioner $\hat{a}(\cdot, \cdot; u)$ that is equal to

$$\hat{a}(w, v) = \sum_{q=1}^Q \bar{\Theta}_q a^q(w, v) + a^{NA}(w, v; g(\tau_{min})) =$$

$$\int_{\Omega_0} \nabla w \nabla v + 0.1 \sum_{i=1}^4 \int_{\Omega_i} \nabla w \nabla v + \int_{\Omega_{na}=\Omega_5} \exp\left(-1.1424 \sqrt{(x_1 - x_1^*)^2 + (x_1 - x_2^*)^2}\right) \nabla w \nabla v + 0.01 \int_{\Gamma_{ext} \setminus \Gamma_{root}} wv, \quad (4.83)$$

where $\bar{\Theta}_q = \min_{\mu \in \mathcal{D}} \Theta^q(\mu)$, $q = 1, \dots, Q$, $\tau_{min} = \min_{\mu \in \mathcal{D}} \tau$. We note that this bound conditioner is independent of μ , the description of the more sophisticated bound conditioners could be found in [52, 57]. From (4.77) we can easily see that $\hat{a}(w, v)$ satisfies (4.37) and hence is suitable for our purposes. For this choice of bound conditioner the constants – as given in (4.37) – assume the

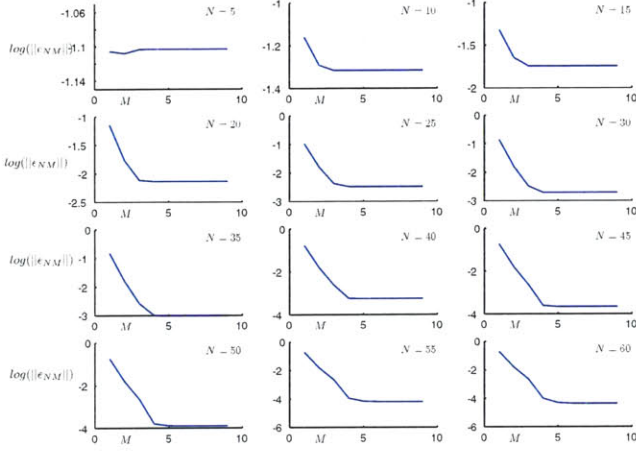


Figure 4-2: Minimax coefficient approximation method, 2D locally non-affine problem: $\|e_{NM}\|_Y$ as a function of M while N is fixed.

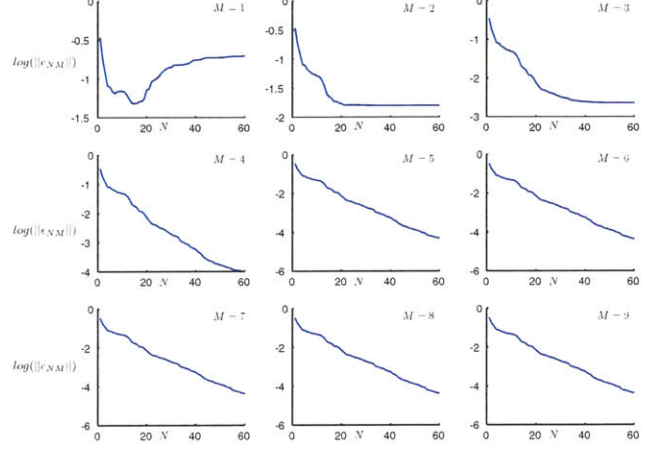


Figure 4-3: Minimax coefficient approximation method, 2D locally non-affine problem: $\|e_{NM}\|_Y$ as a function of N while M is fixed.

following values: $\alpha_0 = 0.041503$, $\rho_{min} = 1$, $\rho_{max} = 100.00$, $\gamma_0 = 10.513675$, $\nu = 0.999957$.

In Figure 4-2 we present the convergence plots for the norm of the error $\|e_{NM}\|_Y$ as a function of M for several constant values of N . The respective fixed values of N are displayed in the top right corner of each subplot of Figure 4-2. Similarly, in Figure 4-3 we provide the dependence of $\|e_{NM}\|_Y$ as a function of N while having M fixed.

As we can see from Figure 4-2, for every fixed value of N after M becomes large enough the accuracy of the solution stops to grow with M and stays at a constant level which corresponds to the flat region of each subplot. Turning to Figure 4-3 we observe that the accuracy in $\|e_{NM}\|_Y$ grows with N ; we also note that the $\|e_{NM}\|$ plots in Figure 4-3 for the values of M which correspond to the flat regions in 4-2 are practically indistinguishable from each other.

In Figure 4-4 we present the convergence plots for both $\|e_{NM}\|$ (in blue) and Δ_{NM} (in red) as functions of N for different (fixed) values of M . We observe that both $\|e_{NM}\|$ and Δ_{NM} decrease as we increase M and N . The plots for $M = 7, 8, 9$ as shown in Figure 4-4 are very close to each other both for $\|e_{NM}\|$ and Δ_{NM} .

We can thus deduce that with a high level of certainty there is no reason to increase M past a certain value since that will incur higher operation count for the online stage both through the calculation of the reduced basis approximation u_{NM} and the error bound Δ_{NM} . We are going to return to this statement later in this Chapter.

In Table 4.2 we provide a the rate of relative convergence for $\frac{\Delta_{NM}}{\|u\|}$ and $\frac{\|e_{NM}\|}{\|u\|}$ for $M = 7$. We chose this value of M based on the observation of Figures 4-2, 4-3 and 4-4.

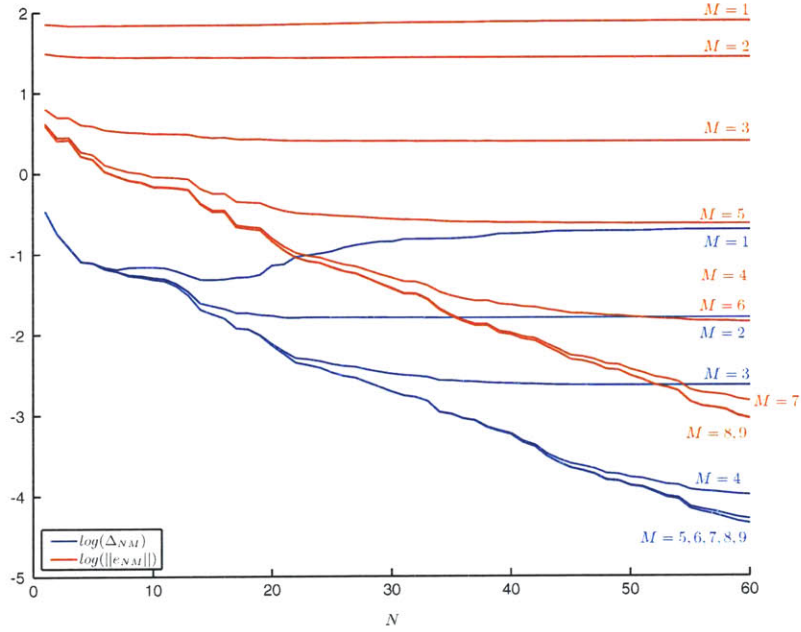


Figure 4-4: Minimax coefficient approximation method, 2D locally non-affine problem: $\|e_{NM}\|_Y$ and Δ_{NM} as functions of N while M is fixed.

We observe from Table 4.2 that with the minimax coefficient approximation method we are able to achieve the accuracy which is by all means acceptable in engineering practice.

Effectivity

In Figures 4-5 and 4-6 we present the plots of effectivity η_{NM} as defined in (4.53) in the same manner we presented the plots for $\|e_{NM}\|$. In Figure 4-5 we present η_{NM} as a function of M for fixed values of N . Furthermore, in Figure 4-6 η_{NM} is plotted as a function of M while N is fixed at different values. From Figure 4-5 we observe that as M becomes large enough η_{NM} reaches a constant level. Consistently with the last observation, we see that the effectivity plots in Figure 4-6 for $M = 6, 7, 8, 9$ are almost identical – the behavior we already observed in the case of convergence plots in Figures 4-2, 4-3, and 4-4.

We next note that the effectivity remains in the range of $[15, 2000]$ for all values of N and M and stays in the range of $[15, 40]$ for higher values of M ($M = 6, 7, 8, 9$). For the smaller values of M ($M = 1, 2, 3$) the effectivity is somewhat high which is partly explained by the L^∞ term in (4.56).

N	$\frac{\ e_{NM}\ _Y}{\ u\ }$	$\frac{\Delta_{NM}}{\ u\ _Y}$
5	6.49e-02	1.28e+00
10	4.43e-02	6.42e-01
15	1.66e-02	3.14e-01
20	6.69e-03	1.37e-01
25	3.21e-03	6.52e-02
30	1.86e-03	3.56e-02
35	9.66e-04	1.60e-02
40	5.46e-04	9.48e-03
45	2.05e-04	4.46e-03
50	1.22e-04	2.88e-03
55	6.37e-05	1.37e-03
60	4.26e-05	8.47e-04

Table 4.2: Minimax coefficient approximation method, 2D locally non-affine problem: relative convergence of $\|e_{NM}\|_Y$, Δ_{NM} as functions of N , $M = 7$.

Error Bound Decomposition

We now revisit the error equation (4.49). As it was shown in Section 4.1.5

$$\begin{aligned} \alpha(\mu)(\|e_{NM}(\mu)\|)^2 &\leq a(e_{NM}(\mu), e_{NM}(\mu); \mu) = R_1(e_{NM}(\mu); \mu) + R_2(e_{NM}(\mu); \mu) \leq \\ &\leq \tilde{\Delta}_{NM,1}(\mu) + \tilde{\Delta}_{NM,2}(\mu) + 2\sqrt{\tilde{\Delta}_{NM,1}(\mu)\tilde{\Delta}_{NM,2}(\mu)}. \end{aligned} \quad (4.84)$$

We now look how well $\tilde{\Delta}_{NM,1}$ and $\tilde{\Delta}_{NM,2}$ approximate $|R_1(e_{NM})|$ and $|R_2(e_{NM})|$, respectively. In Figures 4-7, 4-8, 4-9 and 4-10 we provide the comparison of $\tilde{\Delta}_{NM,1}$, $|R_1(e_{NM})|$ and $\tilde{\Delta}_{NM,2}$, $|R_2(e_{NM})|$ as functions of M and N in the same manner as we presented the plots for the effectivity. Our goal now is to explain what the cause of high effectivities is for the low values of M .

As we observe from Figure 4-7, for $M = 1$ the difference between $|R_1(v)|$ and $\tilde{\Delta}_{NM,1}$ is relatively big compared to the same difference for larger values of M , when the mentioned two quantities appear to be quite close to each other. As we show in our further analysis, this effect does not significantly affect the effectivity since when it occurs the contribution of $\Delta_{NM,1}$ to Δ_{NM} is very small. For $M \geq 1$ the difference between $|R_1(v)|$ and $\tilde{\Delta}_{NM,1}$ generally remains at a constant and relatively small level.

From Figures 4-9 and 4-10 we notice that the difference between $|R_2(v)|$ and $\tilde{\Delta}_{NM,2}$ remains somewhat high which is explained by the L^∞ contribution in (4.56) and the effectivity becomes small only when the relative contribution of $\tilde{\Delta}_{NM,2}$ (hence, the same applies to $\Delta_{NM,2}$) is small as

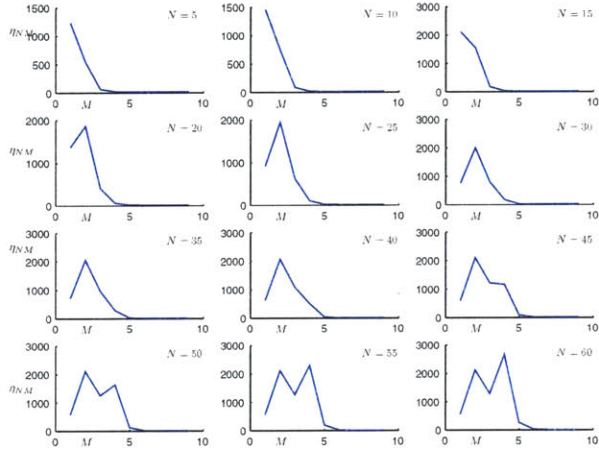


Figure 4-5: Minimax coefficient approximation method, 2D locally non-affine problem: effectiveness η_{NM} as a function of M while N is fixed.

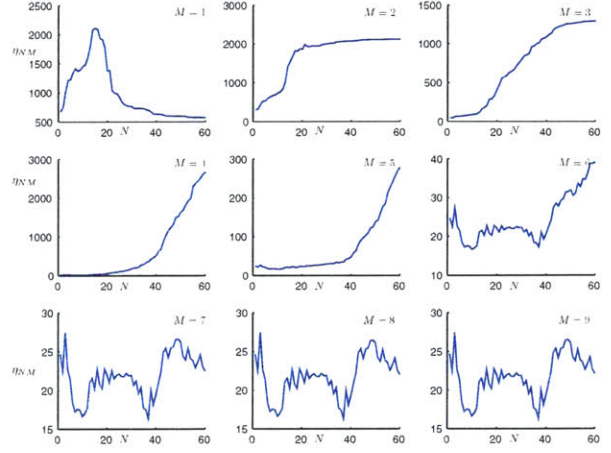


Figure 4-6: Minimax coefficient approximation method, 2D locally non-affine problem: effectiveness η_{NM} as a function of N while M is fixed.

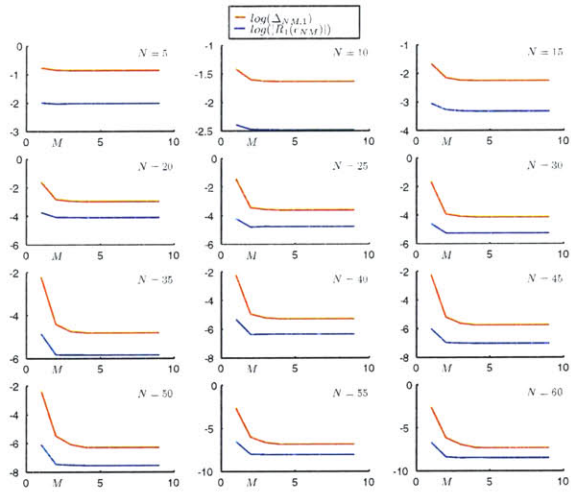


Figure 4-7: Minimax coefficient approximation method, 2D locally non-affine problem: $\hat{\Delta}_{NM,1}$ and $|R_1(e_{NM})|$ as functions of N while M is fixed.

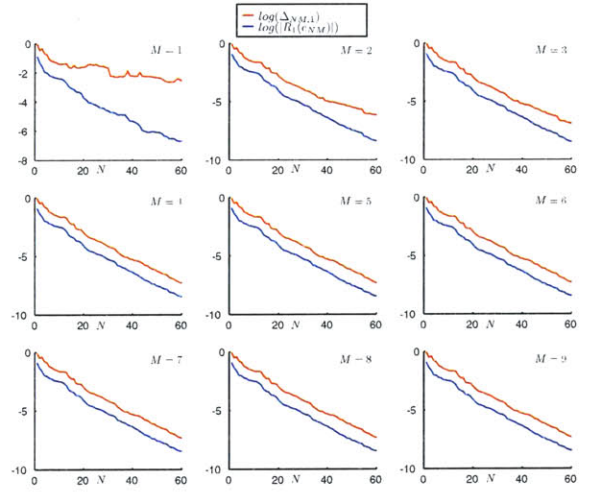


Figure 4-8: Minimax coefficient approximation method, 2D locally non-affine problem: $\hat{\Delta}_{NM,1} Y$ and $|R_1(e_{NM})|$ as functions of M while N is fixed.

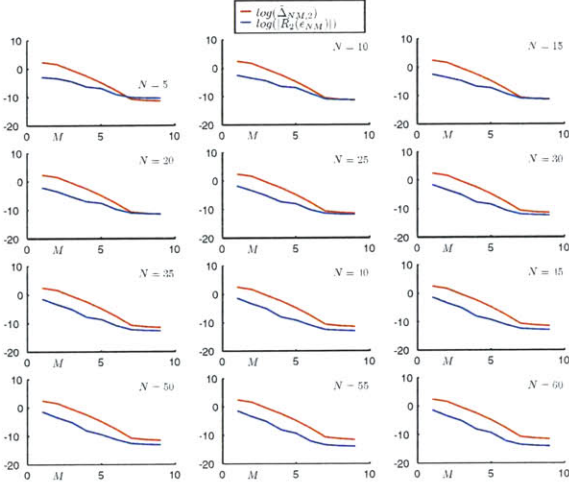


Figure 4-9: Minimax coefficient approximation method, 2D locally non-affine problem: $\tilde{\Delta}_{NM,2}$ and $|R_2(e_{NM})|$ as functions of N while M is fixed.

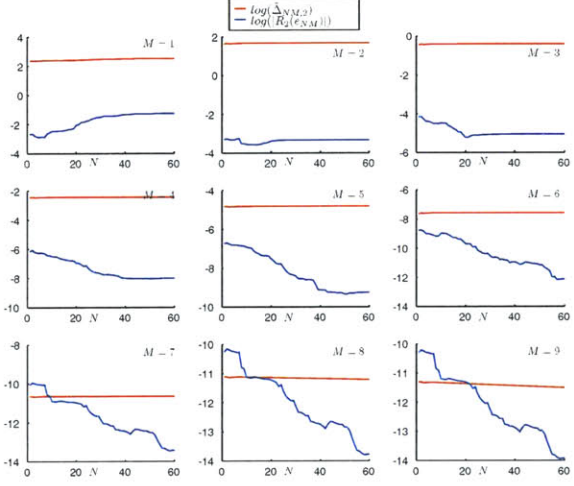


Figure 4-10: Minimax coefficient approximation method, 2D locally non-affine problem: $\tilde{\Delta}_{NM,2}$ and $|R_2(e_{NM})|$ as functions of M while N is fixed.

well. This analysis is presented in Figures 4-11 and 4-12.

Based on these observations for Figures 4-7, 4-8, 4-9, and 4-10 we can expect to have slightly higher than usual effectivity for smaller values of M . Going back to Figures 4-5 and 4-6 we see that this intuition is consistent with the actual values of effectivity η_{NM} .

It is also important to look at the relative contribution of each of the terms $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to the error bound. As we see from Figure 4-11 initially for small values of M the contribution of $\Delta_{NM,2}$ dominates that of $\Delta_{NM,1}$. However, as we keep increasing M $\Delta_{NM,1}$ and $\Delta_{NM,2}$ switch roles: now $\Delta_{NM,1}$ becomes the main contributor to the error bound Δ_{NM} whereas the contribution of $\Delta_{NM,1}$ becomes rather small.

From Figure 4-12 we observe that for $M = 1, \dots, 5$ and for small values of N $\Delta_{NM,1}$ initially constitutes the major part of Δ_{NM} . As we increase N , $\Delta_{NM,1}$ and $\Delta_{NM,2}$ trade places: $\Delta_{NM,2}$ becomes dominant while the effect of $\Delta_{NM,1}$ vanishes. That explains the fact that though in Figure 4-8 the spread between $R_1(v)$ and $\tilde{\Delta}_{NM,1}$ is rather big, the overall effectivity remains in the reasonable range, since when $M = 1$ the relative contribution of $\Delta_{NM,1}$ is quite small. We also observe that for $M = 7, 8, 9$ $\Delta_{NM,1}$ dominates $\Delta_{NM,2}$ for all values of N .

There is one more factor which might affect effectivities in a bad way. We look again at (4.84) to see that so far we did not take into account the $2\sqrt{\tilde{\Delta}_{NM,1}\tilde{\Delta}_{NM,2}}$ term which is always positive as well as $\tilde{\Delta}_{NM,1}$ and $\tilde{\Delta}_{NM,2}$; at the same time $R_1(e_{NM})$ and $R_2(e_{NM})$ can be negative.

Taking another look at Figure 4-11 we note that for each N there exists a value of M which

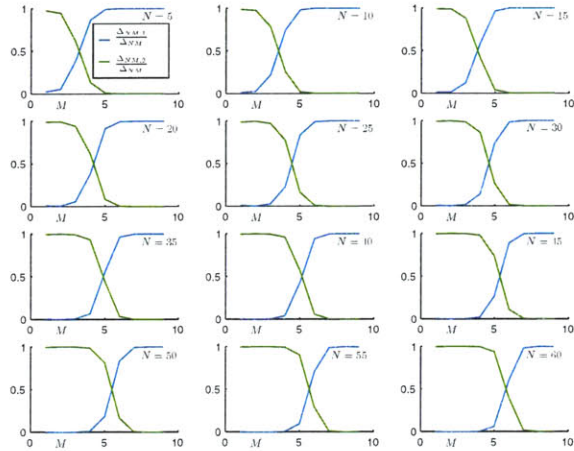


Figure 4-11: Minimax coefficient approximation method, 2D locally non-affine problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of M while N is fixed.

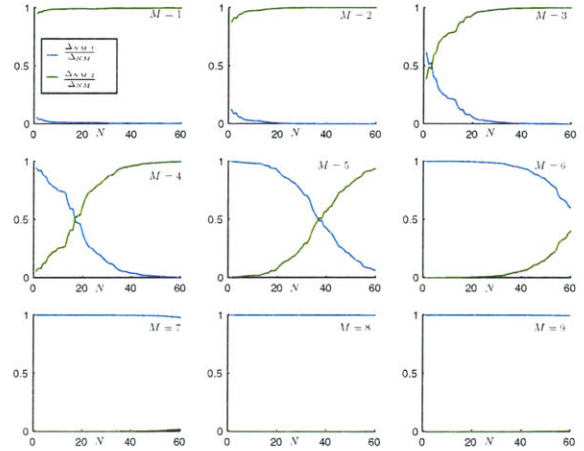


Figure 4-12: Minimax coefficient approximation method, 2D locally non-affine problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of N while M is fixed.

we denote $M^*(N)$ when the contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ are approximately equal to each other. Increasing M further past $M^*(N)$ brings us to the point where the effect of $\Delta_{NM,2}$ becomes small compared to that of $\Delta_{NM,1}$. We denote this second characteristic value of M as $M^{**}(N)$. To be more rigorous,

$$M^{**}(N) = \arg \min_M \frac{\Delta_{NM,2}}{\Delta_{NM}} \leq 0.01. \quad (4.85)$$

In Figure 4-13 we present the plots for M^* and M^{**} as functions of N . Clearly, $\forall N, M^*(N) < M^{**}(N)$.

Further $M - N$ analysis

Taking into consideration the operation count of $O(M^2) + O((Q + M)N^2) + O(N^3) + O((Q + M)^2N^2) + O(Mn_{na})$ for the on-line stage as it was shown in Section 4.1.6 our goal would be to choose N and M as small as possible and at the same provide the good accuracy. As we can see from the convergence plots 4-3, $\|e_{NM}\|$ is a strictly decreasing function of N . Thus, in order to obtain the best accuracy we should choose N as large as possible. However, in an arbitrary case the user might have certain time constraints which might prevent him or her from choosing the largest value of N . As we can see, there is a trade-off between the accuracy of the reduced basis approximation and the on-line computational complexity.

However, if somehow the value of N is chosen, we need to determine the optimal value of M for this particular value of N . To maximize the accuracy while keeping M as small as possible our

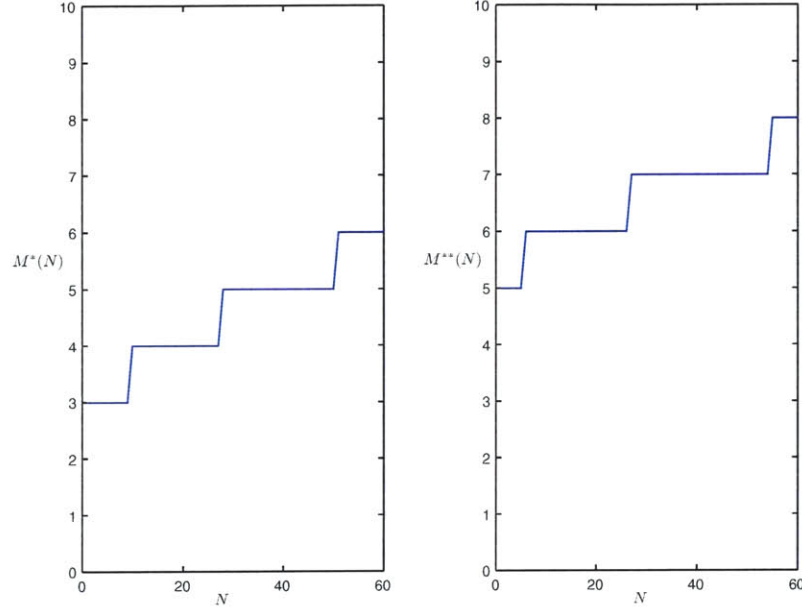


Figure 4-13: Minimax coefficient approximation method, 2D locally non-affine problem: M^* and M^{**} as functions of N .

suggestion is to choose M to be $M^* \leq M \leq M^{**}$. Such a choice of M is supported by Figure 4-2 – for these M the accuracy essentially reaches its best value, it is slightly better for M^{**} than for M^* ; the difference is relatively small as we can see from the following convergence plots as we plot e_{NM} and Δ_{NM} as functions of N and $M^*(N)$, $M^{**}(N)$ in Figure 4-14.

We now provide similar plots for the effectivity as a function of N and the choice of M equal to $M^*(N)$ and $M^{**}(N)$. We note from Figure 4-14 that the choice of M^{**} yields sharper effectivity than that of M^* , the difference ranges from the factor of 1.4 to 6 which is consistent with the definitions of M^* and M^{**} .

Computational Costs

In Figure 4-16 we present the plots for $t_{u_{NM}}$ and $t_{\Delta_{NM}}$ the actual computational times required to obtain u_{NM} and Δ_{NM} as functions of M for several different values of N . In the same Figure we provide t_{tot} , the total computational time for the on-line stage, as a function of M .

As we observe from Figure 4-16, $t_{\Delta_{NM}}$ displays stronger dependence on M than $t_{u_{NM}}$ which is consistent with their operation counts $O(M^2) + O((Q + M)N^2) + O(N^3)$ and $O((Q + M)^2N^2) + O(Mn_{na})$, respectively. At the same time both $t_{u_{NM}}$ and $t_{\Delta_{NM}}$ exhibit strong dependence on N .

In Figure 4-17 we provide the dependence of t_{tot} for the cases of $M = M^*(N)$ and $M = M^{**}(N)$. As we can see the difference in time is rather small and – since in the case of M^{**} we have sharper

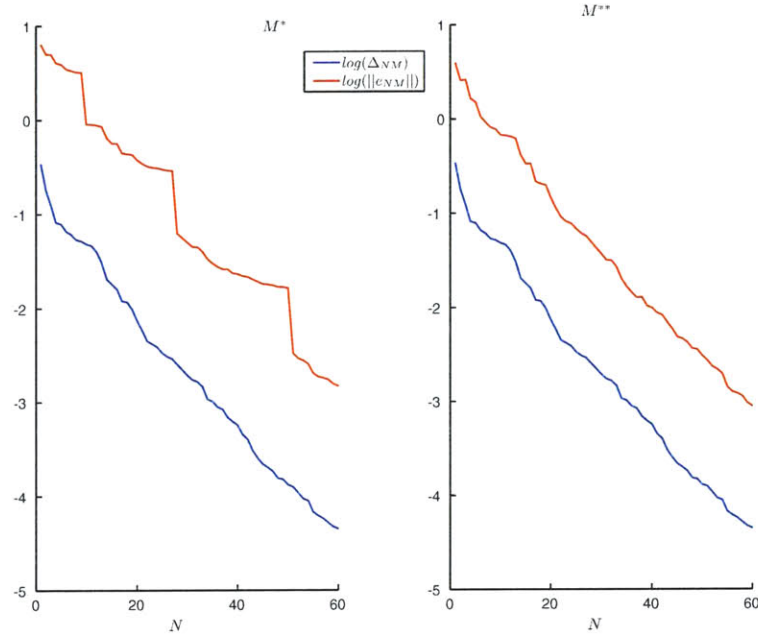


Figure 4-14: Minimax coefficient approximation method, 2D locally non-affine problem: $\|e_{NM}\|_Y$ and Δ_{NM} as functions of N and $M^*(N)$, $M^{**}(N)$.

error bounds – for this particular model problem it makes more practical sense to choose $M = M^{**}$. Even for the highest values of M and N we are able to reduce the computational time approximately by a factor of 150: $\frac{t_{tr}}{t_{tot, M^{**}}} = \frac{2.30}{0.0154}$ which means that in this case the computational savings are quite significant.

Conclusions

Based on the numerical results provided in Section 4.2.1 we can state that the minimax coefficient approximation method provides an efficient and accurate way to solve coercive problems with locally non-affine parametric dependence. The minimax coefficient approximation method yields good results in terms of convergence, error estimation and the sharpness of the error bounds and at the same time provides significant computational savings. Overall, the method is able to fulfill the goals that were set in Chapter 1. We now proceed to the comparison of the MCAM and the PUM so that we can focus on advantages and disadvantages of each method.

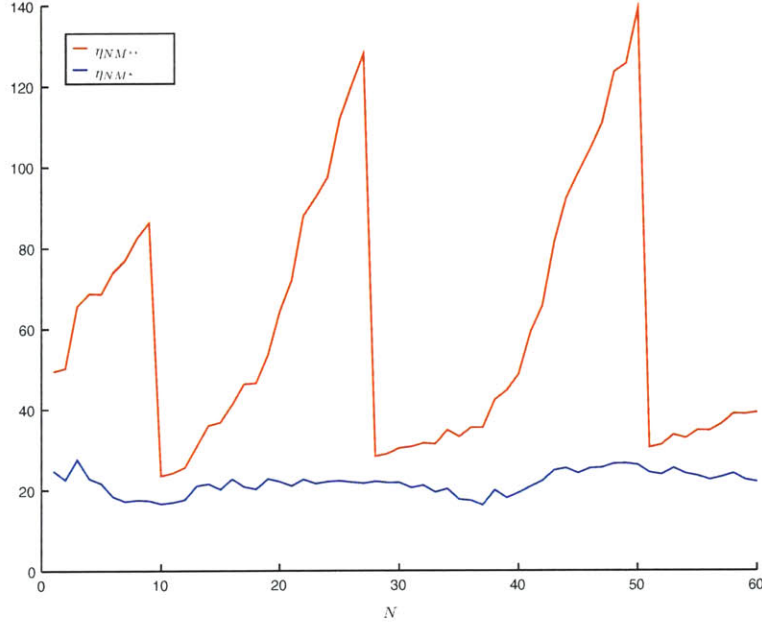


Figure 4-15: Minimax coefficient approximation method, 2D locally non-affine problem: effectivity η_{NM} as a function of N and $M^*(N)$, $M^{**}(N)$.

4.2.3 Comparison of Partition of Unity and Minimax Coefficient Approximation Methods

In this Section we provide the comparison between the partition of unity and the minimax coefficient based on their performance with respect to the model problem we introduced in this Chapter.

Convergence and Effectivity

In the left part of Figure 4-18 we present the convergence plots for the partition of unity method: the norm of the error $\|e_N\|_Y$ (in blue) and the error bound Δ_{NS} (in red) the construction of which was explained in Section 3.2.3. In the right part of the Figure we present similar plots for the minimax coefficient approximation method: for each value of N we select $M = M^{**}(N)$ as given by (4.85). Based on this choice of N and M we calculate the values of $\|e_{NM}\|_Y$ and Δ_{NM} according to Section 4.1.5.

As we observe from Figure 4-18, the PUM yields slightly better results than the MCAM both in terms of the norm of the error $\|e_N\|$ and the error bound Δ_{NS} , especially for the smaller values of N . This behavior is explained by the fact that the reduced basis approximation $\tilde{u}_N(\mu)$ as introduced in Section 3.2.2 is comprised of members of both W_N and X_S , whereas $u_{NM}(\mu)$ is only a member of W_N . As we increase N , this slight advantage of PUM vanishes and the convergence plots for

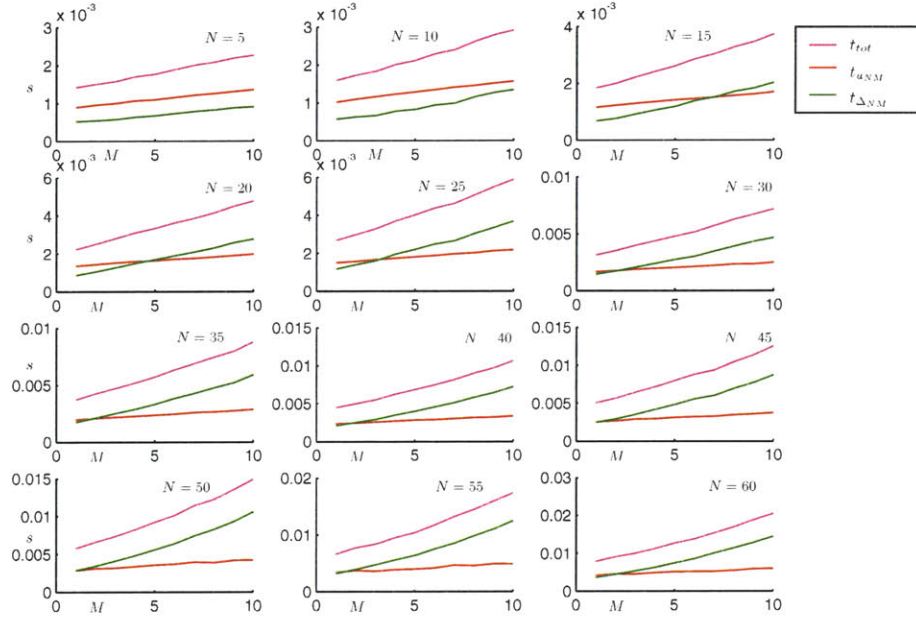


Figure 4-16: Minimax coefficient approximation method, 2D locally non-affine problem: on-line computational times as functions of M while N is fixed.

both methods become practically indistinguishable from each other. This same behavior is also reflected by the plots of the effectivities in Figure 4-19.

Computational Costs

In Tables 4.3 we present the actual total time for the on-line stage for both the PUM and MCAM as functions of $\|e_N\|$, $\|e_{NM}\|$, i.e. in such a way that $\|e_N\| \approx \|e_{NM}\|$ (the difference is contained within several percent at most, note the leftmost and the rightmost columns of Table 4.3).

We notice that the MCAM performs much better time-wise which is consistent with the on-line numerical algorithms for both methods. Comparing the on-line operation counts for the MCAM: $O(M^2) + O((Q + M)N^2) + O(N^3) + O((Q + M)^2N^2) + O(Mn_{na})$, and the PUM: $O(QN^2) + O(n_{na}N^2) + O(N^3) + O(QS) + O(n_{na}S^2) + O(QSN) + O(n_{na}SN) + O(S^\kappa) + O(\sum_{i=1}^J Q_i^2(S_i + N)^2) + O(Gn_{na}^\kappa)$, we observe that the main difference in the on-line computational times is explained by the terms responsible for the evaluation of the non-affine contribution: $O(Mn_{na})$ for the MCAM and $O(Gn_{na}^\kappa)$ for the PUM. It is much cheaper to form a vector of size n_{na} and determine its maximum value (the MCAM) than to form and solve system of size $n_{na} \times n_{na}$ (the PUM).

In practice, we of course do not have a value of $\|e_{NM}\|$ since its calculation requires solution of the original problem and we are trying to avoid that. Hence, we can only operate in terms of error

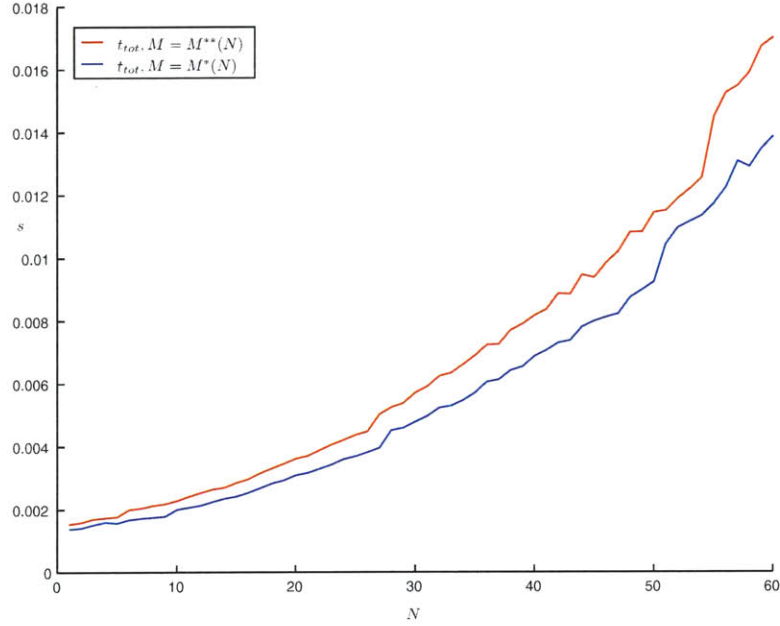


Figure 4-17: Minimax coefficient approximation method, 2D locally non-affine problem: t_{tot} , the total on-line computational times as a function of N and $M^*(N)$, $M^{**}(N)$.

bound $\Delta_{NM}(\mu)$. In Table 4.4 we provide the actual on-line computational times for the PUM and the MCAM for close values of $\Delta_{NM} \approx \Delta_{NS}$ which stay within several percent from each other. Thus, we determine the relevant N and M based on the value of the error bound which means that the Table 4.4 is a sense more important than Table 4.3. However, the PUM holds a clear advantage over the MCAM with respect to the time required to run the off-line stage since the precomputing of the matrices for error estimation for the PUM – as we discussed in Chapter 3 – requires solution of the much smaller systems of equations than that for the MCAM. This observation follows from comparison of numerical algorithms in Sections 3.2.4 and 4.1.6 (compare steps 4 and 5).

As mentioned earlier in this Chapter, the $O(Q + M)^2 N^2$ term in the operation count imposes a restriction on the parametric complexity of the problems we can treat with the MCAM. This also applies to problems where the affine and non-affine dependencies are interwoven with each other – e.g., see the problem we introduced in Section 3.3 – which results even in a greater operation count: $O(Q + \beta M)^2 N^2$, where β is a whole number greater than 1.

Conclusions

For the model problem considered in this Chapter the MCAM and the PUM demonstrate essentially the same rate of convergence and the sharpness of error bounds; at the same time the MCAM shows

MCAM			"truth"	PUM		
$\ e_{NM}\ _Y$	t_{tot}	$\frac{t_{tr}}{t_{tot}}$	t_{tr}	$\frac{t_{tr}}{t_{tot}}$	t_{tot}	$\ e_N\ _Y$
1.23e-01	1.70e-03	1352.30	2.30	21.67	1.06e-01	1.28e-01
8.19e-02	1.74e-03	1323.06	2.30	20.75	1.11e-01	8.18e-02
5.38e-02	2.13e-03	1079.51	2.30	20.36	1.13e-01	5.41e-02
1.80e-02	2.86e-03	804.53	2.30	18.41	1.25e-01	1.76e-02
5.78e-03	3.72e-03	618.98	2.30	17.24	1.33e-01	6.44e-03
3.88e-03	4.22e-03	544.51	2.30	16.61	1.38e-01	3.88e-03
1.66e-03	6.25e-03	367.81	2.30	15.09	1.52e-01	1.68e-03
8.47e-04	7.26e-03	316.79	2.30	14.37	1.60e-01	8.44e-04
5.73e-04	8.18e-03	281.32	2.30	13.87	1.66e-01	5.56e-04
2.20e-04	9.38e-03	245.32	2.30	12.84	1.79e-01	2.12e-04
9.02e-05	1.25e-02	183.29	2.30	11.59	1.98e-01	8.96e-05
6.27e-05	1.53e-02	150.77	2.30	11.24	2.05e-01	6.55e-05
4.53e-05	1.70e-02	135.19	2.30	10.76	2.14e-01	4.56e-05

Table 4.3: Comparison of partition of unity method and minimax coefficient approximation method: on-line computational times as functions of error bound.

MCAM			"truth"	PUM		
Δ_{NM}	t_{tot}	$\frac{t_{tr}}{t_{tot}}$	t_{tr}	$\frac{t_{tr}}{t_{tot}}$	t_{tot}	Δ_{NS}
1.65e+00	1.74e-03	1323.06	2.30	20.66	1.11e-01	1.74e+00
8.14e-01	2.13e-03	1079.51	2.30	20.54	1.12e-01	8.44e-01
6.18e-01	2.65e-03	867.01	2.30	18.76	1.23e-01	6.06e-01
4.13e-01	2.71e-03	848.71	2.30	18.41	1.25e-01	3.85e-01
1.11e-01	3.72e-03	618.98	2.30	17.24	1.33e-01	1.22e-01
8.12e-02	4.08e-03	564.36	2.30	17.04	1.35e-01	8.17e-02
5.64e-02	5.04e-03	456.69	2.30	16.10	1.43e-01	5.71e-02
2.01e-02	6.63e-03	347.14	2.30	14.97	1.54e-01	2.11e-02
8.72e-03	8.37e-03	274.89	2.30	13.87	1.66e-01	9.00e-03
5.91e-03	9.47e-03	242.93	2.30	13.26	1.73e-01	6.18e-03
3.66e-03	1.08e-02	212.59	2.30	12.50	1.84e-01	3.74e-03
1.97e-03	1.25e-02	183.29	2.30	11.59	1.98e-01	1.98e-03
8.93e-04	1.70e-02	135.19	2.30	10.76	2.14e-01	9.17e-04
--	--	--		10.53	2.18e-01	8.37e-04

Table 4.4: Comparison of the partition of unity and the minimax coefficient approximation methods: on-line computational times as functions of Δ_{NS} and Δ_{NM} .

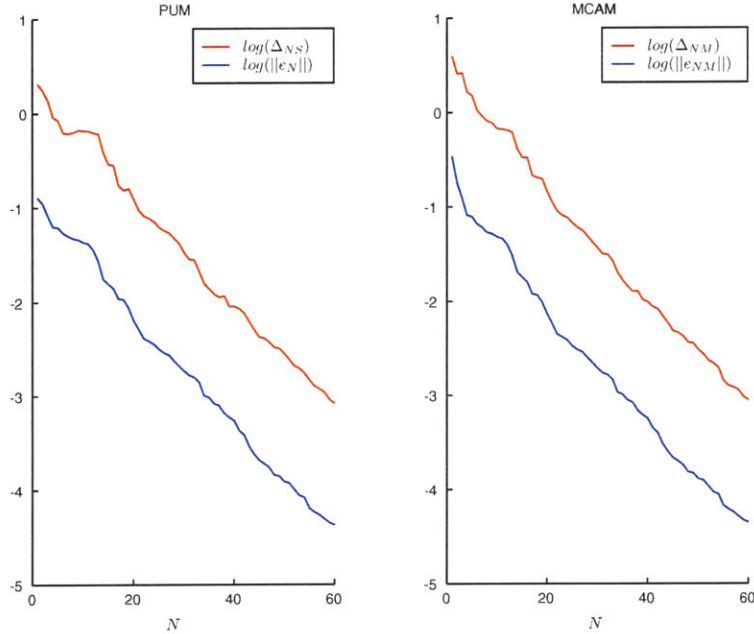


Figure 4-18: Comparison of the partition of unity (left) and the minimax coefficient approximation (right) methods: $\|e_N\|_Y$ and Δ_{NS} as functions of N ; $\|e_{NM}\|_Y$ and Δ_{NM} as functions of N and $M^{**}(N)$.

significantly better computational times for the on-line stage. Another advantage of the minimax coefficient approximation method is that its numerical algorithm for the on-line stage does not deal with the specifics of the locally non-affine dependence. In general, the processing of the non-affine dependence can be quite time consuming and might depend on the computational and programming means available to the user. For example, if we use the partition of unity method for the model problems studied in Sections 3.3 and 4.2 it is necessary to perform the numerical integration of the locally non-affine component every time we run the on-line stage. The MCAM allows us to take care of this issue during the off-line stage, whereas during the online stage only standard matrix and vector operations are required.

However, the minimax coefficient approximation method has certain disadvantages as compared to the partition of unity method. The operation count for the error bound Δ_{NM} scales at least as $(Q + M)^2 N^2$ which imposes a much harder restriction on how large Q can be so that the method remains computationally efficient. That is the reason why in this Chapter we had to consider a simplified version of the model problem introduced in Section 3.3 and to reduce the number of affine parameters Q from 26 to 6. This drawback becomes especially evident when the affine and non-affine dependencies are interwoven with each other – as they are in Section 3.3 – which

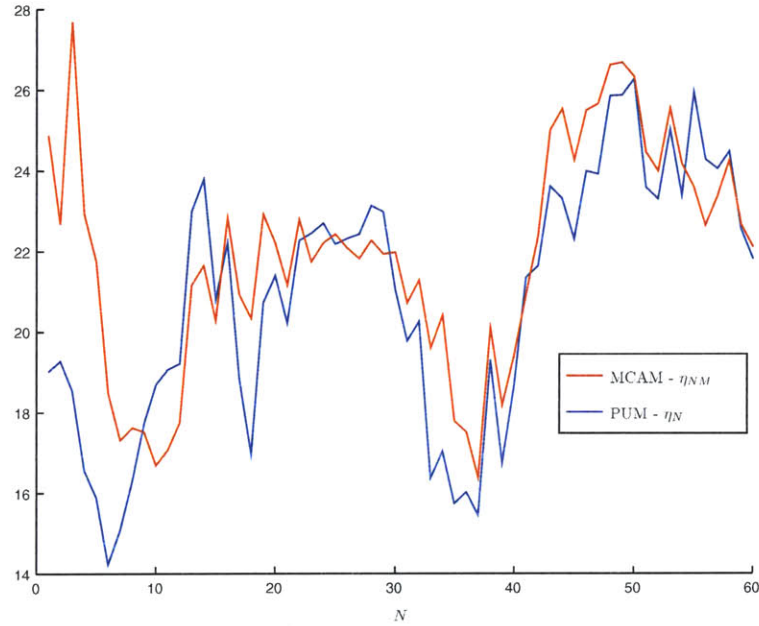


Figure 4-19: Comparison of the partition of unity and the minimax coefficient approximation methods, effectivities η_N as function of N and η_{NM} as a function of N and $M^{**}(N)$.

results in an additional multiplier $\beta > 1$ in the operation count: $(Q + \beta M)^2 N^2$. For these more parametrically complex problems we expect the PUM to be advantageous to the MCAM both as regards the off-line and on-line stages.

Generally, both methods are capable of addressing locally non-affine problems. Based on the analysis of the numerical results we performed in Chapters 3 and 4 we can say the partition of unity method better suits the cases when the number affine parameters, Q , is relatively big ($Q > 10$), whereas the minimax coefficient approximation method will work better for the cases of smaller Q ($Q < 10$).

Chapter 5

Locally Non-linear Problems: Minimax Coefficient Approximation Method

In Chapter 2, 3, and 4 we described application of the reduced basis methods to problems which are governed by linear partial differential equations. In this Chapter we address problems which are governed by non-linear partial differential equations and describe application of the modified minimax coefficient approximation method to such problems. We limit the generality of the problems we consider to locally non-linear problems, the rigorous definition of which is given later in this Chapter.

5.1 General Abstract Problem Statement

We start by briefly repeating the notation we use throughout this thesis. As before, we consider a suitably regular (smooth) parameter-independent domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, or 3 , and associated function space $Y \subset (H^1(\Omega))^p$ with inner product $(\cdot, \cdot)_Y$, norm $\|\cdot\|_Y = (\cdot, \cdot)_Y^{1/2}$, and dual space Y' ; we define a parameter set $\mathcal{D} \subset \mathbb{R}^P$, a particular point in which will be denoted μ .

As it was indicated in the previous Chapters, for the cases when the underlying strong formulation of the governing partial differential equations is linear, the form $a(\cdot, \cdot; \cdot)$ corresponding to the weak formulation of the same equations is bilinear. However, when the underlying strong formulation is non-linear, the form $a(\cdot, \cdot; \cdot)$ loses the property of bilinearity.

In general, we are going to address *well-posed* problems the weak formulation of which is given by

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in Y, \quad (5.1)$$

where the form $a(\cdot, \cdot; \cdot) : Y \times Y \rightarrow \mathbb{R}$ allows the following decomposition

$$a(w, v; \mu) = a^A(w, v; \mu) + a^{NL}(w, v; \mu), \quad \forall w, v \in Y, \quad (5.2)$$

where, in turn, $a^A(w, v; \mu)$ is a bilinear, symmetric, coercive:

$$0 < \alpha_0 \leq \alpha(\mu) = \inf_{v \in Y} \frac{a^A(v, v; \mu)}{\|v\|_Y^2}, \quad \forall \mu \in \mathcal{D}, \quad (5.3)$$

and continuous:

$$a^A(w, v; \mu) \leq \gamma(\mu) \|w\|_Y \|v\|_Y \leq \gamma_0 \|w\|_Y \|v\|_Y, \quad \forall \mu \in \mathcal{D}, \quad \forall w, v \in Y \quad (5.4)$$

form as described in Chapters 2, 3 and 4 that admits the regular affine decomposition

$$a^A(w, v; \mu) = \sum_{q=1}^Q \Theta_q(\mu) a^q(w, v), \quad \forall w, v \in Y; \quad (5.5)$$

and $a^{NL}(g(w), v; \mu)$ can be expressed as follows

$$a^{NL}(w, v; \mu) = \int_{\Omega_{nl}} g(w; \mu) v, \quad \forall w, v \in Y, \quad (5.6)$$

where $\Omega_{nl} \subset \Omega$ and $\forall \mu \in \mathcal{D} \ g(w(\underline{z}); \mu) \in C^\infty(\bar{\Omega}) : \mathbb{R} \times \mathcal{D} \rightarrow \mathbb{R}, \underline{x} \in \mathbb{R}^d$. We require that the integral defined in (5.6) is *well-defined*¹ $\forall w, v \in Y$. In some cases the fact that the integral (5.6) is well defined could be verified by invoking Sobolev's embedding theorems and trace theorems [17, 45, 1]. Another requirement we impose is that $g(\cdot, \cdot)$ is *monotonically increasing* in its *first* argument:

$$\forall y_1, y_2 \in \mathbb{R} : y_2 \geq y_1, \text{ then } g(y_2; \mu) \geq g(y_1; \mu). \quad (5.7)$$

¹Generally speaking, the situation when (5.6) is not well-defined $\forall w, v \in Y$ (which is a necessary condition for the well-posedness of the problem) in some cases could be addressed by selecting a different function space Y . At the moment we assume that $Y = H^1(\Omega)$, however, the methodology we present in this thesis could be extended to other function spaces which broadens the classes of problems to which our methods could be applied.

We only apply our reduced basis methods to locally non-linear problems, i.e. problems for which the area of Ω_{nl} is small compared to Ω . In fact, Ω_{nl} might be of lower dimension than Ω , for example it can be a part of the boundary of Ω . From the finite element method standpoint we require that the number of nodes in the discretization of Ω_{nl} – which we denote n_{nl} – is small compared to \mathcal{N} , the total number of nodes in the discretization of Ω . To be more precise, we require that $n_{nl} \leq \frac{\mathcal{N}}{10}$.

5.1.1 Newton Method

To ease the future referencing we first provide a brief description of the Newton method which is used to solve non-linear systems of equations numerically. We consider an arbitrary system of equations

$$\begin{aligned} F_1(z_1, \dots, z_n) &= 0, \\ &\vdots \\ F_n(z_1, \dots, z_n) &= 0, \end{aligned} \tag{5.8}$$

or, switching to the vector notation

$$\underline{F}(\underline{z}) = 0, \tag{5.9}$$

where $\underline{z}, \underline{F} \in \mathbb{R}^n$. The solution \underline{z}^* of (5.8) is obtained through the following iterative procedure

$$\underline{J}_F(\underline{z}^k)(\underline{z}^{k+1} - \underline{z}^k) = -\underline{F}(\underline{z}^k), \tag{5.10}$$

where $\underline{J}_F(\underline{z}^k)$ is an $n \times n$ Jacobian matrix,

$$J_{F_{ij}}(\underline{z}^k) = \frac{\partial F_i}{\partial x_j}(\underline{z}^k). \tag{5.11}$$

We define $\delta \underline{z}^k$ as $\delta \underline{z}^k = \underline{z}^{k+1} - \underline{z}^k$ and rewrite (5.10) as

$$\underline{J}_F(\underline{z}^k)\delta \underline{z}^k = -\underline{F}(\underline{z}^k). \tag{5.12}$$

We stop the procedure (5.12) once $\|\delta \underline{z}^k\| = \|\underline{z}^{k+1} - \underline{z}^k\|$ and $\|\underline{F}(\underline{z}^k)\|$ both reach certain thresholds.

5.1.2 Limitations of Classical Reduced Basis Method

We first would like to mention some of the previous and contemporary studies of reduced basis methods applied to non-linear problems. application of the reduced basis method to Burgers equation and Navier-Stokes equations is presented in [55] and [54, 49], respectfully. In [56] a simple "Galerkin" reduced basis method is described and subsequently applied to a problem with cubic non-linearity. In this Section we review this method in the context of problems with a general power non-linearity and point to the weak sides and limitations of this method. Also in [28] a study is presented of the reduced basis method which relies on the minimax coefficient approximation ideas [5] in application to *globally* non-affine and *globally* non-linear problems; the key difference between [28] and this thesis is that we address *locally* non-affine and *locally* non-linear problems for which we are able to provide absolutely rigorous error bounds, whereas the error estimation in [28] is obtained by admitting a certain loss of rigor.

Turning to the specifics of the problems for the "Galerkin" reduced basis method we require that $a^{NL}(w, v; \mu)$ should satisfy all the regularity requirements of Section 5.1; furthermore $a^{NL}(w, v; \mu)$ can be represented in the following way

$$a^{NL}(w, v; \mu) = \Theta_{nl}(\mu) \int_{\Omega_{nl}} w^\kappa v, \quad (5.13)$$

where $\kappa \in \mathbb{Z}^+$.² Under these assumptions we proceed to the description of the method itself.

Reduced Basis Approximation

We start by introducing a sample in parameter domain \mathcal{D}

$$S_N = \{\mu_1, \dots, \mu_N\}, \quad (5.14)$$

where $\mu_i \in \mathcal{D}, i = 1, \dots, N$. We then define our reduced-basis approximation space W_N as

$$W_N = \text{span}\{\zeta_i \equiv u(\mu_i), i = 1, \dots, N\}, \quad (5.15)$$

²Revisiting the issue of whether (5.13) is well-defined $\forall v, w \in Y$, depending on the value of κ and the dimension of the problem it might be necessary in some cases to switch to function space different from $H^1(\Omega)$.

where $u(\mu_i) \in Y$ is the solution to (5.1) for $\mu = \mu_i$. For any $\mu \in \mathcal{D}$, we define our reduced-basis approximation for the field solution as $u_N(\mu) \in W_N$ which is obtained through Galerkin projection of $u(\mu)$ onto W_N

$$a(u_N(\mu), v) = f(v), \quad \forall v \in W_N. \quad (5.16)$$

As (5.2), the formulation (5.16) is a non-linear system of equations, the advantage of the latter is that the number of unknowns is equal to N , the factor which we expect will reduce the operation count per Newton iteration. Since we require that $u_N(\mu) \in W_N$, $u_N(\mu)$ admits the following decomposition

$$u_N(\mu) = \sum_{i=1}^N \beta_i(\mu) \zeta_i. \quad (5.17)$$

Using the expression for $u_N(\mu)$ we can write the polynomial expression for $(u_N(\mu))^\kappa$ as

$$(u_N(\mu))^\kappa = \sum_{i_1=1}^N \dots \sum_{i_\kappa=1}^N \beta_{i_1} \dots \beta_{i_\kappa} \zeta_{i_1} \dots \zeta_{i_\kappa}. \quad (5.18)$$

It is an easy matter to show that we can rearrange the terms in (5.18) in such a way that

$$(u_N(\mu))^\kappa = \sum_{\underline{j} \in \mathcal{P}_\kappa^N} c_{\underline{j}} \beta_{j_1} \dots \beta_{j_\kappa} \zeta_{j_1} \dots \zeta_{j_\kappa}, \quad (5.19)$$

where \mathcal{P}_κ^N is the set of uniquely ordered κ -tuples of integers $1, \dots, N$ such that for each $\underline{j} \in \mathcal{P}_\kappa^N$: $j_1 \leq j_2 \leq \dots \leq j_\kappa$. The cardinality of \mathcal{P}_κ^N is equal to³

$$T(N, \kappa) = \frac{(N-1+\kappa)!}{(N-1)!\kappa!}. \quad (5.20)$$

For large values of N $T(N, \kappa) \sim \frac{N^\kappa}{\kappa!}$. We next plug (5.19) into (5.16) to obtain the following non-linear system of equations

$$\sum_{k=1}^N A_{N_i k}^A \beta_k + \Theta_{nl}(\mu) \sum_{\underline{j} \in \mathcal{P}_\kappa^N} c_{\underline{j}} \beta_{j_1} \dots \beta_{j_\kappa} \int_{\Omega_{nl}} \zeta_{j_1} \dots \zeta_{j_\kappa} = f(\zeta_i), \quad i = 1, \dots, N. \quad (5.21)$$

As we see, this formulation is consistent with (5.8) and (5.10). It is a simple matter to demonstrate that in this case the assembly of the right hand side of (5.10) requires $O(QN^2) + O(\frac{N^{\kappa+1}}{(\kappa-1)!})$

³The proof of this fact is provided in Appendix A.

operations, we also have the same operation count for the assembly of the Jacobian.

We note that though we hope that N will remain small, the operation count of $O(QN^2) + O(\frac{N^{\kappa+1}}{(\kappa-1)!})$ can be quite high especially for larger values of κ .

Error Estimation

We then define the error function $e_N(\mu)$ as $e_N(\mu) = u(\mu) - u_N(\mu)$ and using (5.1) and (5.16) we arrive at the following error equation which holds $\forall v \in Y$ (compare to (2.21), (3.32), (3.33), (3.128) and (4.40))

$$a^A(e_N(\mu), v; \mu) + a^{NL}(u(\mu), v; \mu) - a^{NL}(u_N(\mu), v; \mu) = \underbrace{f(v) - a(u_N(\mu), v; \mu)}_{R(v; \mu)}. \quad (5.22)$$

We denote $f(v) - a(u_N(\mu), v; \mu)$ as $R(v; \mu)$. Using (5.19), the polynomial expansion, and (5.5), the affine decomposition, we can rewrite $R(v; \mu)$ as

$$R(v; \mu) = f(v) - \sum_{q=1}^Q \sum_{i=1}^N \Theta_q(\mu) \beta_i(\mu) a^q(\zeta_i, v; \mu) - \Theta_{nl}(\mu) \sum_{\underline{j} \in \mathcal{P}_\kappa^N} c_{\underline{j}} \beta_{j_1} \dots \beta_{j_\kappa} \int_{\Omega_{nl}} \zeta_{j_1} \dots \zeta_{j_\kappa} v. \quad (5.23)$$

Since $g(., .)$ is monotonic in its first argument we note that

$$a^{NL}(u(\mu), e_N(\mu); \mu) - a^{NL}(u_N(\mu), e_N(\mu); \mu) = \Theta_{nl}(\mu) \int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_N(\mu); \mu))(u(\mu) - u_N(\mu)) \geq 0. \quad (5.24)$$

We next plug $v = e_N(\mu)$ into (5.22) and using (5.3) – the coercivity of $a^A(., .; \mu)$ – and (5.24) we arrive at the following result (for more details on the dual norm definition and its relation to bound conditioners see Sections 2.1 and 2.3.3)

$$\alpha(\|e_N(\mu)\|_Y)^2 \leq \|R(v; \mu)\|_{Y'} \|e_N(\mu)\|_Y. \quad (5.25)$$

We readily construct the error bound $\Delta_N(\mu)$ as

$$\|e_N(\mu)\|_Y \leq \Delta_N(\mu) = \frac{1}{\alpha} \|R(v; \mu)\|_{Y'}. \quad (5.26)$$

However, construction of $\Delta_N(\mu)$ requires evaluation of $\|R(v; \mu)\|_{Y'}$. The calculation of $\|R(v; \mu)\|_{Y'}$ – as it was shown in Sections 2.3.3 and 2.3.5 – depends on the number of terms in the decomposition

of $R(v; \mu)$ (5.23) which is equal to $K = 1 + QN + \frac{(N-1+\kappa)!}{(N-1)!\kappa!}$ which is roughly equal to $QN + \frac{N^\kappa}{\kappa!}$ for larger values of N . Thus, the operation count for the error bound scales as K^2 which is a very big number even for moderate values of N . Even for the case when $\kappa = 3$ this becomes extremely computationally expensive.

Thus, the major fallacies of the extension of the classical reduced basis method are (i) the impossibility to treat non-linearities different from power non-linearities described in (5.13) and (ii) the extremely high operation count for both Newton steps and the error bound evaluation. Further in this Chapter we present a new reduced basis method based on the minimax coefficient approximation ideas [5] which allows us to resolve these difficulties.

5.2 Minimax Coefficient Approximation Method

We now switch back to the more general case of a non-linear problem as described in Section 5.1. We explain the procedure that allows to us to replace the non-linear term (5.6) with an affine-like approximation. application of the minimax coefficient approximation method to the non-linear case is in fact very similar to how it was done in the non-affine case in Chapter 4 and essentially repeats the same steps as laid in [5].

5.2.1 S_M^g Sample Selection

We begin from building a recursive sequence μ_M^g , $M = 1, \dots$. We start from some random μ_1^g , set $\xi_1(\underline{x}) = g(u(\underline{x}; \mu_1^g); \mu_1^g)$ and then generate μ_2^g, μ_3^g, \dots , using the following recursive formula

$$\mu_M^g = \arg \max_{\mu \in S_D} \epsilon_{M-1}^*(\mu), \quad (5.27)$$

$$\xi_M(\underline{x}) = g(u(\underline{x}; \mu_M^g); \mu_M^g), \quad (5.28)$$

where

$$\epsilon_M^*(\mu) = \min_{\gamma \in \mathbb{R}^M} \max_{x \in \Omega_{nt}} |g(u(\mu); \mu) - \sum_{j=1}^M \gamma_j \xi_j(\underline{x})|, \quad (5.29)$$

$$\underline{\gamma}^* = \arg \min_{\gamma \in \mathbb{R}^M} \max_{x \in \Omega_{nt}} |g(u(\underline{x}; \mu); \mu) - \sum_{j=1}^M \gamma_j \xi_j(\underline{x})|, \quad (5.30)$$

$$g_M^*(u(\underline{x}; \mu); \mu) = \sum_{j=1}^M \gamma_j^* \xi_j(\underline{x}). \quad (5.31)$$

When we solve (5.27) we choose candidates for μ_M^g from a sufficiently fine sample (mesh) $S_{\mathcal{D}}$. As we discussed earlier in Sections 2.3.4 and 4.1.2, depending on P , the number of parameters which define our problem, it might be computationally unrealistic to actually provide such a mesh. In this case we would have to replace $S_{\mathcal{D}}$ with $S_{M_{pool}}$, that is, a pool of M_{pool} randomly chosen points. Another complicating feature of this computational algorithm is that we actually have to solve the original problem (5.1) for *all* sample points in $S_{\mathcal{D}}$ (or $S_{M_{pool}}$) which is expensive; note that we no longer have a choice similar to the one described in Section 2.3.4 where we could choose between the expensive "truth" (2.42) and the cheap reduced basis (2.43) approaches. Thus, for the selection of S_M^g we are forced to use the "truth" approach.

We repeat steps (5.27) and (5.29) until $\epsilon_{M-1}^*(\mu_M^g)$ gets sufficiently small and set M_{max} equal to the current M . We next define the sample $S_{M_{max}}^g = \{\mu_1^g, \dots, \mu_{M_{max}}^g\}$ and introduce the function space $V_{M_{max}}^g = \text{span}\{\xi_M(\underline{x}) = g(u(\underline{x}; \mu_M^g), \mu_M^g), M = 1, \dots, M_{max}\}$. We note that $\epsilon_1^*(\mu_2^g) \geq \epsilon_2^*(\mu_3^g) \geq \dots \geq \epsilon_{M_{max}-1}^*(\mu_{M_{max}}^g) \geq \epsilon_0$ since $V_1^g \subset V_2^g \subset \dots \subset V_{M_{max}}^g$ from (5.29).

We now proceed by introducing the following Lemma.

Lemma 5.1. The dimension of space V_M^g is equal to M , $M = 1, \dots, M_{max}$.

The proof is analogous to the proof of Lemma 4.1.

5.2.2 "Magic Points" Selection

As in the case of the locally non-affine problems, once we are done with the selection of the sample $S_{M_{max}}^g$ we run another recursive procedure. We now introduce a sequence of "residual" functions $r_M(\underline{x})$, $M = 1, \dots, M_{max}$ and a sequence of physical points \underline{t}_M , $M = 1, \dots, M_{max}$ that are constructed according to the following procedure

$$r_1(\underline{x}) = \xi_1(\underline{x}), \tag{5.32}$$

$$\underline{t}_1 = \arg \max_{\underline{x} \in \Omega} |r_1(\underline{x})|, \tag{5.33}$$

$$\underline{B}_{11} = \xi_1(\underline{t}_1); \tag{5.34}$$

then, for $M = 2, \dots, M_{max}$

$$\underline{B}^{M-1} \sigma^{M-1} = [\xi_M(\underline{t}_1), \dots, \xi_M(\underline{t}_{M-1})]^T, \quad (5.35)$$

$$r_M(\underline{x}) = \xi_M(\underline{x}) - \sum_{j=1}^{M-1} \sigma_j^{M-1} \xi_j(\underline{x}), \quad (5.36)$$

$$\underline{t}_M = \arg \max_{\underline{x} \in \Omega_{nl}} |r_M(\underline{x})|, \quad (5.37)$$

$$B_{ij}^M = \xi_j(\underline{t}_i), 1 \leq i, j \leq M. \quad (5.38)$$

We refer to the sequence $\underline{t}_M, M = 1, \dots, M_{max}$ as to the "magic points". We now formulate the Lemma which ensures that we indeed can generate such a sequence.

Lemma 5.2. The rank of \underline{B}^M is equal to $M, M = 1, \dots, M_{max}$.

The proof is similar to the proof of Lemma 4.2.

When we solve the original problem for each Newton iteration we have to evaluate $g(\bar{u}(\mu); \mu)$ where $\bar{u}(\mu)$ is function obtained at the k^{th} step of the Newton method. At the moment it is not important for us how this $\bar{u}(\mu)$ is obtained. For example, $\bar{u}(\mu)$ can be a regular member of Y as in (5.1) and thus there would be \mathcal{N} coefficients which define $\bar{u}(\mu)$ or, alternatively, $\bar{u}(\mu)$ can be a member of W_N , hence we would only need N coefficients to describe $\bar{u}(\mu)$. For every $\bar{u} \in Y$ we are now ready to introduce the affine-like approximation $g_M^{\bar{u}}(\underline{x}; \mu); \mu$ for $g(\underline{x}; \mu) \forall M \leq M_{max}$ as

$$g_M^{\bar{u}}(\underline{x}; \mu) = \sum_{m=1}^M \alpha_m(\mu) \xi_m(\underline{x}), \quad \underline{\alpha}(\mu) \in \mathbb{R}^M, \quad (5.39)$$

$$\underline{\alpha}(\mu) = \underline{B}_M^{-1} [g(\bar{u}(\underline{t}_1; \mu); \mu), \dots, g(\bar{u}(\underline{t}_M; \mu); \mu)]^T. \quad (5.40)$$

From (5.40) we see that

$$g_M^{\bar{u}}(\underline{t}_i; \mu) = g(\bar{u}(\underline{t}_i; \mu); \mu), \quad i = 1, \dots, M. \quad (5.41)$$

We then introduce the interpolation error $\epsilon_M(\mu)$ as

$$\epsilon_M(\mu) = \|g(\bar{u}(\underline{x}; \mu); \mu) - g_M^{\bar{u}}(\underline{x}; \mu)\|_{L^\infty(\Omega_{nl})}; \quad (5.42)$$

and define "Lebesgue" constant Λ_M as

$$\Lambda_M = \sup_{x \in \Omega_{nt}} \sum_{k=1}^M |V_k^M(\underline{x})|, \quad (5.43)$$

where

$$\sum_{j=1}^M \underline{B}_{ji}^M V_j(\underline{x}) = \xi_i(\underline{x}), i = 1, \dots, M. \quad (5.44)$$

We are now ready to state the next Lemma which provides a "stability" link between $\epsilon_M(\mu)$ and $\epsilon_M^*(\mu)$.

Lemma 5.3. $\epsilon_M(\mu) \leq \epsilon_M^*(\mu)(1 + \Lambda_M), \forall \mu \in \mathcal{D}, M \leq M_{max}.$

The proof repeats the proof of Lemma 4.3. This Lemma ensures that $g_M^{\bar{u}}$ and g are indeed close provided that (i) M is sufficiently big – and hence ϵ_M^* is small – and at the same time (ii) Λ_M is small. In [5] an upper bound for Λ_M is described; this upper bound appears to be very pessimistic (exponential in M), however in usual practice Λ_M remains relatively small ($O(10)$). We provide concrete numerical results for Λ_M in Table 5.1 when we discuss the 2D model problem. We finally conclude this Section with the following Corollary

Corollary 5.1. $\epsilon_{M-1}^*(\mu_M^g) \leq |r_M(\underline{t}_M)| \leq (1 + \Lambda_M)\epsilon_{M-1}^*(\mu_M^g).$

The proof directly follows from (5.29), (5.36), and Lemma 5.3.

5.3 Application of Minimax Coefficient Approximation Method to Locally Non-linear Problems

In this Section we describe how we can use the results of Section 5.2 to construct the minimax coefficient approximation reduced basis method.

5.3.1 Reduced Basis Approximation

As described in Section 5.2.1, we start by selecting the sample

$$S_M^g = \{\mu_1^g, \dots, \mu_M^g\} \quad (5.45)$$

and constructing the basis for the function space V_M^g

$$V_M^g = \text{span}\{\xi_m(\underline{x}) = g(u(\underline{x}; \mu_m^g), \mu_m^g), m = 1, \dots, M\}. \quad (5.46)$$

We then build the matrix \underline{B}^M as given in (5.38). We next introduce the sample in the parameter domain \mathcal{D}

$$S_N = \{\mu_1, \dots, \mu_N\}, \quad (5.47)$$

where $\mu_i \in \mathcal{D}, i = 1, \dots, N$. We define our reduced-basis approximation space W_N as

$$W_N = \text{span}\{\zeta_i \equiv u(\mu_i), i = 1, \dots, N\}, \quad (5.48)$$

where $u(\mu_i) \in Y$ is the solution to (5.1) for $\mu = \mu_i$. The selection of S_N could be accomplished using the ideas presented in Section 2.3.4.

For any $\mu \in \mathcal{D}$ we look for $u_{NM}(\mu) \in W_N$ which satisfies

$$a^A(u_{NM}(\mu), v; \mu) + \int_{\Omega_{nl}} g_M^{u_{NM}}(\mu)v = f(v), \quad \forall v \in W_N, \quad (5.49)$$

$$g_M^{u_{NM}}(\mu) = \sum_{m=1}^M \alpha_m(\mu)\xi_m(\underline{x}), \quad (5.50)$$

$$\underline{\alpha}(\mu) = B_M^{-1}[g(u_{NM}(\underline{t}_1; \mu); \mu), \dots, g(u_{NM}(\underline{t}_M; \mu); \mu)]^T. \quad (5.51)$$

In (5.49) we have replaced the non-linearity in g with an affine-like approximation $g_M^{u_{NM}}$ which provides an efficient way to perform Newton iterations because - as we are going to show later in this Chapter - the Jacobian of the system of non-linear equations (5.49) essentially has an affine structure.

5.3.2 A Posteriori Error Estimation Based on Positivity of Non-linear Contribution

We start by introducing a bound conditioner [18, 39, 52, 57] - a symmetric, continuous, and coercive bilinear form $\hat{a}(v, v; \mu)$ such that $\forall \mu \in \mathcal{D}, \forall v \in Y$

$$\alpha(\mu)(\|v\|_Y)^2 \leq \rho_{min}(\mu)\hat{a}(v, v; \mu) \leq a^A(v, v; \mu) \leq \rho_{max}(\mu)\hat{a}(v, v; \mu) \leq \gamma(\mu)(\|v\|_Y)^2, \quad (5.52)$$

where

$$1 \leq \rho_{min}(\mu), \quad \rho_{max}(\mu) \leq \rho, \quad \alpha_0 \leq \alpha(\mu), \quad \gamma(\mu) \leq \gamma_0, \quad (5.53)$$

for some (preferably small) constant $\rho \in \mathbb{R}$. The more detailed of description of different types of bound conditioners is contained in [52, 57].

We then define the error function $e_{NM}(\mu) = u(\mu) - u_{NM}(\mu)$. Using (5.1), (5.2), (5.6), and (5.49) we arrive at the following error equation which holds $\forall v \in Y$

$$\begin{aligned} a^A(e_{NM}(\mu), v; \mu) + \int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_{NM}(\mu); \mu))v = \\ f(v) - a^A(u_{NM}(\mu), v; \mu) - \int_{\Omega_{nl}} g_M^{u_{NM}}(\mu)v + \int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))v. \end{aligned} \quad (5.54)$$

We next introduce $R_1(v; \mu), R_2(v; \mu) \in Y'$ such that

$$R_1(v; \mu) = f(v) - a^A(u_{NM}(\mu), v; \mu) - \int_{\Omega_{nl}} g_M^{u_{NM}}(\mu)v, \quad (5.55)$$

$$R_2(v; \mu) = \int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))v. \quad (5.56)$$

We now consider two subproblems

$$\hat{a}(\hat{e}_{NM,1}(\mu), v; \mu) = R_1(v; \mu), \quad (5.57)$$

$$\hat{a}(\hat{e}_{NM,2}(\mu), v; \mu) = R_2(v; \mu). \quad (5.58)$$

We next introduce $\tilde{\Delta}_{NM,1}(\mu)$

$$\tilde{\Delta}_{NM,1}(\mu) = \hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu). \quad (5.59)$$

We subsequently invoke (5.56), (5.58) and Schwartz' inequality to arrive at

$$\hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu) = \int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu)) \hat{e}_{NM,2}(\mu) \leq$$

$$\begin{aligned} & \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}} \left(\int_{\Omega_{nl}} (\hat{e}_{NM,2}(\mu))^2 \right)^{\frac{1}{2}} \leq \\ & \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}} \sqrt{\frac{1}{\alpha(\mu)} \sup_{v \in Y} \frac{\|v\|_{L^2(\Omega_{nl})}}{\|v\|_Y} (\hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu))^{\frac{1}{2}}}; \end{aligned} \quad (5.60)$$

from (5.60) we readily introduce $\tilde{\Delta}_{NM,2}(\mu) \geq \hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu)$ as follows

$$\tilde{\Delta}_{NM,2}(\mu) = \frac{1}{\alpha(\mu)} \int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \left(\sup_{v \in Y} \frac{\|v\|_{L^2(\Omega_{nl})}}{\|v\|_Y} \right)^2. \quad (5.61)$$

We then plug $v = e_{NM}(\mu)$ into (5.54) to obtain

$$\begin{aligned} a^A(e_{NM}(\mu), e_{NM}(\mu); \mu) + \int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_{NM}(\mu); \mu)) e_{NM}(\mu) &= R_1(e_{NM}(\mu); \mu) + R_2(e_{NM}(\mu); \mu) = \\ & \hat{a}(\hat{e}_{NM,1}(\mu), e_{NM}(\mu); \mu) + \hat{a}(\hat{e}_{NM,2}(\mu), e_{NM}(\mu); \mu) \leq \\ & \left(\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)^{\frac{1}{2}} + \hat{a}(\hat{e}_{NM,2}(\mu), \hat{e}_{NM,2}(\mu); \mu)^{\frac{1}{2}} \right) \hat{a}(e_{NM}(\mu), e_{NM}(\mu); \mu)^{\frac{1}{2}} \leq \\ & ((\tilde{\Delta}_{NM,1})^{\frac{1}{2}} + (\tilde{\Delta}_{NM,2})^{\frac{1}{2}}) \hat{a}(e_{NM}(\mu), e_{NM}(\mu); \mu)^{\frac{1}{2}}; \end{aligned} \quad (5.62)$$

we now explain how we arrived at the final result of (5.62): in the first step we invoked (5.54), (5.55), and (5.56), in the second – (5.57) and (5.58), in the the third – Cauchy-Schwartz' inequality, and, finally, for the last step – the definitions of $\tilde{\Delta}_{NM,1}$ and $\tilde{\Delta}_{NM,2}$. Since g is monotonic in its first argument we can write that

$$\int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_{NM}(\mu); \mu))(u(\mu) - u_{NM}(\mu)) \geq 0. \quad (5.63)$$

Using (5.52) and (5.63) we note that

$$\alpha(\|e_{NM}(v)\|_Y)^2 \leq a^A(e_{NM}(\mu), e_{NM}(\mu); \mu) \leq \left((\tilde{\Delta}_{NM,1})^{\frac{1}{2}} + (\tilde{\Delta}_{NM,2})^{\frac{1}{2}} \right)^2. \quad (5.64)$$

We finally introduce $\Delta_{NM}(\mu)$ such that

$$\|e_{NM}(\mu)\|_Y \leq \Delta_{NM}(\mu) = \sqrt{\frac{\tilde{\Delta}_{NM,1}}{\alpha(\mu)}} + \sqrt{\frac{\tilde{\Delta}_{NM,2}}{\alpha(\mu)}} =$$

$$\sqrt{\frac{\hat{\alpha}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)}{\alpha(\mu)}} + \frac{1}{\alpha(\mu)} \sup_{v \in Y} \frac{\|v\|_{L^2(\Omega_{nl})}}{\|v\|_Y} \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}}; \quad (5.65)$$

if we define $\Phi(\mu) = \frac{1}{\alpha(\mu)} \sup_{v \in Y} \frac{\|v\|_{L^2(\Omega_{nl})}}{\|v\|_Y}$ we can rewrite (5.65) as

$$\Delta_{NM}(\mu) = \sqrt{\frac{\hat{\alpha}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)}{\alpha(\mu)}} + \Phi(\mu) \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}}. \quad (5.66)$$

As (5.65) indicates, we have two terms which contribute to the error bound $\Delta_{NM}(\mu)$. Let us denote these terms as $\Delta_{NM,1}$ and $\Delta_{NM,2}$ such that

$$\Delta_{NM} = \Delta_{NM,1} + \Delta_{NM,2}, \quad (5.67)$$

$$\Delta_{NM,1} = \sqrt{\frac{\hat{\alpha}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)}{\alpha(\mu)}}, \quad (5.68)$$

$$\Delta_{NM,2} = \frac{1}{\alpha(\mu)} \sup_{v \in Y} \frac{\|v\|_{L^2(\Omega_{nl})}}{\|v\|_Y} \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}}. \quad (5.69)$$

We are going to discuss the importance and relative contribution of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ in Section 5.4 when we consider application of the minimax coefficient approximation reduced basis method to a concrete model problem.

As usual, we introduce the effectivity $\eta_{NM}(\mu)$ which serves as a measure of the sharpness of the error bound $\Delta_{NM}(\mu)$

$$1 \leq \eta_{NM}(\mu) = \frac{\Delta_{NM}(\mu)}{\|e_{NM}(\mu)\|_Y}. \quad (5.70)$$

5.3.3 Off-line/On-line Procedure

Reduced Basis Approximation

The system of equations (5.49) is non-linear and in order to solve it we employ Newton method. Since $u_{NM}(\mu) \in W_N$ we can write that

$$u_{NM}(\mu) = \sum_{n=1}^N \beta_n(\mu) \zeta_n. \quad (5.71)$$

Plugging this expression into (5.49) we obtain

$$\sum_{q=1}^Q \sum_{j=1}^N A_{Nij}^q \beta_j(\mu) + \sum_{j=1}^M C_{NMij} \alpha_j(\mu) = F_{N_i}, \quad i = 1, \dots, N, \quad (5.72)$$

where

$$A_{Nij}^q = a^q(\zeta_i, \zeta_j), \quad q = 1, \dots, Q, \quad i, j = 1, \dots, N, \quad (5.73)$$

$$C_{NMij} = \int_{\Omega_{nl}} \zeta_i \xi_j, \quad i = 1, \dots, N, \quad j = 1, \dots, M, \quad (5.74)$$

$$F_{N_i} = f(\zeta_i), \quad i = 1, \dots, N, \quad (5.75)$$

and $\underline{\alpha}(\mu)$ is obtained as follows

$$\sum_{j=1}^M \underline{B}_{ij}^M \alpha_j(\mu) = g\left(\sum_{j=1}^N T_{ij} \beta_j(\mu)\right), \quad i = 1, \dots, M, \quad (5.76)$$

where, in turn,

$$T_{ij} = \zeta_j(\underline{t}_i), \quad i = 1, \dots, M, \quad j = 1, \dots, N. \quad (5.77)$$

Using (5.76) and (5.77), we now write (5.72) as

$$\sum_{q=1}^Q \sum_{j=1}^N A_{Nij}^q \beta_j(\mu) + \sum_{j=1}^M \underline{U}_{NMij} g\left(\sum_{n=1}^N T_{jn} \beta_n(\mu)\right) = F_{N_i}, \quad i = 1, \dots, N, \quad (5.78)$$

where $\underline{U}_{NM} \in \mathbb{R}^{N \times M}$

$$\underline{U}_{NM} = \underline{C}_{NM}(\underline{B}_M)^{-1}. \quad (5.79)$$

Based on (5.49) we thus can write a linear system of equations which we need to solve for each Newton iteration in terms of $\underline{\beta}$ and $\delta\underline{\beta}$ as follows

$$\left(\sum_{q=1}^Q \sum_{j=1}^N A_{N_{ij}}^q + \sum_{k=1}^M \sum_{j=1}^N U_{NM_{ik}} g' \left(\sum_{n=1}^N T_{kn} \beta_n(\mu) \right) T_{kj} \right) \delta\beta_j = F_i - \sum_{j=1}^N A_{N_{ij}} \beta_j - \sum_{k=1}^M U_{NM_{ik}} g \left(\sum_{n=1}^N T_{kn} \beta_n(\mu) \right). \quad (5.80)$$

As mentioned in Section 5.1.1, we keep making Newton steps till both $\|\delta\underline{\beta}\|$ and $\|\underline{F}\|$ become small enough.

Error Bound

We rewrite (5.55), the expression for $R_1(v; \mu)$, as

$$R_1(v; \mu) = f(v) - \sum_{q=1}^Q \sum_{j=1}^N \Theta_q(\mu) \beta_j(\mu) a^q(\zeta_j, v) - \sum_{m=1}^M \alpha_m(\mu) \int_{\Omega_{nl}} \xi_m v. \quad (5.81)$$

Renumbering (5.81) we obtain

$$R_1(v; \mu) = \sum_{i=1}^{1+QN+M} \tau_i(\mu) \Upsilon_i(v), \quad (5.82)$$

where for

$$\tau_1(\mu) = 1, \Upsilon_1(v) = f(v); \quad (5.83)$$

for $i = 2, \dots, 1 + QN$:

$$i = 1 + (q-1)j, \tau_i(\mu) = -\Theta_q(\mu) \beta_j(\mu), \Upsilon_i(v) = a^q(\zeta_j, v), \quad q = 1, \dots, Q, j = 1, \dots, N; \quad (5.84)$$

for $i = 2 + QN, \dots, 1 + QN + M$:

$$i = 2 + QN + l, \tau_i = -\alpha_l(\mu), \Upsilon_l(v) = \int_{\Omega_{nl}} \xi_l v, \quad l = 1, \dots, M. \quad (5.85)$$

Next, invoking (5.59), (5.83), (5.84), and (5.85), we can rewrite the expression for $\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu)$ as

$$\hat{a}(\hat{e}_{NM,1}(\mu), \hat{e}_{NM,1}(\mu); \mu) = \sum_{i=1}^{1+QN+M} \sum_{i'=1}^{1+QN+M} \tau_i(\mu) \tau_{i'}(\mu) \Lambda_{ii'}, \quad (5.86)$$

where

$$\Lambda_{ii'} = \hat{a}(z_i, z_{i'}; \mu), \quad i, i' = 1, \dots, 1 + QN + M, \quad (5.87)$$

$$\hat{a}(z_i, v; \mu) = \Upsilon_i(v), \quad \forall v \in Y; \quad (5.88)$$

we readily construct the error bound $\Delta_{NM}(\mu) \geq \|e_{NM}(\mu)\|_Y$ from (5.65) as

$$\Delta_{NM}(\mu) = \frac{1}{\sqrt{\alpha(\mu)}} \sqrt{\sum_{i=1}^{1+QN+M} \sum_{i'=1}^{1+QN+M} \tau_i(\mu) \tau_{i'}(\mu) \Lambda_{ii'} + \Phi(\mu) \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}}}. \quad (5.89)$$

Turning to the decomposition of the error bound $\Delta_{NM}(\mu)$ we introduced in (5.67) we rewrite the expressions for $\Delta_{NM,1}$ and $\Delta_{NM,2}$ as

$$\Delta_{NM,1} = \frac{1}{\sqrt{\alpha(\mu)}} \sqrt{\sum_{i=1}^{1+QN+M} \sum_{i'=1}^{1+QN+M} \tau_i(\mu) \tau_{i'}(\mu) \Lambda_{ii'}}, \quad (5.90)$$

$$\Delta_{NM,2} = \Phi(\mu) \left(\int_{\Omega_{nl}} (g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu))^2 \right)^{\frac{1}{2}}. \quad (5.91)$$

In order to calculate $\Delta_{NM,2}$ we need to construct and to evaluate $g_M^{u_{NM}}(\mu) - g(u_{NM}(\mu); \mu)$ at all n_{nl} nodes in Ω_{nl} which requires n_{nl} operations. This is why the assumption of the local nature of the non-linearity of the original problem is crucial for providing rigorous and at the same time fast error estimation.

Numerical Algorithm

Off-line

1. Generate S_M^g as described in Section 5.2.1 and construct the basis for V_M^g ;
2. Generate \underline{B}_M^{-1} as shown in Section 5.2.2;
3. Generate S_N as described in Section 2.3.4 and construct the basis for W_N ;

4. Precompute \underline{A}_N^q , \underline{F}_N , \underline{T}_{NM} , and \underline{U}_{NM} , as in (5.73), (5.75), (5.77), and (5.79), respectively;
5. Precompute $\Lambda_{i'}$ as in (5.87), $i, i' = 1, \dots, 1 + QN + M$.

On-line

1. Calculate $u_{NM}(\mu)$ using Newton method as described in Section 5.3.3. Cost: $K^{iter}O((Q + M)N^2) + O(N^3)$, K^{iter} is the number of Newton iterations required to obtain $u_{NM}(\mu)$;
2. Construct the error bound $\Delta_{NM}(\mu)$ from (5.66). Cost: $O((QN + M)^2) + O(Mn_{nl})$.

Total on-line cost: $K^{iter}O((Q + M)N^2) + O(N^3) + O((QN + M)^2) + O(Mn_{nl})$.

The operation count of 1. results from the fact that we need to assemble the Jacobian and the right hand side of (5.80) which requires $O((Q + M)N^2)$ operations and then solve (5.80) for an updated Newton direction which requires additional $O(N^3)$ operations. The operation count of 2. is obtained as a sum of operation counts to evaluate $\Delta_{NM,1}(\mu)$ and $\Delta_{NM,2}(\mu)$ as given in (5.90) and (5.91). Clearly, evaluation of $\Delta_{NM,1}(\mu)$ requires $O((QN + M)^2)$ operations whereas evaluation of $\Delta_{NM,2}(\mu)$ – the main part of which is evaluation of the integral over Ω_{nl} – takes $O(Mn_{nl})$ operations.

5.4 Model Problem: 2D Heat Conduction Example with Radiation Boundary Conditions

In order to demonstrate the implementation of our minimax approximation coefficient method (MCAM) we choose yet another variation of the two-dimensional thermal fin problem. We consider a thermal fin with fixed geometry that is characterized by 6 parameters which constitute a 6-dimensional vector μ :

$$\mu = \left(\underbrace{k_1}_{\mu_1}, \underbrace{k_2}_{\mu_2}, \underbrace{k_3}_{\mu_3}, \underbrace{k_4}_{\mu_4}, \underbrace{Bi}_{\mu_5}, \underbrace{\sigma}_{\mu_6} \right)$$

where $k_i, i = 1, \dots, 4$ is the conductivity of the i^{th} sub-fin (Ω_i), Bi is the Biot number, a non-dimensional heat transfer coefficient reflecting convective transport to the air at the fin surfaces, and σ is the emissivity of the gray surface Ω_{nl} reflecting the radiative heat transfer to the environment through Ω_{nl} corresponding to Stefan-Boltzmann law. The boundary segment Ω_{nl} is pictured in Figure 5-1 by a thick solid line. We assume the ambient temperature of the environment to be 0 and we neglect the effect of self-irradiation.

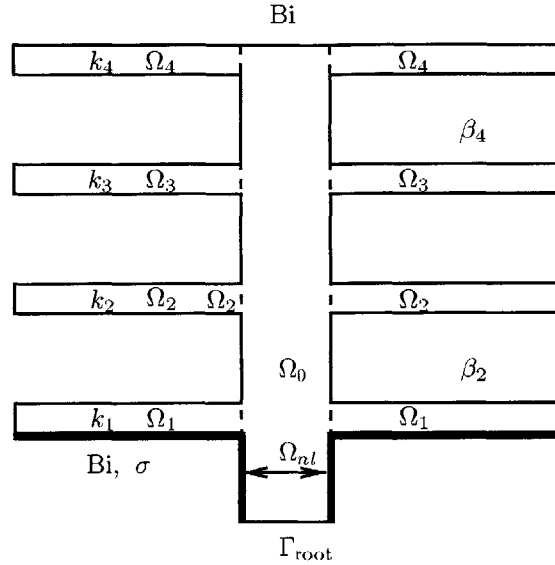


Figure 5-1: 2D locally non-linear model problem: a thermal fin with convective and radiation boundary conditions.

The geometry of the fin is constant; the vertical distance between the sub-fins is fixed at .75, the height and the length of the sub-fins are equal to 0.25 and 2, respectively (all distances are measured relative to the post width). For our parameter domain we choose $\mathcal{D} = [0.1, 10.0]^4 \times [0.01, 1.0] \times [0.0, 1.0]$, that is, $0.1 \leq k_i \leq 10.0$, $i = 1, \dots, 4$ for the conductivities, $0.01 \leq \text{Bi} \leq 1.0$, for the Biot number, and $0.01 \leq \sigma \leq 1.0$, for the emissivity of Ω_{nl} . We only focus on the norm of the error $\|e_{NM}(\mu)\|_Y = \|u(\mu) - u_{NM}(\mu)\|_Y$ – where $u_{NM}(\mu)$ is our reduced basis approximation obtained using the MCAM – since we can express the approximation for the linear outputs through $\|e_{NM}(\mu)\|_Y$ as it was shown in Section 2.3.3.

5.4.1 Governing Partial Differential Equations

We next provide the rigorous mathematical statement of the problem based on the underlying partial differential equations.

Strong Form

The temperature distribution $u(\mu)$, is obtained by solution of the following elliptic partial differential equation:

$$-k_i \nabla^2 u_i(\mu) = 0 \text{ in } \Omega_i, \quad i = 0, \dots, 4 \quad (5.92)$$

where ∇^2 is the Laplacian operator, and $u_i(\mu) \equiv u(\mu)|_{\Omega_i}$ refers to the restriction of $u(\mu)$ to Ω_i . Here Ω_i is the region of the fin with conductivity k_i , $i = 0, \dots, 5$: Ω_0 is thus the central post, and Ω_i , $i = 1, \dots, 4$, corresponds to the four sub-fins. We must also ensure the continuity of temperature and heat flux at the conductivity-discontinuity interfaces $\Gamma_i \equiv \partial\Omega_0 \cap \partial\Omega_i$, $i = 1, \dots, 4$, where $\partial\Omega_i$ denotes the boundary of Ω_i :

$$u_0(\mu) = u_i(\mu), \quad (5.93)$$

$$-(\nabla u_0(\mu) \cdot \hat{n}_i) = -k_i(\nabla u_i(\mu) \cdot \hat{n}_i) \text{ on } \Gamma_i, \quad i = 1, \dots, 5; \quad (5.94)$$

here \hat{n}_i is the outward normal on $\partial\Omega_i$.

We next introduce the external boundary conditions. We impose Neumann boundary condition on the fin root:

$$-(\nabla u_0(\mu) \cdot \hat{n}_o) = -1 \text{ on } \Gamma_{\text{root}}, \quad (5.95)$$

which models the heat source; and the Robin boundary condition on the whole external boundary of the fin with the exception of its root and the radiative surface Ω_{nl} :

$$-k_i(\nabla u_i(\mu) \cdot \hat{n}_i) = \text{Bi } u_i(\mu) \text{ on } \Gamma_{\text{ext } i} \setminus (\Gamma_{\text{root}} \cup \Omega_{nl}), \quad i = 0, \dots, 5, \quad (5.96)$$

which models the convective heat losses; finally we impose both Robin and the non-linear power law boundary condition on Ω_{nl}

$$-k_i(\nabla u_i(\mu) \cdot \hat{n}_i) = \text{Bi } u_i(\mu) + \sigma(u_i(\mu))^4 \text{ on } \Omega_{nl}, \quad i = 0, 1. \quad (5.97)$$

Weak Form

We now produce the weak form of our problem by multiplying equation (5.92) by an arbitrary function $v \in Y$ and applying the divergence theorem and boundary conditions (5.95), (5.96), and (5.97)

$$\underbrace{\sum_{i=0}^4 \int_{\Omega_i} k_i \nabla u(\mu) \nabla v + \text{Bi} \int_{\Gamma_{\text{ext}} \setminus \Gamma_{\text{root}}} u(\mu) v}_{a^A(u(\mu); v; \mu)} + \underbrace{\sigma \int_{\Omega_{nl}} u^4(\mu) v}_{a^{NL}(g(u(\mu)); \mu), v)} - \underbrace{\int_{\Gamma_{\text{root}}} v}_{f(v)} = 0, \quad \forall v \in Y. \quad (5.98)$$

In the absence of the geometric transformations the equation (5.98) defines the model problem abstract formulation which is consistent with (5.2). The model problem is well-posed, i.e. the solution $u(\mu)$ depends continuously on μ , also for our choice of $\mathcal{D} u(\mu)$ remains non-negative. We here provide an outline of the proof of these facts, more details could be found in [26].

We start by formulating the following Theorem.

Theorem 5.1. Let $\Omega \in \mathbb{R}^2$ be a bounded domain with a Lipschitz boundary Γ . Then for every real number $p \geq 1$ there exists a single continuous linear mapping $Z : H^1(\Omega) \rightarrow L^p(\Gamma)$ which satisfies $Zv = v$ on $\Gamma \forall v \in C^\infty(\overline{\Omega})$.

Theorem 5.1 guarantees that $a^{NL}(w, v; \mu)$ is well-defined for arbitrary $w, v \in H^1(\Omega)$ and hence the weak formulation makes sense. We next define a set of admissible functions U as

$$U = \{v \in H^1(\Omega) | v \geq 0\}, \quad (5.99)$$

and a function $\Pi(v) : H^1(\Omega) \rightarrow \mathbb{R}$ such that

$$\Pi_\mu(v) = \frac{1}{2}a^A(v, v; \mu) + \frac{1}{5}a^{NL}(v, v; \mu) - f(v); \quad (5.100)$$

By Theorem 5.1 $\Pi_\mu(v)$ is well-defined and continuous in $H^1(\Omega)$, moreover $\Pi_\mu(v)$ is also G -differentiable and strictly convex in U ; these conditions ensure [46] that there exists precisely one element $\bar{u}(\mu) \in U$ in which $\Pi_\mu(v)$ attains its minimum over the set U . This element fulfills

$$\mathbf{D}\Pi_\mu(\bar{u}(\mu), v) = \mathbf{D}\Pi_\mu(w, v) = a^A(w, v; \mu) + a^{NL}(w, v; \mu) - f(v) = 0, \quad (5.101)$$

which concludes the proof of the existence and uniqueness of the weak solution $u(\mu) = \bar{u}(\mu)$ in the sense of formulation (5.1). It is next standard to show that the weak solution depends continuously on μ .

It is important to emphasize that the the required monotonicity of $g(u) = u^4$ is only attained for positive values of $u(\mu)$, $u_{NM}(\mu)$, this why the positivity is essential for the construction of our algorithm. Since we cannot a priori prove the positivity $u_{NM}(\mu)$ we need to check that the reduced basis approximation stays positive in Ω_{nl} in all points which incurs extra n_{nl} operations. Since as it was shown in Section 5.3.3 the operation count for $\Delta_{NM}(\mu)$ has a $O(Mn_{nl})$ term in it, the additional n_{nl} operations are not going to change the operation count significantly. If for some reason – though we never encountered this in our numerical tests – $u_{NM}(\mu)$ takes a negative value

that means that either M or N is too small, since as we increase N and M we expect $u_{NM}(\mu)$ to more and more close to $u(\mu)$ which is in turn positive everywhere in Ω . We provide more analysis on the issue of positivity of $u_{NM}(\mu)$ and the related implications later in this Chapter when we discuss the error bound decomposition.

5.4.2 Numerical Results

To obtain the “truth” solution we consider a finite element mesh that consists of 21802 points.

Effect of Radiation

We now demonstrate that radiation plays an important role in our model problem. In Figure 5-2 we provide the plots of the field solution $u(\mu)$ for 4 different values of σ for the same values of $k_1, k_2, k_3, k_4, \text{Bi}$. We note that the field distribution depends on σ in a meaningful way.

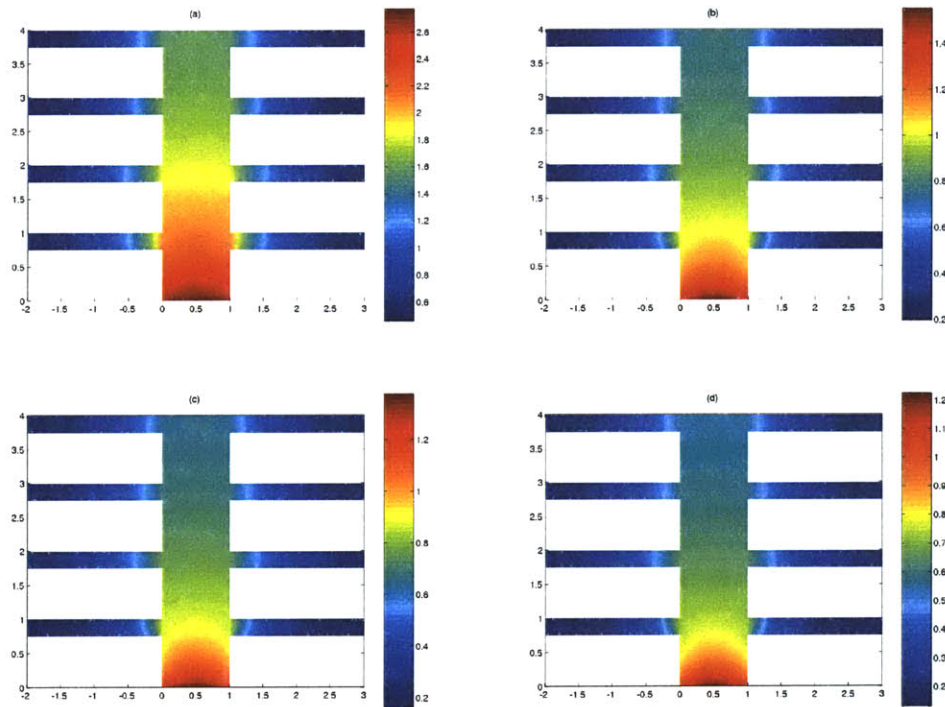


Figure 5-2: 2D locally non-linear model problem: the field solution $u(\mu)$ for

- (a) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 0.01\}$,
- (b) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 0.25\}$,
- (c) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 0.50\}$,
- (d) $\mu = \{0.10, 0.10, 0.10, 0.10, 0.01, 1.00\}$.

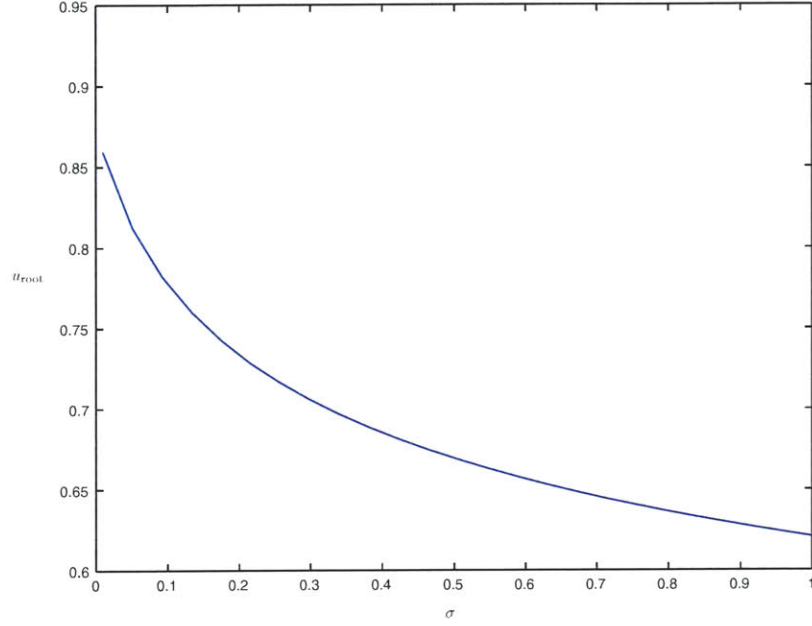


Figure 5-3: 2D locally non-linear model problem: variation of u_{root} as a function of σ , $k_1 = 0.50$, $k_2 = 0.50$, $k_3 = 0.50$, $k_4 = 0.50$, $\text{Bi} = 0.50$.

We continue the illustration of the importance of the non-linear parameter by presenting the plot of u_{root} , the mean temperature along Γ_{root} , as a function of σ while having all other parameters fixed. As we observe from Figure 5-3, just by varying σ we achieve significant output variation.

Approximation of Non-linearity

We first present the numerical results related to the approximation of the non-linear contribution in $g(u(\underline{x}; \mu); \mu)$. As described in Section 5.2.1, we begin from sequentially generating the sample S_M^g . We then generate the space of functions $V_M^g = \text{span}\{\xi_m(\underline{x}) = g(u(\underline{x}; \mu_m^g), \mu_m^g), m = 1, \dots, M\}$ over which we actually perform the interpolation as given in (5.51). We choose the members of S_M^g from a bigger sample set $S_{M_{\text{pool}}}$ of randomly generated parameter points, where M_{pool} , the cardinality of $S_{M_{\text{pool}}}$ is equal to 10000. As discussed earlier in Section 5.2.1, even in the case of 6 parameters, it becomes too computationally expensive to perform a search over a fine mesh in \mathcal{D} .

In Figure 5-4 we provide the dependencies of $\epsilon_M^*(\mu_{M+1}^g)$ (in blue) and $r_{M+1}(\underline{t}_{M+1})$ (in red) as functions of M . We observe that – as predicted by Corollary 5.1 – $\epsilon_M^*(\mu_{M+1}^g) \leq r_{M+1}(\underline{t}_{M+1})$.

In Table 5.1 we provide the dependencies of Λ_M as defined in (5.43) and the condition number of the matrix \underline{B}^M as functions of M . As we see, $\text{cond}(\underline{B}^M)$ grows significantly with M . At the same time Λ_M remains quite small and consistent with Corollary 5.1 $\epsilon_M^*(\mu_{M+1}^g)$ and $r_{M+1}(\underline{t}_{M+1})$ –

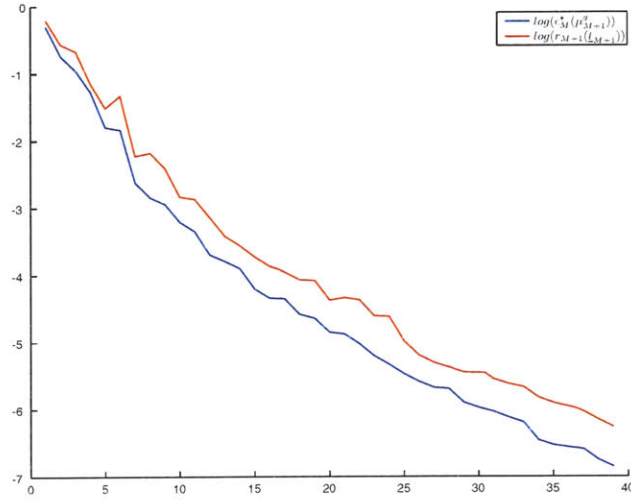


Figure 5-4: Minimax coefficient approximation method, 2D locally non-linear problem: non-linear contribution approximation.

as observed from Figure 5-4 – remain close to each other. Since our reduced basis method requires inversion of \underline{B}^M – see Section 5.3.3, (5.79) – we might encounter numerical difficulties due to the high condition number of \underline{B}^M as M becomes larger. In any event we would not want M to be too big since it might compromise the numerical efficiency of our reduced basis method because of the operation count for the on-line stage presented in Section 5.3.3. The alternative approach to construct \underline{B}^M based on the orthogonalization procedure is described in [5]; there also exists an alternative technique for magic points selection, [28], [10], however, for the purposes of this thesis there is no need to employ more complicated procedures.

Convergence

As described in Section 5.3.2, we introduce a bound conditioner $\hat{a}(\cdot, \cdot; u)$ that is equal to

$$\hat{a}(w, v) = \sum_{q=1}^Q \bar{\Theta}_q a^q(w, v) = \int_{\Omega_0} \nabla w \nabla v + 0.1 \sum_{i=1}^4 \int_{\Omega_i} \nabla w \nabla v + \nabla w \nabla v + 0.01 \int_{\Gamma_{ext} \setminus \Gamma_{root}} wv, \quad (5.102)$$

where $\bar{\Theta}_q = \min_{\mu \in \mathcal{D}} \Theta^q(\mu)$, $q = 1, \dots, Q$. We note that this bound conditioner is independent of μ , the description of the more sophisticated bound conditioners could be found in [52, 57]. From (5.98) we can easily see that $\hat{a}(w, v)$ satisfies (5.52) and hence is suitable for our purposes. For this choice of bound conditioner the constants – as given in (5.52) – assume the following values: $\alpha_0 = 0.040197$, $\rho_{min} = 1$, $\rho_{max} = 100.00$, $\gamma_0 = 10.529323$.

M	$\text{cond}(\underline{B}^M)$	Λ_M
3	6.85e+01	1.70
6	1.18e+03	2.98
9	9.81e+03	2.93
12	8.00e+04	2.96
15	3.31e+05	4.63
18	7.72e+05	4.73
21	4.58e+06	7.62
24	5.39e+06	7.00
27	3.01e+07	13.05
30	1.98e+08	27.29
35	2.32e+08	12.82
40	4.32e+08	8.93

Table 5.1: Minimax coefficient approximation method, 2D locally non-linear problem: $\text{cond}(B^M)$ and Λ_M as functions of M .

In Figure 5-5 we present the convergence plots for the norm of the error $\|e_{NM}\|_Y = \|u(\mu) - u_{NM}(\mu)\|_Y$ as a function of M for several constant values of N . All the numerical results – unless specifically stated otherwise – are averaged over S_{test} , a sample set of cardinality $N_{test} = 10000$ which provides some level of statistical significance for our test. For each sub-plot in Figure 5-5 the corresponding value of N is displayed in the top right corner of that sub-plot.

As we observe from Figure 5-5, for every particular value of N the accuracy of the solution becomes better as we increase M . However, once M becomes sufficiently large, the accuracy of the solution stops to grow with M ($\|e_{NM}\|$ stops to decrease) and remains at practically the same level which is reflected by a "flat region" of each sub-plot of Figure 5-5. This behavior of $\|e_{NM}\|$ strongly suggests that there is no need to pick M to be too big since it does not contribute to the accuracy of the solution and at the same time incurs a higher operation count for both off-line and – even more important – on-line stages. We provide more details on $M - N$ analysis later in this Chapter.

Turning to Figure 5-6 we see that for every particular value of M $\|e_{NM}\|$ is a *strictly decreasing* function of N . Essentially, Figure 5-6 demonstrates the same kind of phenomenon we observed in Figure 5-5: for $M = 15, \dots, 40$ the convergence plots appear to be very close to each other.

In Figure 5-7 we present the plots for both $\|e_{NM}\|$ and Δ_{NM} as functions of N for several fixed values of M . We note that both $\|e_{NM}\|$ and Δ_{NM} are strictly decreasing functions of N and when M becomes large enough the plots for both $\|e_{NM}\|$ and Δ_{NM} collapse into two single plots. We

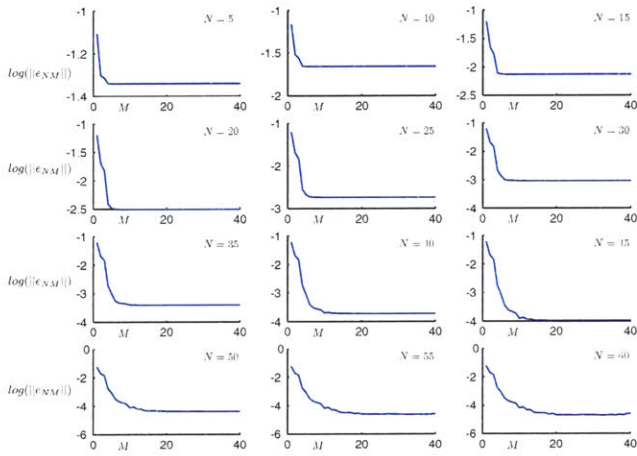


Figure 5-5: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ as a function of M while N is fixed.

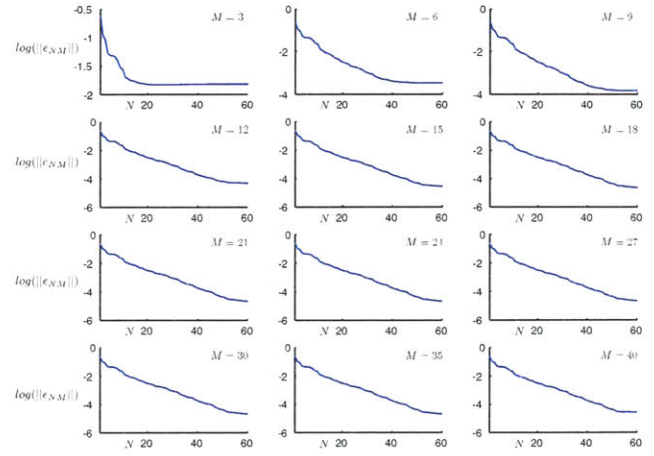


Figure 5-6: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ as a function of N while M is fixed.

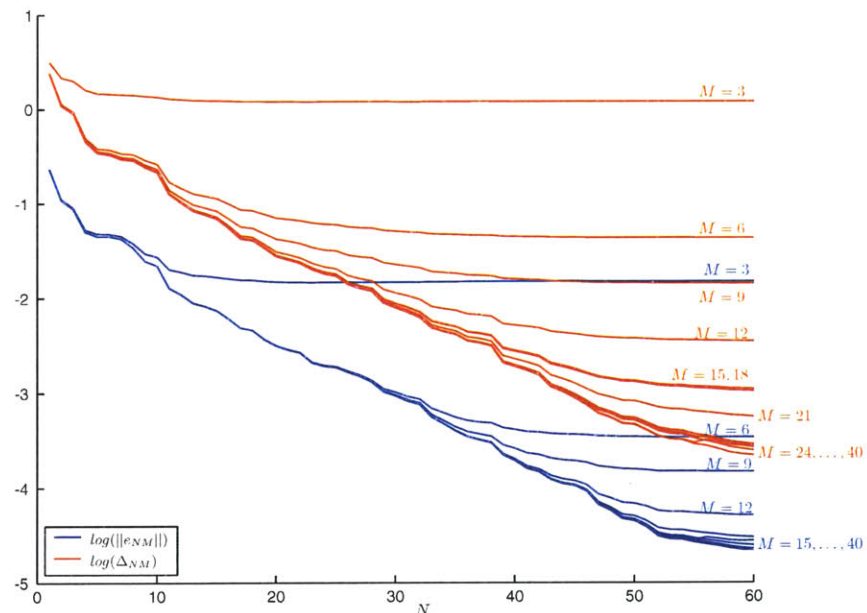


Figure 5-7: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ and Δ_{NM} as functions of N while M is fixed.

note that the convergence in $\|e_{NM}\|$ with respect to M is slightly better than that of Δ_{NM} .

N	$\frac{\ e_{NM}\ _Y}{\ u\ _Y}$	$\frac{\Delta_{NM}}{\ u\ _Y}$
5	4.55e-02	3.56e-01
10	2.24e-02	2.22e-01
15	7.62e-03	7.21e-02
20	3.29e-03	2.89e-02
25	1.98e-03	1.84e-02
30	1.01e-03	8.55e-03
35	4.24e-04	4.48e-03
40	2.05e-04	2.10e-03
45	1.12e-04	1.02e-03
50	4.87e-05	5.74e-04
55	2.86e-05	3.76e-04
60	2.38e-05	3.04e-04

Table 5.2: Minimax coefficient approximation method, 2D locally non-linear problem: relative convergence of $\|e_{NM}\|_Y$, Δ_{NM} as functions of N , $M = 30$.

In Table 5.2 we provide the rate of relative convergence for $\frac{\|e_{NM}\|_Y}{\|u\|_Y}$ and $\frac{\|\Delta_{NM}\|_Y}{\|u\|_Y}$ as functions of N for $M = 30$. We chose this particular value of M based on the observation of Figures 5-5, 5-6, and 5-7. We conclude that the MCAM is able to achieve high level of relative accuracy which is by all means sufficient for engineering applications.

Effectivity

In Figures 5-8 and 5-9 we treat the plots of effectivity η_{NM} as defined in (5.70) in the same manner as we presented the plots for $\|e_{NM}\|$. In Figure 5-8 we present η_{NM} as a function of M for several fixed values of N . Next, in Figure 5-9 η_{NM} is plotted as a function of M while N is fixed at different values. From Figure 5-8 we observe that when M becomes large enough η_{NM} reaches a constant level. Consistently with the last observation, we see that the effectivity plots in Figure 5-9 for $M = 27, \dots, 40$ are almost identical - the behavior we already observed in the case of convergence plots in Figures 5-6 and 5-7: since for these values of M both $\|e_{NM}\|$ and Δ_{NM} almost do not change with M we would expect the same kind of behavior from the effectivity.

We next note that the effectivity remains in the range of [8,150] for all values of N and M and stays in the range of [8,15] for higher values of M ($M = 24, \dots, 40$) which means that for these values of M our error bounds are indeed sharp. For the smaller values of M ($M = 1, \dots, 10$) the effectivity is slightly higher. We explain why this happens in the next Section. At the same time we

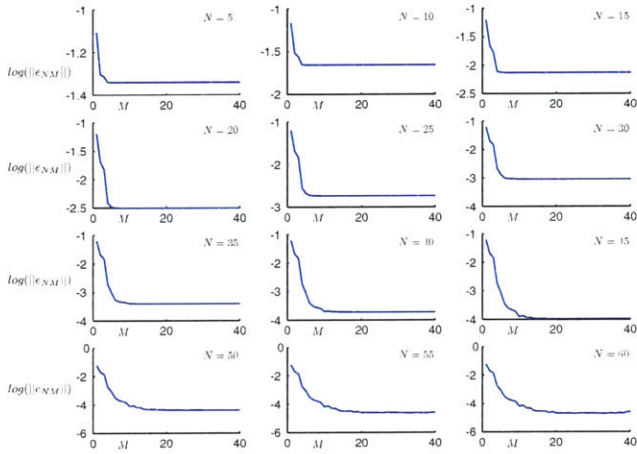


Figure 5-5: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ as a function of M while N is fixed.

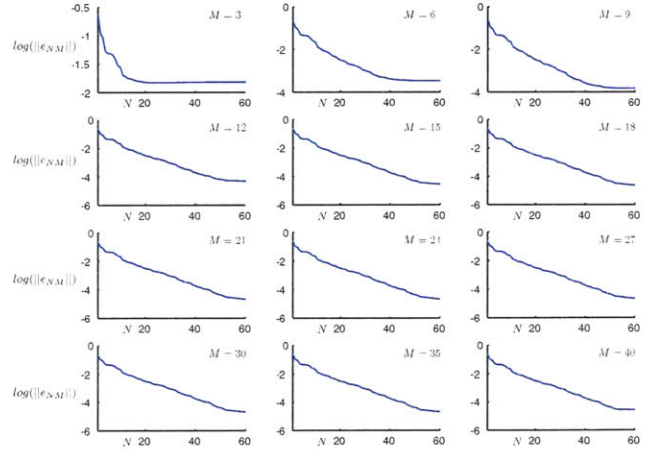


Figure 5-6: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ as a function of N while M is fixed.

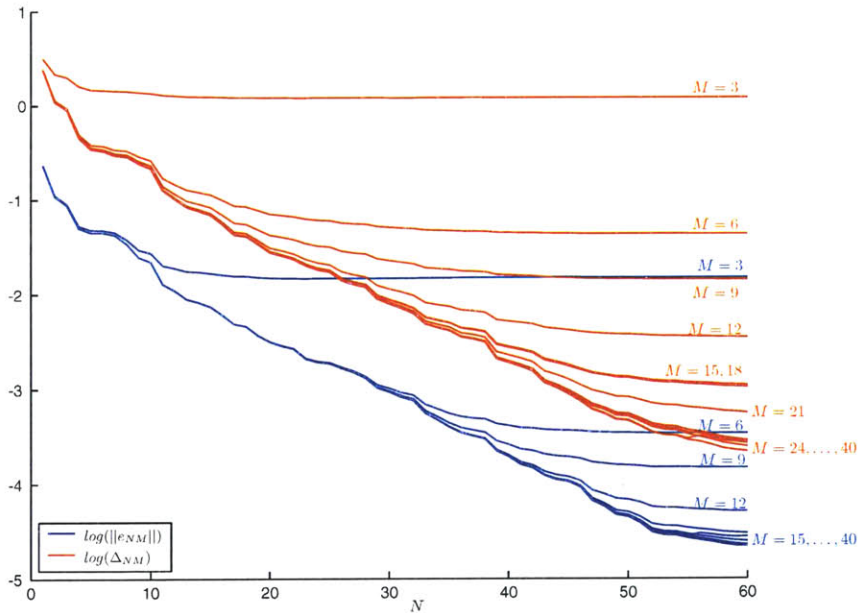


Figure 5-7: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ and Δ_{NM} as functions of N while M is fixed.

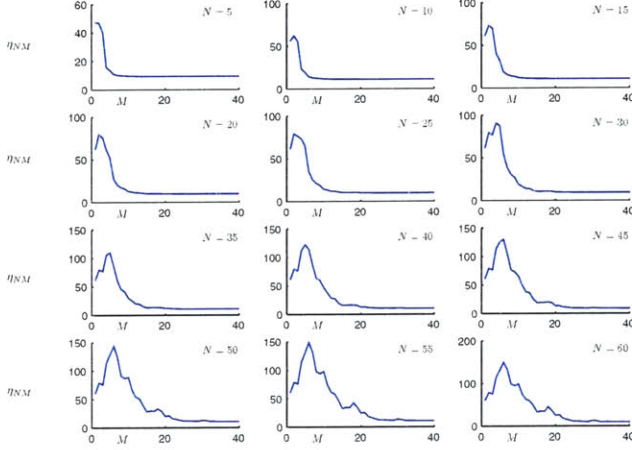


Figure 5-8: Minimax coefficient approximation method, 2D locally non-linear problem: effectivity η_{NM} as a function of M while N is fixed.

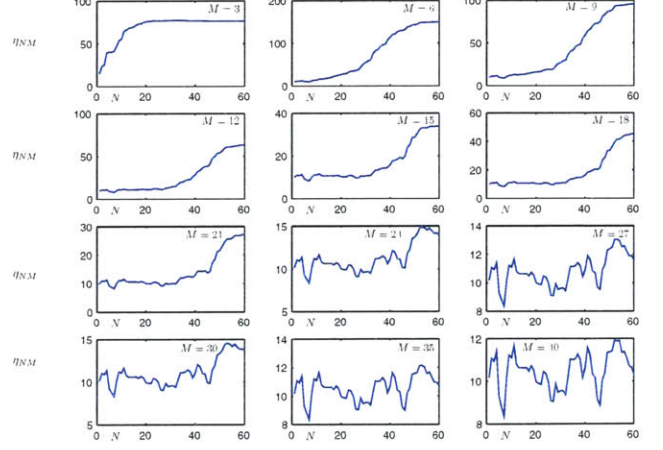


Figure 5-9: Minimax coefficient approximation method, 2D locally non-linear problem: effectivity η_{NM} as a function of N while M is fixed.

note that these smaller values of M are usually not that interesting from the practical point of view since the accuracy of the solution for these values is marginal at best; to support this statement we turn to Figures 5-5, 5-6, and 5-7.

Error Bound Decomposition

We now revisit the error equation (5.54). As it was shown in Section 5.3.2

$$\alpha(\mu)(\|e_{NM}(\mu)\|)^2 \leq a^A(e_{NM}(\mu), e_{NM}(\mu); \mu) + \int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_{NM}(\mu); \mu))e_{NM}(\mu) =$$

$$R_1(e_{NM}(\mu); \mu) + R_2(e_{NM}(\mu); \mu) \leq \tilde{\Delta}_{NM,1}(\mu) + \tilde{\Delta}_{NM,2}(\mu) + 2\sqrt{\tilde{\Delta}_{NM,1}(\mu)\tilde{\Delta}_{NM,2}(\mu)}. \quad (5.103)$$

In Figures 5-10, 5-11, 5-12 and 5-13 we provide the comparison of $\tilde{\Delta}_{NM,1}$, $|R_1(e_{NM})|$ and $\tilde{\Delta}_{NM,2}$, $|R_2(e_{NM})|$ as functions of M and N in the same manner as we presented the plots for the effectivity. Our goal now is to explain what the cause of high effectivities is for the low values of M .

As we observe from Figures 5-10 and 5-11, the difference between $|R_1(e_{NM})|$ and $\tilde{\Delta}_{NM,1}$ remains quite small. At the same time from Figures 5-12 and 5-13 we observe that for smaller values of M ($M = 1, \dots, 9$) the spread between $\tilde{\Delta}_{NM,2}$ and $|R_2(e_{NM})|$ remains relatively large. To complete our analysis in Figures 5-14 and 5-15 we provide the dependencies of the relative contributions of

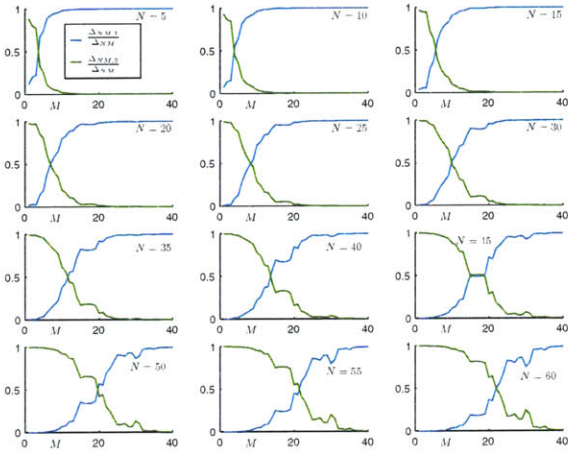


Figure 5-14: Minimax coefficient approximation method, 2D locally non-linear problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of M while N is fixed.

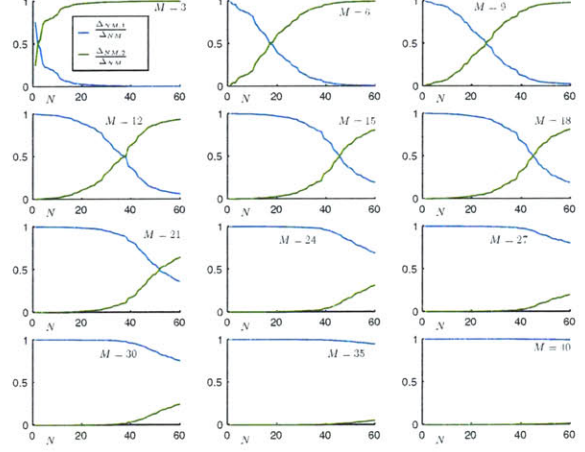


Figure 5-15: Minimax coefficient approximation method, 2D locally non-linear problem: contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to Δ_{NM} as functions of N while M is fixed.

each of the terms $\Delta_{NM,1}$ and $\Delta_{NM,2}$ to the error bound Δ_{NM} (note the absence of the \sim sign) as functions of M and N .

As we see from Figure 5-14, initially for small values of M the contribution of $\Delta_{NM,2}$ dominates that of $\Delta_{NM,1}$. However, as we keep increasing M $\Delta_{NM,1}$ and $\Delta_{NM,2}$ switch roles: now $\Delta_{NM,1}$ becomes the main contributor to the error bound Δ_{NM} whereas the contribution of $\Delta_{NM,2}$ becomes rather small.

From Figure 5-15 we observe that for $M = 1, \dots, 21$ and for small values of N $\Delta_{NM,1}$ initially constitutes the major part of Δ_{NM} . As we increase N , $\Delta_{NM,1}$ and $\Delta_{NM,2}$ trade places: $\Delta_{NM,2}$ becomes dominant while the effect of $\Delta_{NM,1}$ almost vanishes. This observation coupled with the observation about the large spread between $|R_2(e_{NM})|$ and $\tilde{\Delta}_{NM,2}$ in Figure 5-13 explains the high values of effectivities in Figures 5-8 and 5-9 for low values of M .

There are two more factors which might affect effectivities in a bad way though, fortunately – as we observe from Figures 5-8 and 5-9 – it does not happen. First, we look again at (5.103) to see that so far we did not take into account the $2\sqrt{\tilde{\Delta}_{NM,1}\tilde{\Delta}_{NM,2}}$ term which is always positive as well as $\tilde{\Delta}_{NM,1}$ and $\tilde{\Delta}_{NM,2}$; at the same time $R_1(e_{NM})$ and $R_2(e_{NM})$ can be negative.

Second, in (5.103) we note the presence of $\int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_{NM}(\mu); \mu))e_{NM}(\mu)$ term which – provided that both $u(\mu)$ and $u_{NM}(\mu)$ are positive in Ω_{nl} – is also positive. Again we emphasize that this is the reason why the positivity of the solutions $u(\mu)$ and $u_{NM}(\mu)$ in this case is so important

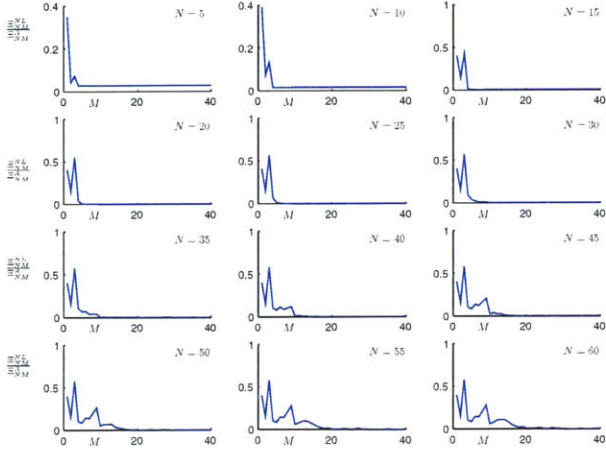


Figure 5-16: Minimax coefficient approximation method, 2D locally non-linear problem: study of contribution of non-linear term: $\frac{\Xi_{NM}^{NL}}{\Xi_{NM}^A}$ and as a function of M while N is fixed.

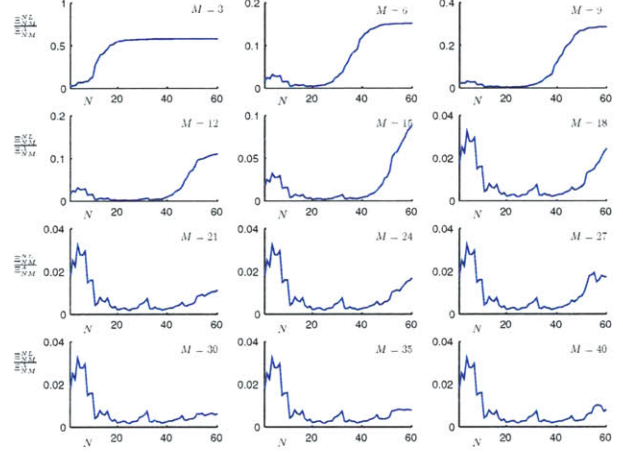


Figure 5-17: Minimax coefficient approximation method, 2D locally non-linear problem: study of contribution of non-linear term: $\frac{\Xi_{NM}^{NL}}{\Xi_{NM}^A}$ and as a function of N while M is fixed.

to us. We now denote the two terms in (5.103) as

$$\Xi_{NM}^A(\mu) = \alpha^A(e_{NM}(\mu), e_{NM}(\mu); \mu) \quad (5.104)$$

$$\Xi_{NM}^{NL}(\mu) = \int_{\Omega_{nl}} (g(u(\mu); \mu) - g(u_{NM}(\mu); \mu)) e_{NM}(\mu) \quad (5.105)$$

In Figures 5-16 and 5-17 we provide the the ratio $\frac{\Xi_{NM}^{NL}}{\Xi_{NM}^A}$ as a function of N and M . Based on our numerical simulation $\Xi_{NM}^{NL}(\mu)$ as well as $u_{NM}(\mu)$ stay positive for all values of μ in S_{test} (though we can guarantee the positivity of $u(\mu)$ in mathematically rigorous way – we cannot say the same about $u_{NM}(\mu)$) and what is also quite good, the value of $\Xi_{NM}^{NL}(\mu)$ is small compared to $\Xi_{NM}^A(\mu)$ which ensures that the former does not affect the effectivity in a significantly bad way. Turning to actual numbers, we note that Ξ_{NM}^{NL} never exceeds $\Xi_{NM}^A(\mu)$ and at most reaches only half of the value of $\Xi_{NM}^A(\mu)$.

We finally take another look at Figure 5-14 and note that for each N there exists a value of M – which we denote $M^*(N)$ – when the contributions of $\Delta_{NM,1}$ and $\Delta_{NM,2}$ are approximately equal to each other. Increasing M further past $M^*(N)$ brings us to the point where the effect of $\Delta_{NM,2}$ becomes small compared to that of $\Delta_{NM,1}$. We denote this second characteristic value of

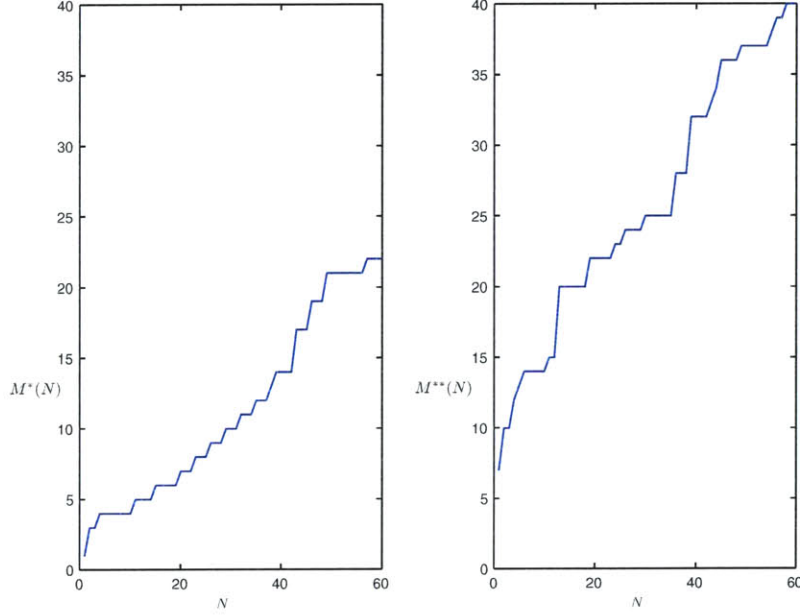


Figure 5-18: Minimax coefficient approximation method, 2D locally non-linear problem: M^* and M^{**} as functions of N .

M as $M^{**}(N)$. To be more rigorous,

$$M^{**}(N) = \min \left[\arg \min_M \frac{\Delta_{NM,2}}{\Delta_{NM}} \leq 0.01, M_{max} \right] \quad (5.106)$$

In Figure 5-18 we present the plots for M^* and M^{**} as functions of N . Clearly, $\forall N, M^*(N) < M^{**}(N)$. As we observe from Figure 5.4.2 for high values of N , $M^{**}(N) = M_{max}$.

Further $M - N$ Analysis

Taking into consideration the operation count of $K^{iter}O((Q + M)N^2) + O(N^3) + O((Q + M)^2N^2) + O(Mn_{nl})$ for the on-line stage as it was shown in Section 5.3.3 our goal would be to choose N and M as small as possible and at the same provide good accuracy for our reduced basis approximation. As we can see from the convergence plots of Figure 5-6, $\|e_{NM}\|$ is a strictly decreasing function of N . Also, looking at Figures 5-5 and 5-6, we note that the accuracy of the solution depends on N in a greater extent than on M . Thus, in order to obtain the best accuracy we should choose N as large as possible. However, in an arbitrary case the user might have certain constraints which might prevent him or her from choosing the largest value of N . Clearly, there is a trade-off between the accuracy of the reduced basis approximation and the on-line computational complexity.

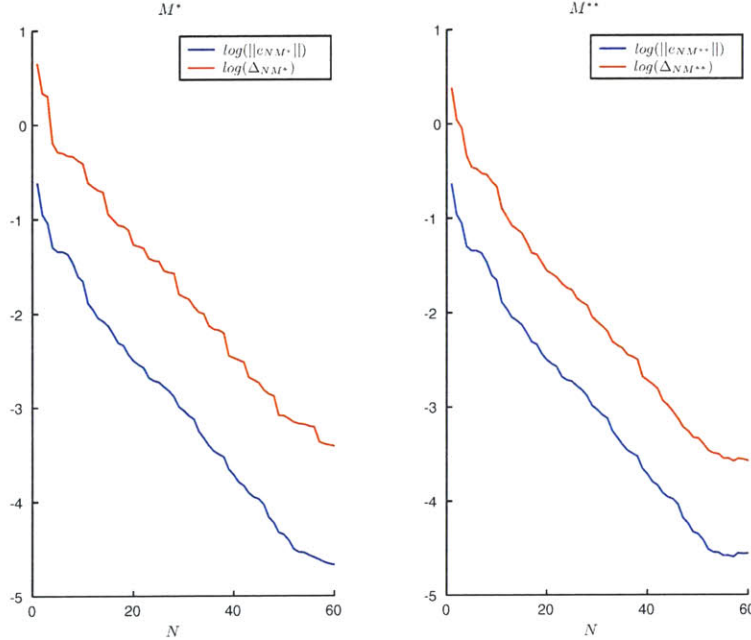


Figure 5-19: Minimax coefficient approximation method, 2D locally non-linear problem: $\|e_{NM}\|_Y$ and Δ_{NM} as functions of N and $M^*(N)$, $M^{**}(N)$.

However, if somehow the value of N is chosen we need to determine the optimal value of M for this particular value of N . To maximize the accuracy while keeping M as small as possible our suggestion is to choose M to be $M^* \leq M \leq M^{**}$. Such a choice of M is supported by Figure 5-5 – for these values of M the accuracy essentially reaches its best value, it is slightly better for M^{**} than for M^* ; the difference is quite small as we can see from the following convergence plots as we plot e_{NM} and Δ_{NM} as functions of N , and $M^*(N)$, $M^{**}(N)$ in Figure 5-19.

In Figure 5-20 we provide similar plots for the effectivity as a function of N and M equal to $M^*(N)$ and $M^{**}(N)$. We note that the choice of $M = M^{**}$ yields slightly sharper effectivity than that of M^* , by a factor that ranges between 1.5 and 3.5 which appears to be consistent with the definitions of M^* and M^{**} .

As in the most cases with the reduced basis methods, we cannot – especially in the multi-parameter case – provide an a priori algorithm that would guarantee a certified level of accuracy for all future parameter values. However, the heuristic algorithm for selecting optimal M and N is the following. We start from constructing the sample S_M^g such that we achieve a good accuracy in M . We can measure that by running sufficient number of tests and comparing the value of $\epsilon_M(\mu)$ as defined in (5.42) to the accuracy of the finite element method [51]. We also know that if M becomes too large the condition number of \underline{B}_M would be become huge and this fact puts a limitation which

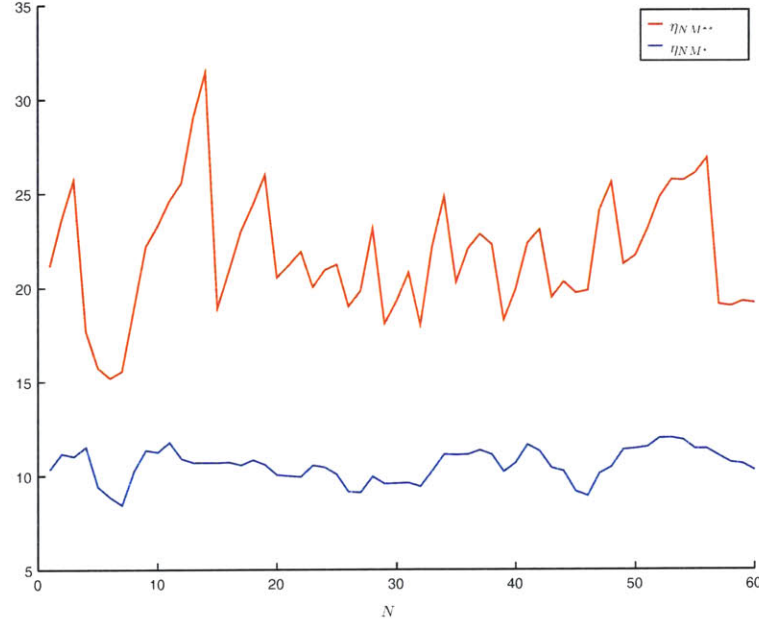


Figure 5-20: Minimax coefficient approximation method, 2D locally non-linear problem: effectivity η_{NM} as a function of N , and $M^*(N)$, $M^{**}(N)$.

is simultaneously an indirect indication that M is already sufficiently big. Given the fact that we can approximate the non-linear term well enough, our goal now is to find the value of N which provides the desired level of accuracy. The validity of our selection could be again tested based on a big enough sample of randomly chosen points. Once we established the approximately "optimal" values of M and N we can tweak them a little bit to see if, for example M is too big or if can increase M or N just slightly to gain a big increase in the accuracy. This heuristic algorithm is in fact consistent with a choice of $M = M^*(N)$ or $M^{**}(N)$. We conclude this Section by reiterating that regardless of the choice of M , N , S_M^g and S_N our error bounds remain rigorous.

Computational Costs

In Figures 5-21 and 5-22 we present the dependencies of the actual on-line computational times on N and M . We denote the time required to perform one Newton iteration for the reduced basis method as $t_{u_{NM},iter}$ (or just $t_{u_{NM}}$) and the time to evaluate the error bound as $t_{\Delta_{NM}}$. Since the operation counts to perform a Newton iteration and to construct the error bound are $O((Q + M)N^2 + N^3)$ and $O((Q + M)^2N^2) + O(n_{nl})$, respectively, we expect both $t_{u_{NM},iter}$ and $t_{\Delta_{NM}}$ to increase with M and N . We plot $t_{u_{NM},iter}$ (in red) and $t_{\Delta_{NM}}$ (in green) in Figures 5-21 and 5-22.

As we observe from Figure 5-21, $t_{u_{NM}}$ exhibits slightly stronger dependence on N than $t_{\Delta_{NM}}$

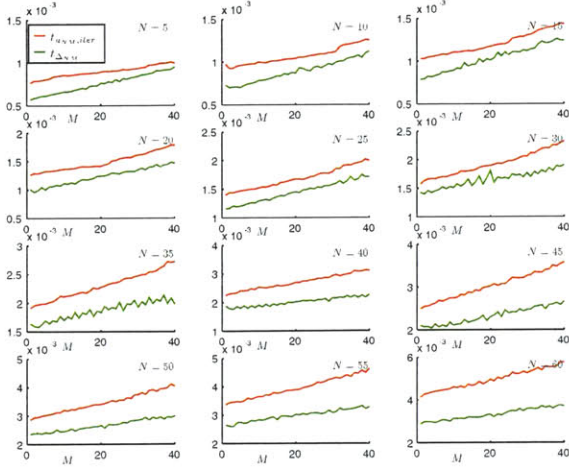


Figure 5-21: Minimax coefficient approximation method, 2D locally non-linear problem: on-line computational times as functions of N while M is fixed.

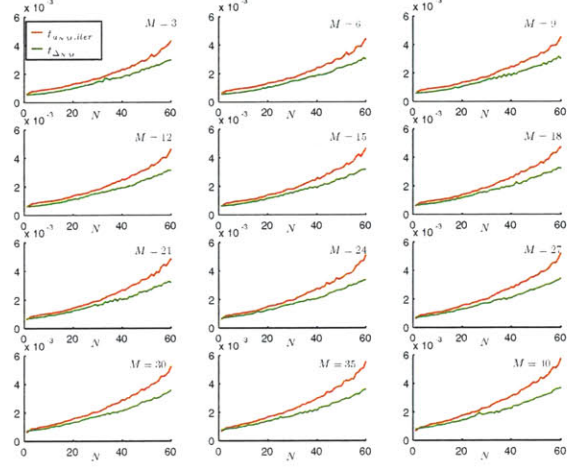


Figure 5-22: Minimax coefficient approximation method, 2D locally non-linear problem: on-line computational times as functions of M while N is fixed.

which is supported by the presence of $O(N^3)$ term in the operation count of the former. At the same time, both $t_{u_{NM}}$ and $t_{\Delta_{NM}}$ grow at the same rate with M as shown in Figure 5-22.

Finally, in Table 5.3 we present the dependence of the total computational time $t_{tot} = K^{iter} t_{u_{NM}} + t_{\Delta_{NM}}$ for the online stage as a function of N when M is selected to be either $M^*(N)$ or $M^{**}(N)$. As we see from this Table, both choices yield significant computational savings compared to the conventional finite element method, even for the largest values of N the efficiency ratio $\frac{t_{tr}}{t_{tot}}$ stays close to 500. The main reason why the computational savings for the non-linear case are significantly higher than the savings we observed in Chapters 2, 3, and 4 is that we actually perform Newton iterations for the MCAM much faster than for the finite element "truth" method and this effect is magnified by the factor K^{iter} . We also observe that for this particular model problem the choice of M^* or M^{**} does not result in a significant difference in the on-line computational times.

Conclusions

Based on the theoretical and numerical results provided in this Chapter we can state that the minimax coefficient approximation reduced basis method provides an efficient and yet accurate way to solve locally non-linear problems with parametric dependence. The reduced basis method yields certified high accuracy with respect to the "truth" solution, the rigorous error bounds we provide appear to be quite sharp. At the same time the method demonstrates significant computational savings. The performance of the minimax approximation coefficient method fulfills the goals set in

N	$M^*(N)$		t_{tr}	$M^{**}(N)$	
	t_{tot,M^*}	$\frac{t_{tr}}{t_{tot,M^*}}$		$\frac{t_{tr}}{t_{tot,M^{**}}}$	$t_{tot,M^{**}}$
5	7.76e-03	3368.59	26.13	3108.02	8.41e-03
10	9.26e-03	2821.15	26.13	2635.54	9.92e-03
15	1.05e-02	2495.64	26.13	2265.29	1.15e-02
20	1.31e-02	2000.96	26.13	1820.90	1.44e-02
25	1.47e-02	1772.90	26.13	1568.11	1.67e-02
30	1.72e-02	1522.65	26.13	1342.59	1.95e-02
35	2.09e-02	1250.61	26.13	1119.01	2.34e-02
40	2.49e-02	1048.68	26.13	918.87	2.84e-02
45	2.84e-02	919.40	26.13	785.14	3.33e-02
50	3.36e-02	777.98	26.13	678.56	3.85e-02
55	3.80e-02	686.90	26.13	588.77	4.44e-02
60	4.88e-02	535.21	26.13	470.18	5.56e-02

Table 5.3: Minimax coefficient approximation method, 2D locally non-linear problem: t_{tot} , the total on-line computational time as functions of N and $M^*(N)$, $M^{**}(N)$.

Section 1.3.1.

Chapter 6

Summary and Future Work

In this final Chapter we conclude the thesis by providing final observations regarding the reduced basis methods we described and giving an outline for the directions of the future work.

6.1 Summary

In this thesis we described and studied the following reduced basis methods: the classical reduced basis method (CRBM), the partition of unity method (PUM), and the minimax coefficient approximation method (MCAM). All of these methods are applied to problems which are governed by partial differential equations with parametric dependencies.

All of the reduced basis methods we present are developed according to the following common standards. The construction of these methods relies on a two stage off-line/on-line computational framework. We first run the expensive off-line stage during which we precompute and store all the necessary data (vectors and matrices) for the on-line stage. It is important to emphasize that the off-line stage is run only once and after that for each new value of the parameter μ we only have to run the cheap on-line stage. The numerically efficient on-line stage consists of obtaining an accurate reduced basis approximation $u_N(\mu)$ to the "truth" finite element solution $u(\mu)$ and subsequent rigorous a posteriori error bound $\Delta_N(\mu)$ for the norm of the error $\|u(\mu) - u_N(\mu)\|$. We require that $u_N(\mu)$ is a precise approximation of $u(\mu)$ and that the error bound is sharp in a sense that the ratio of $\frac{\Delta_N(\mu)}{\|u(\mu) - u_N(\mu)\|}$ – which we call effectivity – stays close $O(10)$. All three methods – the CRBM, the PUM, and the MCAM – are compliant with these standards within the limits of their applicability, the last statement is confirmed by the numerical tests described in Chapters 2,

3, 4, and 5.

The classical reduced basis methods previously described in [49, 52, 18, 42] is important for us because we build the methodology for the PUM and the MCAM using the ideas first developed for the CRBM. The area of application for the CRBM is limited to linear affine coercive problems.

We also provide the description of the PUM which can be applied to a broader class of locally non-affine coercive problems. The PUM relies on domain decomposition techniques [20, 19] and allows us to decouple contributions from affine and locally non-affine components of the problem. We also speak about application of the PUM to purely affine problems and note that this method has a clear advantage over the CRBM in terms of the off-line computational effort. The comparative analysis of the CRBM and the PUM features identifies that the PUM is a more efficient method with respect to the off-line stage. There is no clear winner with respect to the on-line computational efficiency because for some problems the PUM performs better and for other problems it is the CRBM which has the edge.

The MCAM is capable of treating both locally non-affine and locally non-linear problems and relies on replacing the underlying non-affine/non-linear component by an affine like approximation [5]. The comparative analysis of the MCAM and the PUM applied to locally non-affine problems leads us to the following conclusions. The PUM is again a better method with respect to the off-line stage and it also better treats locally non-affine problems with more complex parametric dependencies. At the same time for problems with simpler parametric dependencies the MCAM demonstrates a much better performance than the PUM in terms of the on-line computational times.

6.2 Future Work

We now suggest the ideas for the future work which could be carried out based on the contributions presented in this thesis.

First, we believe it is possible to construct a "synergy" PUM-MCAM method for locally non-affine problems. As it was described in Section 3.2.3, the domain decomposition nature of the PUM essentially allows us to break the error estimation procedure into several sub problems most of which are affine. We propose to apply the MCAM to the remaining non-affine problems (3.33) because we would expect simpler parametric dependence from these problems and this is the area where the MCAM demonstrates its best performance. The other advantage of this "synergy" method

is that it will allow us to avoid solution of the $n_{na} \times n_{na}$ system of equations as given by (3.33) and hence potentially generate additional computational savings. We cannot guarantee that this "synergy" method will yield better results for all the problems in the class of locally non-affine problems, however we expect it to be true at least for some problems in this class. In general, the PUM-MCAM "synergy" method will enable us to address problems with more complicated parametric dependencies more efficiently.

Second, as we mentioned in Section we compare our reduced basis approximation to a finite element solution $u_{\mathcal{N}}(\mu)$. Recently, there appeared several works [59, 58] which made it possible to evaluate the error of the finite element approximation with respect to the analytical (the real "truth") solution. It is thus quite interesting to be able to provide the error bounds for the reduced basis approximation not with respect to the finite element but rather with respect to the analytical solution.

Finally, the methodology developed in this thesis can also be extended to other classes of problems with locally non-affine/locally non-linear parametric dependence such as time-dependent [10] and non-coercive problems [56, 22], some of these of directions are currently explored by other researchers. It is quite very important to find new applications for the reduced methods since it is the particular applications which motivate us to improve the current methodology and helps us to discover new aspects of our reduced methods.

Appendix A

Proof of Formula for $\mathcal{T}(N, \kappa)$

In order to obtain the expression we construct the set $\mathcal{K}(N, \kappa)$ containing $N + \kappa - 1$ elements: $\{1, \dots, N, j_1^{gap}, \dots, j_N^{gap}\}$. For each κ -tuple $\underline{j} \in \mathcal{P}_\kappa^N$ we construct a subset $\mathcal{K}_j \subset \mathcal{K}(N, \kappa)$ of cardinality κ which comprises of all the *unique* integers in \underline{j} plus all elements j_i^{gap} corresponding to those i for which $j_i = j_{i-1}, i = 2, \dots, \kappa$. It is easy to see that this mapping between \mathcal{P}_κ^N and $\mathcal{K}(N, \kappa)$ is a bijection and it spans all possible subsets of cardinality κ in $\mathcal{K}(N, \kappa)$. Hence, the cardinality of \mathcal{P}_κ^N is equal to $\mathcal{T}(N, \kappa) = \frac{(N-1+\kappa)!}{\kappa!(N-1)!}$.

Bibliography

- [1] A. Valli A. Quarteroni. *Domain decomposition methods for partial differential equations*. Oxford ; New York : Clarendon Press, 1999.
- [2] S. Ali. *Real-time Optimal Parametric Design using the Assess-Predict-Optimize Strategy*. PhD thesis, Singapore-MIT Alliance, Nanyang Technological University, Singapore, 2003.
- [3] B. O. Almroth, P. Stern, and F. A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16:525–528, May 1978.
- [4] E. Balmes. Parametric families of reduced finite element models. theory and applications. *Mechanical Systems and Signal Processing*, 10(4):381–394, 1996.
- [5] M. Barrault, N. C. Nguyen, Y. Maday, and A. T. Patera. An “empirical interpolation” method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Série I.*, 2004. Accepted, to appear.
- [6] A. Barrett and G. Reddien. On the reduced basis method. *Z. Angew. Math. Mech.*, 75(7):543–549, 1995.
- [7] K.-J. Bathe. *Finite element procedures*. Prentice Hall, 1996.
- [8] J. P. Fink and W. C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *Z. Angew. Math. Mech.*, 63:21–28, 1983.
- [9] M. Fortin and F. Brezzi. *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer Verlag, July 1991.
- [10] M. Grepl. *Reduced-Basis Approximations for Time-Dependent Partial Differential Equations: Application to Optimal Control*. PhD thesis, Massachusetts Institute of Technology, 2005. In progress.

- [11] M. A. Grepl, N. C. Nguyen, K. Veroy, A. T. Patera, and G. R. Liu. Certified rapid solution of parametrized partial differential equations for real-time applications. In *Proceedings of the 2nd Sandia Workshop of PDE-Constrained Optimization: Towards Real-Time and On-Line PDE-Constrained Optimization*, SIAM Computational Science and Engineering Book Series, 2004. Submitted.
- [12] M. A. Grepl and A. T. Patera. *A Posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN Math. Model. Numer. Anal.*, 2004. Submitted.
- [13] K. Ito and S. S. Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics*, 143(2):403–425, July 1998.
- [14] K. Ito and J. D. Schroeter. Reduced order feedback synthesis for viscous incompressible flows. *Mathematical and Computer Modelling*, 33(1-3):173–192, Jan-Feb 2001.
- [15] M. Lin Lee. Estimation of the error in the reduced basis method solution of differential algebraic equation systems. *SIAM Journal on Numerical Analysis*, 28(2):512–528, 1991.
- [16] T. Leurent. Reduced basis output bounds for linear elasticity: Application to microtruss structures. Master’s thesis, Massachusetts Institute of Technology, 2001.
- [17] G.P. Akilov L.V. Kantorovich. *Functional Analysis in Normed Spaces*. The Macmillan Company, 1964.
- [18] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Acad. Sci. Paris, Série I*, 331(2):153–158, July 2000.
- [19] L. Machiels, Y. Maday, and A. T. Patera. A “flux-free” nodal Neumann subproblem approach to output bounds for partial differential equations. *C. R. Acad. Sci. Paris, Série I*, 330(3):249–254, February 2000.
- [20] L. Machiels, Y. Maday, and A. T. Patera. Output bounds for reduced-order approximations of elliptic partial differential equations. *Comp. Meth. Appl. Mech. Engrg.*, 190(26-27):3413–3426, 2001.

- [21] L. Machiels, A. T. Patera, and D. V. Rovas. Reduced basis output bound methods for parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 2001. Submitted.
- [22] Y. Maday, A. T. Patera, and D. V. Rovas. A blackbox reduced-basis output bound method for noncoercive linear problems. In D. Cioranescu and J.-L. Lions, editors, *Nonlinear Partial Differential Equations and Their Applications, Collège de France Seminar Volume XIV*, pages 533–569. Elsevier Science B.V., 2002.
- [23] Y. Maday, A. T. Patera, and D.V. Rovas. Petrov-Galerkin reduced-basis approximations to noncoercive linear partial differential equations. In progress.
- [24] Y. Maday, A. T. Patera, and G. Turinici. Global *a priori* convergence theory for reduced-basis approximation of single-parameter symmetric coercive elliptic partial differential equations. *C. R. Acad. Sci. Paris, Série I*, 335(3):289–294, 2002.
- [25] Y. Maday, A. T. Patera, and G. Turinici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing*, 17(1–4):437–446, December 2002.
- [26] Z. Milka. Finite element solution of a stationary heat conduction equation with the radiation boundary condition. *Appl. of Math.*, 38(1):67–79, 1993.
- [27] D. A. Nagy. Modal representation of geometrically nonlinear behaviour by the finite element method. *Computers and Structures*, 10:683–688, 1979.
- [28] N. C. Nguyen. *Reduced-Basis Approximation and A Posteriori Error Bounds for Nonaffine and Nonlinear Partial Differential Equations: Application to Inverse Analysis*. PhD thesis, Singapore-MIT Alliance, National University of Singapore., 2005. In progress.
- [29] A. K. Noor. Recent advances in reduction methods for nonlinear problems. *Comput. Struct.*, 13:31–44, 1981.
- [30] A. K. Noor. On making large nonlinear problems small. *Comp. Meth. Appl. Mech. Engrg.*, 34:955–985, 1982.
- [31] A. K. Noor, C. M. Andresen, and J. A. Tanner. Exploiting symmetries in the modeling and analysis of tires. *Comp. Meth. Appl. Mech. Engrg.*, 63:37–81, 1987.

- [32] A. K. Noor, C. D. Balch, and M. A. Shibut. Reduction methods for non-linear steady-state thermal analysis. *Int. J. Num. Meth. Engrg.*, 20:1323–1348, 1984.
- [33] A. K. Noor and J. M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18(4):455–462, April 1980.
- [34] A. K. Noor and J. M. Peters. Multiple-parameter reduced basis technique for bifurcation and post-buckling analysis of composite plates. *Int. J. Num. Meth. Engrg.*, 19:1783–1803, 1983.
- [35] A. K. Noor, J. M. Peters, and C. M. Andersen. Mixed models and reduction techniques for large-rotation nonlinear problems. *Comp. Meth. Appl. Mech. Engrg.*, 44:67–89, 1984.
- [36] I. B. Oliveira. *A “HUM” Conjugate Gradient Algorithm for Constrained Nonlinear Optimal Control: Terminal and Regulator Problems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2002.
- [37] A. T. Patera, D. Rovas, and L. Machiels. Reduced-basis output-bound methods for elliptic partial differential equations. *SIAG/OPT Views-and-News*, 11(1), April 2000.
- [38] J. S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM J. Sci. Stat. Comput.*, 10(4):777–786, July 1989.
- [39] T. A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, October 1985.
- [40] T. A. Porsching and M. Lin Lee. The reduced basis method for initial value problems. *SIAM Journal on Numerical Analysis*, 24(6):1277–1287, 1987.
- [41] C. Prud’homme and A. T. Patera. Reduced-basis output bounds for approximately parametrized elliptic coercive partial differential equations. *Computing and Visualization in Science*, 2003. Accepted.
- [42] C. Prud’homme, D. Rovas, K. Veroy, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, March 2002.
- [43] C. Prud’homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reduced-basis output bound methods for parametrized partial differential equations. In *Proceedings Singapore-MIT Alliance Symposium*, January 2002.

- [44] Christophe Prud'homme, Dimitrios V. Rovas, Karen Veroy, and Anthony T. Patera. A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *M2AN Math. Model. Numer. Anal.*, 36(5):747–771, 2002.
- [45] J.-L. Lions R. Dautray. *Mathematical Analysis and Numerical Methods for Science and Technology*. Springer-Verlag, 1988.
- [46] B. Daya Reddy. *Introductory Functional Analysis with Application to Boundary Value Problems and Finite Elements*. Texts in Applied Mathematics. Springer, 1998.
- [47] W. C. Rheinboldt. On the theory and error estimation of the reduced basis method for multi-parameter problems. *Nonlinear Analysis, Theory, Methods and Applications*, 21(11):849–858, 1993.
- [48] D. V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. *IMA Journal of Applied Mathematics*, 2004. Submitted.
- [49] D.V. Rovas. *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, October 2002.
- [50] Y. Saad. *Iterative methods for sparse linear systems*. Philadelphia : SIAM, 2003.
- [51] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.
- [52] K. Veroy. *Reduced-Basis Methods Applied to Problems in Elasticity: Analysis and Applications*. PhD thesis, Massachusetts Institute of Technology, 2003. In progress.
- [53] K. Veroy, T. Leurent, C. Prud'homme, D. Rovas, and A. T. Patera. Reliable real-time solution of parametrized elliptic partial differential equations: Application to elasticity. In *Proceedings Singapore-MIT Alliance Symposium*, January 2002.
- [54] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible navier-stokes equations; Rigorous reduced-basis *a posteriori* error bounds. *Submitted to International Journal for Numerical Methods in Fluids*, 2004. (Special Issue — Proceedings for 2004 ICFD Conference on Numerical Methods for Fluid Dynamics, Oxford).
- [55] K. Veroy, C. Prud'homme, and A. T. Patera. Reduced-basis approximation of the viscous Burgers equation: Rigorous *a posteriori* error bounds. *C. R. Acad. Sci. Paris, Série I*, 337(9):619–624, November 2003.

- [56] K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera. *A posteriori* error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations (AIAA Paper 2003-3847). In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, June 2003.
- [57] K. Veroy, D. Rovas, and A. T. Patera. *A Posteriori* error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: “Convex inverse” bound conditioners. *Control, Optimisation and Calculus of Variations*, 8:1007–1028, June 2002. Special Volume: A tribute to J.-L. Lions.
- [58] J. Peraire Z.C. Xuan, K.H. Lee. The efficient computation of bounds for functionals of finite element solutions in large strain elasticity. *Computer Methods in Applied Mechanics and Engineering*, 191(43):4807–4826, 2002.
- [59] J. Peraire Z.C. Xuan, K.H. Lee. A posteriori output bound for partial differential equations based on elemental error bound computing. *Computational Science and its Application*, pages 1035–1044, 2003.