Reduced Basis Approximation and *A Posteriori* Error Estimation for Parametrized Partial Differential Equations

Anthony T. Patera   Gianluigi Rozza

Massachusetts Institute of Technology

Department of Mechanical Engineering

to appear in (tentative rubric)

MIT Pappalardo Graduate Monographs in Mechanical Engineering

Current Edition: V1.0 January 2007

Notice

This book is available at URL `http://augustine.mit.edu`.

In downloading this "work" (defined as any version of the book or part of the book), you agree

1. to attribute this work as:

   A.T. Patera and G. Rozza, *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*, Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering.

2. not to alter or transform this work in any way;
3. not to distribute this work in any form to any third parties; and
4. not to use this work for any commerical purposes.

Preface

Motivation

This book is about the evaluation of *input-output relationships* in which the *output* is evaluated as a functional of a *field* that is the solution of an *input-parametrized* partial differential equation (PDE). We will focus on applications in mechanics — heat and mass transfer, solid mechanics, acoustics, fluid dynamics — but we do not preclude other domains of inquiry within engineering (e.g., electromagnetics) or even more broadly within the quantitative disciplines (e.g., finance).

The *input-parameter* vector typically characterizes the geometric configuration, the physical or effective properties of the constitutive or phenomenological model, the boundary conditions and initial conditions, and any loads and sources. The *outputs of interest* might be the maximum system temperature, a crack stress intensity factor, structural resonant frequencies, an acoustic waveguide transmission loss, or a channel flowrate or pressure drop. Finally, the *fields* that connect the input parameters to the outputs can represent a distribution function, temperature or concentration, displacement, (acoustic) pressure, or velocity.

Our interest is in two particular contexts: the *real-time context*, and the *many-query context*. Both these contexts are crucial to computational engineering and to more widespread adoption and application of numerical methods for PDEs in engineering practice.

We first consider the *real-time context*; we can also characterize this context as "deployed" or "in the field" or "embedded." Typical activities in the real-time context fall within the broad "Assess, Predict, Act" framework. In the Assess stage we pursue parameter estimation or inverse analysis — to characterize current state; in the Predict stage we consider prognosis — to understand possible subsequent states; and in the Act stage we intervene to achieve our objectives — to influence realized future state. We now give several examples of these real-time processes;

note in all cases not all three stages of Assess, Predict, Act are invoked.

(*i*) Consider a crack in a critical structural component such as a composite-reinforced concrete support (or an aircraft engine). In the Assess stage we pursue Non-Destructive Evaluation (NDE) — say by vibration or thermal transient analysis — to determine the location and configuration of the delamination crack in the support. In the Predict stage we evaluate stress intensity factors to determine the critical load for brittle failure or the anticipated crack growth due to fatigue. And in the Act stage we modify the installation or subsequent mission profile to prolong life. In all cases, safety and economics require rapid and reliable response in the field. (See Part VIII for a detailed discussion of this particular problem.)

(*ii*) Consider an "immersed" object such as a tumor (or unexploded ordnance, or moving military target). In the Assess stage we apply parameter estimation techniques — say by acoustic or electromagnetic analysis — to determine the tumor location and geometric and physical characteristics. In the Predict stage we evaluate the potential lethality of the tumor. And in the Act stage we steer the intervention — therapy or surgery — to minimize the threat. Again, the timeliness *and* reliability of the analysis is all-important to safe and successful conclusion of the operation.

(*iii*) Consider heat treatment of a workpiece such as a turbine disk. In the Assess stage we apply inverse procedures to determine the (effective) heat transfer coefficients between the workpiece and the quenching bath. In the Predict stage we evaluate the anticipated residual stresses in the quenched workpiece for a given annealing schedule. And finally in the Act stage we apply optimal feedforward or feedback control to modify the annealing schedule in order to achieve lower residual stresses. For expensive materials, reliable quality for each workpiece "in process" is critical.

Clearly there are many other examples from other fields in engineering.

We next consider the *many-query context*, in which we make many (many) appeals to the input-output evaluation. One example of the many-query context, directly related to our discussion above, is *Robust* Assess, Predict, and Act scenarios: a parameter estimation, prognosis, and optimization framework in which we determine and subsequently accommodate *all possible model variations* — crack lengths, tumor dimensions, or heat transfer coefficients — consistent with (typically noisy or uncertain) experimental measurements of system and environmental conditions. This robust framework — a form of uncertainty quantification admittedly within a restrictive parametric context — requires extensive exploration of parameter space to determine and exploit appropriate "possibility regions"; we discuss this further in Part VIII.

A second important example of the many-query context is multiscale (temporal, spatial) and multiphysics "multimodels," in which behavior at a larger scale must "invoke" many spatial or temporal realizations of behavior at a smaller scale. Particular illustrative cases include stress intensity factor evaluation [7] within a crack fatigue growth model [61]; calculation of spatially varying cell properties [25, 29] within homogenization theory [24] predictions for macroscale composite properties; assembly and interaction of many similar building blocks [81] in large (e.g., cardio-vascular) biological networks; or molecular dynamics computations based on quantum-derived energies/forces [36]. In all these cases, the number of input-output evaluations is often measured in the tens of thousands.

Both the real-time and many-query contexts present a significant and often unsurmountable challenge to "classical" numerical techniques such as the finite element (FE) method. These contexts are often much better served by the reduced basis approximations and associated *a posteriori* error estimation techniques described in this book. Two important notes that the reader will soon appreciate.

First, the methods in this book do not replace, but rather build upon and are measured (as regards accuracy) relative to, a “truth” finite element approximation [21, 41, 125, 144, 159] — this is not an algorithmic competition, but rather an algorithmic collaboration. Second, the methods in this book are decidedly *ill-suited* for the “single-query” or “few-query” context.

Historical Perspective

The development of the Reduced Basis (RB) method can perhaps be viewed as a response to the imperatives described above. In particular, the two contexts described represent not only computational challenges, but also computational opportunities. We identify two key opportunities that can be gainfully exploited.

(I) In the parametric setting, we restrict our attention to a typically smooth and rather low-dimensional parametrically induced manifold: the set of fields engendered as the input varies over the parameter domain; in the case of single parameter, the parametrically induced manifold is a one-dimensional filament within the infinite dimensional space which characterizes *general* solutions to the PDE. Clearly, generic approximation spaces are unnecessarily rich and hence unnecessarily expensive within the parametric framework.

(II) In the real-time or many-query contexts, in which the premium is on *marginal cost* (or perhaps asymptotic average cost) per input-output evaluation, we can accept greatly increased pre-processing or “Offline” cost — not tolerable for a single or few evaluations — in exchange for greatly decreased “Online” (or deployed) cost for each new/additional input-output evaluation. Clearly, resource allocation typical for “single-query” investigations will be far from optimal for many-query and real-time exercises.

We shall view the development of RB methods in terms of these two opportunities.

As always, it is difficult to find all initial sources of a good idea, as good ideas tend to be prompted by common stimuli and to simultaneously occur to several investigators; hence we apologize for any omissions. Initial work on the Reduced Basis approximation — Galerkin projection on approximation spaces which focus (typically through Taylor expansions or "snapshots") on the low-dimensional parametrically induced manifold of opportunity (I) — grew out of two related streams of inquiry: from the need for more effective, and perhaps also more interactive, many-query design evaluation — [51] considers linear structural examples; and from the need for more efficient parameter continuation methods — [6, 99, 100, 102, 105, 106] consider nonlinear structural analysis problems. (Several modal analysis techniques from this era [95] are also closely related to RB notions.)

The ideas present in these early somewhat domain-specific contexts were soon extended to (*i*) general finite-dimensional systems as well as certain classes of PDEs (and ODEs) [19, 50, 76, 101, 107, 119, 130, 131], and (*ii*) a variety of different reduced basis approximation spaces — in particular Taylor and Lagrange [118] and more recently Hermite [66]. The next decade(s) saw further expansion into different applications and classes of equations, such as fluid dynamics and the Navier-Stokes equations [57, 65, 66, 67, 68, 112].

However, in these early methods, the approximation spaces tended to be rather local and typically rather low-dimensional in parameter (often a single parameter). In part, this was due to the nature of the applications — parametric continuation. But it was also due to the absence of *a posteriori* error estimators and effective sampling procedures. (In fairness, several early papers [103, 104, 105] did indeed discuss *a posteriori* error estimation and even adaptive improvement of the RB space; however, the approach could not be efficiently or rigorously applied to PDEs due to the computational requirements, the residual norms employed, and the absence of any stability considerations.) It is clear that in more global, higher-dimensional parameter domains the reduced basis predictions "far" from any sample points can not *a priori* be trusted, and hence *a posteriori* error estimators are crucial to reliability (and ultimately, safe engineering interventions *in particular in the real-time context*). It is equally clear that in more global, higher-dimensional — even three-dimensional — parameter domains simple tensor-product/factorial "designs" are not practicable, and hence sophisticated sampling strategies are crucial to efficiency.

Much of this book is devoted to (*i*) recent work on rigorous *a posteriori* error estimation and in particular error bounds for outputs of interest [87, 89, 120], and (*ii*) effective sampling strategies in particular for higher (than one) dimensional parameter domains [32, 98, 134, 149]. In fact, as we shall see, the former are a crucial ingredient in the latter — the inexpensive error bounds permit us first, to explore much larger subsets of the parameter domain in search of most representative or best "snapshots," and second, to know when we have *just enough* basis functions — and hence the simultaneous development of error estimation and sampling capabilities is not a coincidence. (We note that the greedy sampling methods described in this book are similar in objective to, but very different in approach from, more well-known Proper Orthogonal Decomposition (POD) methods [8, 23, 58, 75, 77, 97, 126, 127, 128, 142, 143, 156] typically applied in the temporal domain. However, POD economization techniques can be and have successfully been applied within the parametric RB context [31, 40, 43, 59, 86]. A brief comparative study is provided in Part I and again in Part IV.)

Early work certainly exploited the opportunity (II), but not fully. In particular, and perhaps at least partially because of the difficult nonlinear nature of the initial applications, early RB approaches did not fully decouple the underlying "truth" FEM approximation — of very high dimension $\mathcal{N}_{\mathrm{t}}$ — from the subsequent reduced basis projection and evaluation — of very low dimension N. More precisely, most often the Galerkin stiffness equations for the reduced basis system were generated by direct appeal to the high–dimensional "truth" representation: in nuts and bolts terms, pre- and post-multiplication of the "truth" stiffness system by rectangular basis matrices. As a result of this expensive projection the computational savings provided by RB treatment (relative to classical FEM "truth" evaluation) were typically rather modest [99, 118, 119] *despite* the very small size of the ultimate reduced basis stiffness system.

Much of this book is devoted to *full* decoupling of the "truth" and RB spaces through Offline-Online procedures: the complexity of the Offline stage depends on $\mathcal{N}_{\rm t}$ (the dimension of the "truth" finite element space); but the complexity of the Online stage — in which we respond to a new value of the input parameter — depends only on N (the dimension of the reduced basis space) and the parametric complexity of the operator and data. In essence, we are guaranteed the accuracy of a high-fidelity finite element model but at the very low cost of a reduced-order model. In the context of *affine parameter dependence*, in which the operator is expressible as the sum of products of parameter-dependent functions and parameter-independent operators, the Offline-Online idea is quite self-apparent and indeed has been re-invented often [15, 65, 70, 112]; however, application of the concept to *a posteriori* error estimation — note the Online complexity of both the output *and* the output error bound calculation must be independent of $\mathcal{N}_{\rm t}$ — is more involved and more recent [62, 120, 121]. In the case of *nonaffine parameter dependence* the development of Offline-Online strategies is much less transparent, and only in the last few years have effective procedures — in effect, efficient methods for approximate reduction to affine form — been established [18, 54, 135]. Clearly, Offline-Online procedures are a crucial ingredient in the real-time context.

We note that historically [50] and in this book RB methods have been built upon, and measured (as regards accuracy) relative to, *finite element* "truth" discretizations (or related spectral element approaches [81, 82, 83, 84, 111]) — the variational framework provides a very convenient setting for approximation and error estimation. However there are certainly many good reasons to consider alternative "truth" settings: a systematic finite volume framework for RB approximation and *a posteriori* error estimation is proposed and developed in [60]. We do note that boundary and integral approximations are less amenable to RB treatment or at least Offline-Online decompositions, as the inverse operator will typically not be affine in the parameter.

Scope and Roadmap

We begin with General Preliminaries. The purpose of the General Preliminaries is to recall — in a form relevant to the subsequent development — the background material on which the rest of the book shall rest. We discuss basic elements of functional analysis: Hilbert spaces (real and complex); product spaces; bases; inner products and norms; the Cauchy-Schwarz inequality; linear bounded forms and dual spaces; the Riez representation theorem; and finally bilinear forms. We review the fundamental properties associated with bilinear forms — the coercivity and inf-sup stability conditions [9] and the continuity condition — and introduce associated eigenproblems of computational relevance. And finally we introduce the basic smoothness hypotheses, function spaces, and norms associated with variational formulation and approximation of second order partial differential equations. In all cases we consider both the standard definition as well as the (in most cases, rather self-evident) extension to the parametric context of particular interest in this book.

Following the General Preliminaries each subsequent Part of this book addresses a different class of problems. In each case we first identify the abstract formulation of the problem and then develop particular instantiations (corresponding to particular equations/physical phenomena) and associated specific examples. We then proceed to the formulation and analysis: reduced basis approximation; optimal sampling procedures; *a priori* theory; rigorous *a posteriori* error estimation; and Offline-Online computational strategies. Finally, in each case we provide software in the form of MATLAB® `.m` files that, given an appropriate "truth" finite element model provided by the reader implements the methods developed.

In Part I of this book we treat the particularly simple case of Parametrically Coercive and Compliant Affine Linear Elliptic Problems. This class of problems — in which each term of the parametric development is independently symmetric positive semidefinite, and for which furthermore the load/source functional and output functional coincide — permits a simple exposition of the key ideas of the book: RB spaces and suitably orthogonalized bases; Galerkin projection and optimality; greedy quasi-exhaustive sampling procedures; the role of parametric smoothness in convergence; rigorous and relatively sharp *a posteriori* error estimation; and Offline-Online computational strategies. We illustrate this Part of the book with thermal conduction and linear elasticity examples that involves $O(10\text{–}20)$ independent parameters [137]. (The reader should guard against disappointment: it is really only for this simple class of problems, for which lower bounds for stability constants can be explicitly and readily extracted, that we can entertain so many parameters.)

In Part II of this book we consider the more general case of Coercive Affine Linear Elliptic Problems. We no longer require *parametric* coercivity; we now permit non-symmetric bilinear forms a; and we now consider arbitrary linear (bounded) output functionals — and perhaps multiple outputs. At this stage we can also treat, and we hence we introduce and exercise, the general class of piecewise-affine geometric and coefficient parametric variations consistent with the requirement of affine parameter dependence. Physical instantiations include general heat conduction (the Poisson equation) problems; forced-convection heat transfer (the convection-diffusion equation) problems; and general linear elasticity problems. In relation to Part I, the key new methodological elements are the development of (*i*) primal-dual (with adjoint) approximation [115] for RB [120], (*ii*) an *a posteriori* theory of general (non-compliant, and hence "non-energy") outputs [5, 22, 110] for RB [120, 139], and (*iii*) an efficient Offline-Online computational procedure for the construction of a lower bound for the general coercivity constant [62] required by our *a posteriori* estimators.

In Part III of this book we consider the most general case of *Non-Coercive* Affine Linear Elliptic Problems (we address the particular stability issues [26, 27] associated with Saddle Problems, in particular the Stokes equations of incompressible flow [133, 135, 136], in Part VI). Physical instantiations include the ubiquitous Helmholtz equation relevant to time-harmonic acoustics, elasticity, and electromagnetics. In this Part we also introduce a special formulation (perforce non-coercive) for *quadratic* output functionals [61] — important in such applications as acoustics and linear elastic fracture theory. In relation to Part II, the key new methodological elements are the development of (*i*) *discretely* stable primal-dual RB approximations [89], (*ii*) an efficient Offline-Online computational procedure [62, 139] for the construction of a lower bound for the general *inf-sup* constant [9] required by our *a posteriori* estimators. (The latter, in essence a lower bound for a *singular value*, demands considerable Offline resources and is certainly a limiting factor in the treatment of higher dimensional parameter spaces: an opportunity for further work.)

RB-like snapshot ideas (typically enhanced by sophisticated POD sampling variants) are also common in certain Reduced Order Model (ROM) approaches in the temporal domain [14, 38, 39, 93, 114, 129, 143, 151, 152]; more recently greedy sampling approaches have also been considerated in [20]. However, combined "parameter + time" approaches — essentially the marriage of ROM in time with

RB in parameter, and sometimes referred to as PROM (Parametric ROM) — are relatively uncommon [31, 40, 43, 59, 49, 86, 141]. In Part IV of this book we explore the "parameter + time" paradigm in the important context of Affine Linear Parabolic Problems such as the heat or diffusion equation and the passive scalar convection-diffusion equation (also the Black-Sholes equation of derivative theory [117]). In particular, in Part IV we extend to parabolic PDEs the primal-dual approximations, greedy sampling strategies, *a posteriori* error estimation concepts, and Offline-Online computational strategies developed in Part II for elliptic PDEs — with particular focus on the accommodation of an "evolution" parameter t [56, 60, 132]. Two qualifications: we restrict attention to discrete-time parabolic equations corresponding to simple Euler backward (or Crank-Nicolson) discretization of the original continuous PDE; and except briefly (where we permit a weaker Garding inequality) we only consider parabolic equations associated with *coercive* spatial operators.

In Part V of the book we consider the extension, in both the elliptic and parabolic cases, to nonaffine problems. The strategy is ostensibly simple: reduce the nonaffine operator and data to approximate affine form, and then apply the methods developed for affine operators in Parts II, III and IV. However, this reduction must be done efficiently in order to avoid a proliferation of parametric functions and a corresponding degradation of Online response time. The approach we describe here is based on the Empirical Interpolation Method (EIM) [18]. We first describe the Empirical Interpolation Method for efficient approximation of fields which depend (smoothly) on parameters: a collateral RB space for the offending nonaffine coefficient functions; an interpolation system that avoids costly ($\mathcal{N}_{\mathrm{t}}$-dependent) projections; and several (from less rigorous/simple to completely rigorous/quite cumbersome) *a posteriori* error estimators. We then apply the EIM within the context of RB treatment of elliptic and parabolic PDEs with nonaffine coefficient functions [54, 135, 145]; the resulting approximations preserve the usual Offline-Online efficiency — the complexity of the Online stage is independent of $\mathcal{N}_{\mathrm{t}}$.

In Part VI of the book we treat several elliptic problems with polynomial nonlinearities. Here our coverage is admittedly somewhat more anecdotal: we can not uphold our standards of rigor (in *a posteriori* error estimation) or efficiency (in Online requirements) for general nonlinear problems; we thus consider several representative examples that illustrate essential points. First [149], we consider an elliptic problem with stabilizing cubic nonlinearity [34] . This problem illustrates both the possibility and difficulty of efficient RB Galerkin approximation of (lowish-order) polynomial nonlinearities, and the availability in some very special circumstances of a very simple nonlinear *a posteriori* error theory. Second, we proceed to the (quadratically nonlinear) Navier-Stokes equations [27, 52, 57] of incompressible fluid flow; for simplicity we consider here only a single parameter, the Reynolds number. For the Navier-Stokes equations (and for nonlinear equations more generally) we can not appeal to any simple monotonicity arguments; our focus is thus on the computational (quantitative) realization of the general Brezzi-Rappaz-Raviart ("BRR") *a posteriori* theory [28, 34] — and development of associated sampling procedures — within the reduced basis Offline-Online context [98, 147, 148]. (We also address here the constuction of div-stable [26, 27] RB (Navier)-Stokes approximations [122, 136].) Third and finally, we consider symmetric eigenproblems associated with (say) the Laplacian [10] or linear elasticity operator: we present formulations for one or two lowest eigenvalues [87] and for the first "many" eigenvalues (as relevant in quantum chemistry [35, 36]). Here, implicitly, the interpretation of the BRR theory is unfortunately less compelling due to the (guaranteed!) multiplicity of often nearby solutions.

In Part VII we consider nonpolynomial nonlinearities (in the operator and also output functional) for both elliptic and parabolic PDEs. Our focus is on the application/extension of the Empirical Interpolation Method to this important class of problems [54]: in effect, we expand the nonlinearity in a collateral reduced basis expansion, the coefficients of which are then obtained by interpolation relative to the reduced basis approximation of the field variable. We are therefore able to maintain, albeit with some effort, our usual Offline-Online efficiency — Online evaluation of the output is independent of $\mathcal{N}_{\mathrm{t}}$. (For alternative approaches to nonlinearities in the ROM context, see [16, 39, 113].) Unfortunately, for this difficult class of problems we can not provide (efficient) rigorous *a posteriori* error estimators. (It it thus perhaps not surprising that initial work in RB methods [99, 102], focused on highly nonlinear problems, did not attempt complete Offline-Online decoupling or rigorous error estimation.) The trade-off between rigor and model complexity is inevitable; we hope the reader finds the methods of Part VI useful despite the lower standards of certainty.

In Part VIII we depart from our usual format and instead consider a real-time and many-query application of reduced basis approximation and *a posteriori* error estimation: robust parameter estimation for systems described by elliptic and parabolic PDEs — from outputs we wish to deduce inputs [55]. (Other applications of RB, in particular to optimization and control, can be found in [109, 123].) Our focus is on the rigorous incorporation of experimental error and numerical (RB) error bounds in the specification of "possibility regions": regions of the parameter domain consistent with available (noisy) experimental data. (For well-posed or "identifiable" [17] systems the possibility region will shrink to the unique value of the unknown parameter(s) as the experimental error and reduced basis error tend to zero. However, many interesting "systems" — which should be understood to comprise the model, the experimental measurements, the unknown inputs, and the selected outputs — are not identifiable.) In practice, except for special problems, these possibility regions can not be constructed except by truly exhaustive and exhausting (even at "reduced basis speed") calculation; we thus also consider various efficient procedures for approximating these possibility regions. We consider an example of transient thermal conduction inverse analysis.

Finally, in Part IX, we discuss briefly two topics on the research frontier. First, we shall consider the "reduced basis element method" [81, 82, 83, 84]: a marriage of reduced basis and domain decomposition concepts that permits much greater parametric complexity and also provides a framework for the integration of multiple models. Second, (at least linear) hyperbolic problems are also ripe for further development: although there are many issues related to smoothness, stability, and locality, there are also important proofs-of-concept [60, 111] in both the first order and second order contexts which demonstrate that RB approximation and *a posteriori* error estimation can be gainfully applied to hyperbolic equations. For both topics, we briefly review the current status and identify outstanding challenges.

Intended Audience

We have in mind four audiences. The first audience is professional researchers, faculty, and graduate students in the area of numerical methods for PDEs: *developers* (and analysts) of numerical methods. We hope that the formulations and theory summarized in our research monograph will provide a good foundation for further developments in reduced basis methodology and analysis. (We also hope that the book might be appropriate as a secondary source in a graduate course on numerical methods for PDEs: the RB framework is a very good laboratory in which to understand, exercise, and observe many basic aspects of computational approaches to PDEs.)

The second audience is computational engineers — professionals or graduate students for which application of computational methods for PDEs plays an essential role: advanced *users* of numerical methods. It is for this audience (and educators, see below) that we hope the software component of the book will prove useful: a rapid and easy way to apply the reduced basis approximation, greedy sampling, *a posteriori* error estimation, and Offline-Online approaches described in this book to problems of interest in the research, development, design/optimization, and "Assess-Predict-Act" contexts. We do provide one word of caution: the software we provide is blackbox (actually, only "somewhat" blackbox since the formulations and theory must be understood to properly invoke and connect the modules) once the "truth" finite element approximation is appropriately specified; however, the requisite finite element ingredients can not always be generated by a (third-party) FE program without access to the source code. Hence we implicitly assume that the computational engineer is willing and able to take screwdriver (though hopefully not jackhammer) in hand.

We note that FE packages oriented towards, or based upon, domain decomposition for definition of problem geometry and coefficients are particularly well suited to the reduced basis approach. Example of such packages are the MATLAB PDE Toolbox® or COMSOL Multiphysics$^{\text{TM}}$. In this case, it is possible to create the finite element inputs to the RB software without modification of, or *even access to*, the source code — the available assembly and export features suffice. We shall indicate on several occasions (with the MATLAB PDE Toolbox® as our vehicle) the simple and clean interface between a domain-decomposition "cognizant" finite element package and our own reduced basis software.

The third audience is university engineering educators (and ultimately, as *end users*, students). The application of finite element simulations for visualization, assessment of classical engineering models and approximations, and parameter estimation and design/optimization — both in class and as part of homework assignments and projects — has remained quite limited. Of course, complex user interfaces are part of the problem. But even more fundamental is the relatively slow response time of even very good codes: for an in-class demonstration, one minute or even 10 seconds for typically just a *single* parameter value is an eternity; and even for homework assignments, large operation counts and storage requirements can quickly obscure the pedagogical point. Clearly, this context can benefit from a real-time and many-query (many-student) perspective: in particular, we hope that, with the software we provide, educators can "automatically" and quickly develop *very fast* — and, thanks to the *a posteriori* error estimators, *reliable and physically relevant* — Online modules for visualization and input-output evaluation of complex problems. However, we do again caution that the professor — or able-bodied Teaching Assistant — must have access to the necessary finite element infrastructure in order to develop the "truth" prerequisities.

Our fourth audience is very broad: we hope that our text will become a "coffee-table" book. In this age of technology, all informed citizens should be acutely interested in reduced basis approximation and *a posteriori* error estimation for parametrized partial differential equations. Perhaps families can organize group readings so that both young and old can appreciate the content and implications. Or even better, perhaps each member of the family can purchase his or her own copy of the book to keep — or to give as thoughtful gifts.

Acknowledgements

This book discusses research that is not our own — and that preceded our own research by in some cases several decades — and also research that is our own. But "our own" is a misnomer, as our research project has been a collaborative effort with many colleagues. We give here thanks, we hope without any unintentional omissions.

First and foremost, we would like to acknowledge the many and important contributions of our longstanding co-conspirator, Yvon Maday. Our collective effort on reduced basis concepts (and on other ideas in earlier years) has been enjoyable and productive.

We would also like to acknowledge our very fruitful collaborations with Claude Le Bris, Analisa Buffa, Eric Cancès, Jan Hesthaven, Jaime Peraire, Christophe Prud'homme, Alfio Quarteroni, Einar Rønquist, Gabriel Turinici, and Karen Willcox. In many cases our joint activities have taken the form of co-supervision of talented students — a particularly enjoyable aspect of academia.

We would specifically like to thank the many MIT Masters and PhD students and post-docs, visiting students and post-docs, and also Singapore-MIT Alliance students, who have contributed to our reduced basis effort over the years: Shidrati Ali, ANG Wei Sin, Sebastien Boyaval, Luca Dedè, Simone Deparis, Revanth Garlapati, Ginger, Martin Grepl, Thomas Leurent, Alf Emil Løvgren, HUYNH Dinh Bao Phuong, Luc Machiels, NGUYEN Ngoc Cuong, Ivan Oliveira, George Pau, Jerónimo Rodríguez, Dimitrios Rovas, Sugata Sen, TAN Alex Yong Kwang, and Karen Veroy. A long alphabetical list of this variety perforce approaches anonymity, and hides the singular contributions of many of these very talented individuals. We would also like to acknowledge the contributions of Roberto Milani and Annalisa Quaini, Masters students at Politecnico di Milano.

We can not thank enough Ms Debra Blanchard for all of her efforts in manag-

ing our research projects and collaborations, and also, and in particular, for her many direct contributions to this book: for organizing our sometimes chaotic work flow; for performing all the mathematical typesetting with exacting standards; for preparing all the artwork and figures; and, occasionally, for spotting a mathematical error or two. Ms Blanchard is also responsible for the design and implementation of the website with which this book is associated.

ATP wishes to thank the Chief Technology Officer at his Lincoln office, Ms Anddie Chan, for designing, maintaining, and operating a computer system with all the latest information and document processing technology. This book owes much to Ms Chan's dedication, professionalism, and Victorian work ethic.

We would like to recognize that much of "our own" research reflected in these pages was supported by the US Defense Advanced Research Projects Agency, by the US Air Force Office of Scientific Research, and by the Singapore MIT-Alliance. We are also grateful to the "MIT Pappalardo Graduate Monographs in Mechanical Engineering" for supporting the development of this book and associated website and software.

ATP & GR, January 2007.

Contents

General Preliminaries

Notice

Chapter 1

Mathematical Foundations

The reader should review Sections 1.1, 1.2, and 1.4 prior to embarking upon Parts I and II, and additionally Sections 1.3 and 1.5 prior to continuing with Part III.

1.1 Inner Product Spaces

In this section we describe the basic properties of inner product spaces; for more elaboration on this material, see [2, 71, 79, 96, 108, 158].

1.1.1 Definition

We consider here real linear spaces (see Section 1.5 for the complex case). We first recall that a space Z is a (real) *linear* or *vector* space if, for any $\alpha \in \mathbb{R}$, $w, v \in Z$, $\alpha w + v$ is also an element of Z. Here $\mathbb{R}$ denotes the real numbers — and $\mathbb{R}_+$ and $\mathbb{R}_{+0}$ the positive and non-negative reals; $\mathbb{C}$ shall denote the complex numbers; and $\mathbb{N}_0$ and $\mathbb{N}$ shall denote the natural numbers including and not including zero. Note that the dimension of Z, which we denote $\dim(Z)$, can be either finite or infinite.

We recall (we restrict attention here to $\dim(Z)$ finite) that a basis set for a linear space Z is a set of (linearly independent) elements $z_i \in Z$, $1 \leq i \leq \dim(Z)$, such that for *all* w in Z

there exists a *unique* set of real numbers, $\omega_i \in \mathbb{R}$, $1 \leq i \leq \dim(Z)$, such that

$$w = \sum_{i=1}^{\dim(Z)} \omega_i z_i \ . \tag{1.1}$$

The choice of basis set is, of course, not unique. We can describe our space in terms of a(ny) basis set: $Z = \text{span}\{z_i,\ 1 \leq i \leq \dim(Z)\}$.

An (real) inner product space Z is a *linear space* equipped with an *inner product* $(w, v)_Z$, $\forall\, w, v \in Z$, and *induced norm* $\|w\|_Z = \sqrt{(w,w)_Z}$, $\forall\, w \in Z$. An inner product $w \in Z, v \in Z \rightarrow (w, v)_Z \in \mathbb{R}$ must satisfy several conditions: (bilinearity) for any $\alpha \in \mathbb{R}$, $w, v \in Z$, $z \in Z$, $(\alpha w + v, z)_Z = \alpha(w, z)_Z + (v, z)_Z$ and $(z, \alpha w + v)_Z = \alpha(z, w)_Z + (z, v)_Z$; (symmetry) for any $w, v \in Z$, $(w, v)_Z = (v, w)_Z$; and (positivity) for all $w \in Z$, $(w, w)_Z \geq 0$ with equality only for $w = 0$. (It directly follows from these conditions on the inner product that $\| \cdot \|_Z$ is a valid norm.) The Cauchy-Schwarz inequality,

$$|(w, v)_Z| \leq \|w\|_Z \|v\|_Z, \qquad \forall\, w, v \in Z, \tag{1.2}$$

is a direct consequence of the inner product definition.

We recall that a Hilbert space is a complete inner product space [2].

1.1.2 Cartesian Product Spaces

Given two inner product spaces Z_1 and Z_2, we define the Cartesian product of these spaces as $Z = Z_1 \times Z_2 \equiv \{(w_1, w_2) \mid w_1 \in Z_1, w_2 \in Z_2\}$. Given $w = (w_1, w_2) \in Z$, $v = (v_1, v_2) \in Z$, we define

$$w + v \equiv (w_1 + v_1, w_2 + v_2) \ ; \tag{1.3}$$

it directly follows that Z is a *linear* space. We further equip Z with the inner product

$$(w, v)_Z = (w_1, v_1)_{Z_1} + (w_2, v_2)_{Z_2} \tag{1.4}$$

and induced norm

$$\|w\|_Z = \sqrt{(w, w)_Z} \ ; \tag{1.5}$$

it is readily demonstrated that $(\,\cdot\,,\,\cdot\,)_Z$ is a valid inner product and hence Z an inner product space. (Note the choice of inner product is not unique.)

Given a basis set $\{z_{1\,i}\}$, $1 \le i \le \dim(Z_1)$, for Z_1 and a basis set $\{z_{2\,j}\}$, $1 \le j \le \dim(Z_2)$, for Z_2, we can readily construct a basis set for Z:

$$\begin{aligned} Z \;\; &= \;\; \operatorname{span}\{(z_{1\,i}, 0),\; 1 \le i \le \dim(Z_1)\;, \\ &\qquad\qquad (0, z_{2\,j}),\; 1 \le j \le \dim(Z_2)\}\;\;. \end{aligned} \tag{1.6}$$

It is clear from this identification that $\dim(Z) = \dim(Z_1) + \dim(Z_2)$.

1.2 Linear and Bilinear Forms

1.2.1 Linear Forms and Dual Spaces

A functional $g\colon Z \to \mathbb{R}$ is a *linear functional* or linear form if, for any $\alpha \in \mathbb{R}$, $w, v \in Z$, $g(\alpha w + v) = \alpha g(w) + g(v)$. A linear form is *bounded*, or continuous, over Z if

$$|g(v)| \le C\|v\|_Z, \qquad \forall\, v \in Z\;, \tag{1.7}$$

for some finite real constant C.

We define the *dual space* (to Z), Z', as the space of all linear bounded functionals over Z. We associate to Z' the norm

$$\|g\|_{Z'} = \sup_{v \in Z} \frac{g(v)}{\|v\|_Z}, \qquad \forall\, g \in Z'\;, \tag{1.8}$$

which we shall often denote the "dual norm." The Riesz representation theorem states that, for any $g \in Z'$, there exists a unique $w_g \in Z$ such that

$$(w_g, v)_Z = g(v), \qquad \forall\, v \in Z\;. \tag{1.9}$$

It is direct consequence of (1.9) — we simply insert (1.9) in (1.8) and invoke the Cauchy-Schwarz inequality (1.2) — that

$$\|g\|_{Z'} = \|w_g\|_Z\;. \tag{1.10}$$

This last result will be invoked extensively as a practical computational tool.

1.2.2 Bilinear Forms

A form b: $Z_1 \times Z_2 \to \mathbb{R}$ is a *bilinear form* if, for any $\alpha \in \mathbb{R}$, $w, v \in Z_1$, $z \in Z_2$, $b(\alpha w + v, z) = \alpha b(w, z) + b(v, z)$ and for any $\alpha \in \mathbb{R}$, $z \in Z_1$, $w, v \in Z_2$, $b(z, \alpha w + v) = \alpha b(z, w) + b(z, v)$; in short, a bilinear form is linear in each argument. In the remainder of this subsection we shall consider the particular (and particularly important) case in which $Z_1 = Z_2 = Z$. A bilinear form b: $Z \times Z \to \mathbb{R}$ is *symmetric* if, for any $w, v \in Z$, $b(w, v) = b(v, w)$. A bilinear form b: $Z \times Z \to \mathbb{R}$ is *skew-symmetric* if, for any $w, v \in Z$, $b(w, v) = -b(v, w)$ (note that for a real skew-symmetric bilinear form b, $b(w, w) = 0$). We define the symmetric and skew-symmetric parts of a general bilinear form b: $Z \times Z \to \mathbb{R}$ as $b_{\mathrm{S}}(w, v) = 1/2(b(w, v) + b(v, w))$, $\forall\, w, v \in Z$, and $b_{\mathrm{SS}}(w, v) = 1/2(b(w, v) - b(v, w))$, $\forall\, w, v, \in Z$, respectively.

A bilinear form b: $Z \times Z \to \mathbb{R}$ is *positive definite* if, for any $v \in Z$, $b(v, v)$ $(= b_{\mathrm{S}}(v, v)) \geq 0$ with equality only for $v = 0$; a bilinear form b: $Z \times Z \to \mathbb{R}$ is *positive semidefinite* if, for any $v \in Z$, $b(v, v) \geq 0$. It is clear that an inner product is simply a *symmetric positive-definite* (SPD) bilinear form.

We say that a bilinear form b: $Z \times Z \to \mathbb{R}$ is *coercive* over Z if

$$\alpha \equiv \inf_{w \in Z} \frac{b(w, w)}{\|w\|_Z^2} \tag{1.11}$$

is positive. Note that coercivity implicitly involves only the symmetric part of b — we can replace b in (1.11) with b_{S}. (Of course, many bilinear forms are not coercive: we consider the more general "inf-sup" stability condition in Section 1.3.) We say that a bilinear form b: $Z \times Z \to \mathbb{R}$ is *continuous* over Z if

$$\gamma \equiv \sup_{w \in Z} \sup_{v \in Z} \frac{b(w, v)}{\|w\|_Z \, \|v\|_Z} \tag{1.12}$$

is finite. For a coercive, continuous bilinear form α is denoted the coercivity constant and γ is denoted the continuity constant.

1.2.3 Parametric Linear and Bilinear Forms

We first introduce a closed bounded parameter domain $\mathcal{D} \subset \mathbb{R}^P$; a typical parameter vector, or P-tuple, in $\mathcal{D}$ shall be denoted $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P)$. We assume that $\mathcal{D}$ is suitably regular (for example, with Lipschitz continuous boundary), though in fact this assumption is typically not crucial.

We shall say that $g\colon Z \times \mathcal{D} \to \mathbb{R}$ is a *parametric linear form* if, for all $\boldsymbol{\mu} \in \mathcal{D}$, $g(\cdot; \boldsymbol{\mu}) : Z \to \mathbb{R}$ is a linear form. We say that a parametric linear form g is bounded (or continuous) if, for all $\boldsymbol{\mu} \in \mathcal{D}$, $g(\,\cdot\,; \boldsymbol{\mu}) \in Z'$. Note that the dual norm of a parametric linear form g, $\|g(\,\cdot\,; \boldsymbol{\mu})\|_{Z'}$, will of course be a (finite) function of $\boldsymbol{\mu}$ over $\mathcal{D}$.

Similarly, we shall say that $b\colon Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$ is a *parametric bilinear form* if, for all $\boldsymbol{\mu} \in \mathcal{D}$, $b(\,\cdot\,, \cdot\,; \boldsymbol{\mu})\colon Z_1 \times Z_2 \to \mathbb{R}$ is a bilinear form. In the remainder of this subsection we shall consider the case in which $Z_1 = Z_2 = Z$. We say that a parametric bilinear form $b\colon Z \times Z \to \mathbb{R}$ is *symmetric* if $b(w, v; \boldsymbol{\mu}) = b(v, w; \boldsymbol{\mu})$, $\forall\, w, v \in Z$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$. We define the symmetric part of a general parametric bilinear form $b\colon Z \times Z \times \mathcal{D} \to \mathbb{R}$ as $b_{\mathrm{S}}(w, v; \boldsymbol{\mu}) \equiv 1/2(b(w, v; \boldsymbol{\mu}) + b(v, w; \boldsymbol{\mu}))$, $\forall\, w, v \in Z$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$.

We say a parametric bilinear form $b\colon Z \times Z \times \mathcal{D} \to \mathbb{R}$ is coercive over Z if

$$\alpha(\boldsymbol{\mu}) \equiv \inf_{w \in Z} \frac{b(w, w; \boldsymbol{\mu})}{\|w\|_Z^2} \tag{1.13}$$

is positive for all $\boldsymbol{\mu} \in \mathcal{D}$; we can then define $(0 <)\ \alpha_0 \equiv \min_{\boldsymbol{\mu} \in \mathcal{D}} \alpha(\boldsymbol{\mu})$. (Recall that $\mathcal{D}$ is closed.) We say a parametric bilinear form $b\colon Z \times Z \times \mathcal{D} \to \mathbb{R}$ is continuous over Z if

$$\gamma(\boldsymbol{\mu}) \equiv \sup_{w \in Z} \sup_{v \in Z} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_Z \|v\|_Z} \tag{1.14}$$

is finite for all $\boldsymbol{\mu} \in \mathcal{D}$; we can then define $\gamma_0 = \max_{\boldsymbol{\mu} \in \mathcal{D}} \gamma(\boldsymbol{\mu})$ $(< \infty)$.

1.2.4 A Coercivity Eigenproblem

We first observe that we can rewrite (1.13) as

$$\alpha(\boldsymbol{\mu}) \equiv \inf_{w \in Z} \frac{b_{\mathrm{S}}(w, w; \boldsymbol{\mu})}{\|w\|_Z^2} \ . \tag{1.15}$$

It directly follows from the Rayleigh quotient (1.15) that $\alpha(\boldsymbol{\mu})$ can be expressed as a minimum eigenvalue. For simplicity of exposition we shall consider the case in which $\dim(Z)$ is finite.

We now introduce this "coercivity" symmetric (generalized) eigenproblem associated with the parametric bilinear form b: $Z \times Z \times \mathcal{D} \to \mathbb{R}$: Given $\boldsymbol{\mu} \in \mathcal{D}$, find $(\chi^{\mathrm{co}}, \nu^{\mathrm{co}})_i(\boldsymbol{\mu}) \in Z \times \mathbb{R}$, $i = 1, \ldots, \dim(Z)$, such that

$$b_{\mathrm{S}}(\chi_i^{\mathrm{co}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \nu_i^{\mathrm{co}}(\boldsymbol{\mu})(\chi_i^{\mathrm{co}}(\boldsymbol{\mu}), v)_Z, \qquad \forall\, v \in Z \ , \tag{1.16}$$

and

$$\|\chi_i^{\mathrm{co}}(\boldsymbol{\mu})\|_Z = 1 \ ; \tag{1.17}$$

recall that b_{S} is the symmetric part of b. We order the eigenvalues in ascending order such that $\nu_1^{\mathrm{co}}(\boldsymbol{\mu}) \leq \nu_2^{\mathrm{co}}(\boldsymbol{\mu}) \leq \ldots \leq \nu_{\dim(Z)}^{\mathrm{co}}(\boldsymbol{\mu})$.

It follows directly from (1.15) and (1.16) that if a is coercive then $\alpha(\boldsymbol{\mu}) = \nu_1^{\mathrm{co}}(\boldsymbol{\mu}) > 0$.

1.2.5 Affine Parameter Dependence

We shall say that the parametric bounded linear form g: $Z \times \mathcal{D} \to \mathbb{R}$ is *affine* in the parameter if

$$g(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \Theta_g^q(\boldsymbol{\mu})\, g^q(v), \qquad \forall\, v \in Z \ , \tag{1.18}$$

for some finite Q_g; here the Θ_g^q: $\mathcal{D} \to \mathbb{R}$, $1 \leq q \leq Q_g$, are (typically very smooth) parameter-dependent functions, and the $g^q(v)$: $Z \to \mathbb{R}$, $1 \leq q \leq Q_g$, are parameter-independent bounded linear forms. (The term "affine in the parameter" is somewhat of a misnomer, since (1.18) really constitutes "affine in functions of the parameter"; for brevity, however, we shall abbreviate the latter by the former.)

Similarly, we shall say that the parametric bilinear form b: $Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$ is *affine* in the parameter if

$$b(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu})\, b^q(w, v), \qquad \forall\, w \in Z_1,\ \forall\, v \in Z_2\ , \tag{1.19}$$

for some finite Q_b; here the Θ_b^q: $\mathcal{D} \to \mathbb{R}$, $1 \leq q \leq Q_b$, are (typically very smooth) parameter-dependent functions, and the $a^q(w, v)$: $Z_1 \times Z_2 \to \mathbb{R}$, $1 \leq q \leq Q_b$, are parameter-independent continuous bilinear forms. (Note if b: $Z \times Z \times \mathcal{D}$ is symmetric, we assume that each b^q, $1 \leq q \leq Q_b$, is symmetric.) We remark that the representations (1.18),(1.19) are clearly not unique, though in general there will be minimum Q_g, Q_b for which such an expansion exists.

1.2.6 Parametric Coercivity

We say that an affine parametric (of necessity, coercive) form b: $Z \times Z \times \mathcal{D} \to \mathbb{R}$,

$$b(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu})\, b^q(w, v) \tag{1.20}$$

is "*parametrically* coercive" if $c \equiv b_{\mathrm{S}}$ (the symmetric part of b) admits an affine development

$$c(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \Theta_c^q(\boldsymbol{\mu}) c^q(w, v), \qquad \forall\, w, v \in Z\ , \tag{1.21}$$

that satisfies two conditions:

$$\Theta_c^q(\boldsymbol{\mu}) > 0, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D},\ 1 \leq q \leq Q_c\ , \tag{1.22}$$

and

$$c^q(v, v) \geq 0, \qquad \forall\, v \in Z,\ 1 \leq q \leq Q_c\ . \tag{1.23}$$

(Note we shall also suppose that each of the c^q, $1 \leq q \leq Q_c$, is symmetric.)

We note that if b is affine then certainly (1.21) exists for the choice $Q_c = Q_b$ and $c^q = b_{\mathrm{S}}^q$, $1 \leq q \leq Q_b$. However, this identification is overly restrictive: skew-symmetric components of b in (1.20) need not appear in the expansion of c, (1.21), and hence the parametric dependence of the skew-symmetric components of b need not honor (1.22). This generalization is important in applications such as convection-diffusion in both the elliptic and parabolic contexts.

1.3 The Inf-Sup Stability "Constant"

1.3.1 Definition

Given a parametric bilinear form b: $Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$, we define the inf-sup constant [9] as

$$\beta(\boldsymbol{\mu}) = \inf_{w \in Z_1} \sup_{v \in Z_2} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_{Z_1} \, \|v\|_{Z_2}} \ . \tag{1.24}$$

Here Z_1 and Z_2 are Hilbert spaces with associated inner products and induced norms $(\,\cdot\,,\,\cdot\,)_{Z_1}$, $\|\,\cdot\,\|_{Z_1}$ and $(\,\cdot\,,\,\cdot\,)_{Z_2}$, $\|\,\cdot\,\|_{Z_2}$, respectively. Both the case $Z_1 = Z_2 = Z$ and $Z_1 \neq Z_2$ will be of interest. (We note that for the special case of saddleproblems [26, 27], the inf-sup constant is defined in a different fashion; we introduce the necessary technology "*in vivo*" in Part VI.)

Clearly, $\beta(\boldsymbol{\mu}) \geq 0$ — a claim to the contrary is readily refuted by changing the sign of the allegedly supremizing v and invoking bilinearity. However, in general we can not assume that $\beta(\boldsymbol{\mu})$ is strictly positive. If there *does* exist a positive β_0 such that

$$\beta(\boldsymbol{\mu}) \geq \beta_0, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ , \tag{1.25}$$

then we shall say that b is "inf-sup stable" over $Z_1 \times Z_2$ (or, if $Z_1 = Z_2 = Z$, over Z).

1.3.2 Supremizing Operator

We next introduce the supremizing operator $T_{\boldsymbol{\mu}}$: $Z_1 \to Z_2$ associated with b, defined as

$$T_{\boldsymbol{\mu}} w = \arg \sup_{v \in Z_2} \frac{b(w, v; \boldsymbol{\mu})}{\|v\|_{Z_2}} \ . \tag{1.26}$$

It is a simple matter to obtain an explicit representation for $T_{\boldsymbol{\mu}}$: for any $w \in Z_1$,

$$(T_{\boldsymbol{\mu}} w, v)_{Z_2} = b(w, v; \boldsymbol{\mu}), \qquad \forall\, v \in Z_2 \ ; \tag{1.27}$$

we observe that $T_{\boldsymbol{\mu}}$ is linear. The equivalence between (1.26) and (1.27) is a direct consequence of the Cauchy-Schwarz inequality.

For the case of greatest interest in this book, in which b admits an affine representation (1.19), we may express $T_{\boldsymbol{\mu}} w$ as

$$T_{\boldsymbol{\mu}} w = \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu})\, \mathbb{T}^q w \ , \tag{1.28}$$

where the parameter-independent operators $\mathbb{T}^q \colon Z_1 \to Z_2$ are given by

$$(\mathbb{T}^q w, v)_{Z_2} = b^q(w, v), \qquad \forall\, v \in Z_2,\ 1 \leq q \leq Q_b \ . \tag{1.29}$$

The proof of (1.28) is simple: for any $w \in Z_1$,

$$\begin{aligned} \left(\sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu})\, \mathbb{T}^q w, v \right)_{Z_2} &= \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu}) (\mathbb{T}^q w, v)_{Z_2} \\ &= \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu})\, b^q(w, v) \\ &= b(w, v; \boldsymbol{\mu}) \\ &= (T_{\boldsymbol{\mu}} w, v)_{Z_2} \qquad \forall\, v \in Z_2 \ . \end{aligned} \tag{1.30}$$

This decomposition shall prove useful in developing inf-sup lower bounds.

1.3.3 Alternative Expressions for the Inf-Sup Constant

It follows from (1.24) and (1.26), (1.27) that we may also express $\beta(\boldsymbol{\mu})$ as

$$\begin{aligned} \beta(\boldsymbol{\mu}) &= \inf_{w \in Z_1} \sup_{v \in Z_2} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_{Z_1}\, \|v\|_{Z_2}} \\ &= \inf_{w \in Z_1} \frac{b(w, T_{\boldsymbol{\mu}} w; \boldsymbol{\mu})}{\|w\|_{Z_1}\, \|T_{\boldsymbol{\mu}} w\|_{Z_2}} \\ &= \inf_{w \in Z_1} \frac{(T_{\boldsymbol{\mu}} w, T_{\boldsymbol{\mu}} w)_{Z_2}}{\|w\|_{Z_1}\, \|T_{\boldsymbol{\mu}} w\|_{Z_2}} \\ &= \inf_{w \in Z_1} \frac{\|T_{\boldsymbol{\mu}} w\|_{Z_2}}{\|w\|_{Z_1}} \ ; \end{aligned} \tag{1.31}$$

in the case $Z_1 = Z_2 = Z$ we obtain

$$\beta(\boldsymbol{\mu}) = \inf_{w \in Z} \frac{\|T_{\boldsymbol{\mu}} w\|_Z}{\|w\|_Z} \ . \tag{1.32}$$

It follows from the Rayleigh(-like) quotients (1.31) and (1.32) that $\beta(\boldsymbol{\mu})$ can be readily expressed in terms of an eigenproblem (see Section 1.3.5).

We conclude here with another useful articulation of the inf-sup constant: for any $w \in Z_1$, there exists a $v \in Z_2$ such that

$$\beta(\boldsymbol{\mu})\, \|w\|_{Z_1}\, \|v\|_{Z_2} \leq b(w, v; \boldsymbol{\mu}) \ . \tag{1.33}$$

We thus observe that the inf-sup construct is a fashion by which to find energetically "good" test functions that ensure positivity. It is simple to demonstrate that (1.24) implies (1.33) with $v = T_{\boldsymbol{\mu}} w$:

$$b(w, T_{\boldsymbol{\mu}} w; \boldsymbol{\mu}) \geq \beta(\boldsymbol{\mu})\, \|w\|_{Z_1}\, \|T_{\boldsymbol{\mu}} w\|_{Z_2}, \qquad \forall\, w \in Z_1 \ . \tag{1.34}$$

This will prove quite useful in *a priori* error analysis.

1.3.4 Alternative Expressions for the Continuity Constant

In fact, we can also express our previously introduced continuity constant (here generalized to $Z_1 \neq Z_2$),

$$\gamma(\boldsymbol{\mu}) = \sup_{w \in Z_1} \sup_{v \in Z_2} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_{Z_1}\, \|v\|_{Z_2}} \ , \tag{1.35}$$

in terms of the $T_{\boldsymbol{\mu}}$ operator. In particular, it follows from (1.26), (1.27), and (1.35) that

$$\begin{aligned} \gamma(\boldsymbol{\mu}) &= \sup_{w \in Z_1} \frac{b(w, T_{\boldsymbol{\mu}} w)}{\|w\|_{Z_1}\, \|T_{\boldsymbol{\mu}} w\|_{Z_2}} \\ &= \sup_{w \in Z_1} \frac{(T_{\boldsymbol{\mu}} w, T_{\boldsymbol{\mu}} w)_{Z_2}}{\|w\|_{Z_2}\, \|T_{\boldsymbol{\mu}} w\|_{Z_2}} \\ &= \sup_{w \in Z_1} \frac{\|T_{\boldsymbol{\mu}} w\|_{Z_2}}{\|w\|_{Z_2}} \ ; \end{aligned} \tag{1.36}$$

for $Z_1 = Z_2 = Z$,

$$\gamma(\boldsymbol{\mu}) = \sup_{w \in Z} \frac{\|T_{\boldsymbol{\mu}} w\|_Z}{\|w\|_Z} \ . \tag{1.37}$$

The $\beta(\boldsymbol{\mu})$ and $\gamma(\boldsymbol{\mu})$ are thus *both* related to the same associated eigenproblem. We now discuss this eigenproblem.

1.3.5 Inf-Sup (and Continuity) Eigenproblem

We conclude from (1.31) and (1.36) that

$$\beta^2(\boldsymbol{\mu}) = \inf_{w \in Z_1} \frac{\|T_{\boldsymbol{\mu}} w\|^2_{Z_2}}{\|w\|^2_{Z_1}} , \tag{1.38}$$

$$\gamma^2(\boldsymbol{\mu}) = \sup_{w \in Z_1} \frac{\|T_{\boldsymbol{\mu}} w\|^2_{Z_2}}{\|w\|^2_{Z_1}} , \tag{1.39}$$

and hence that $\beta^2(\boldsymbol{\mu})$ and $\gamma^2(\boldsymbol{\mu})$ are the minimum and maximum of a Rayleigh quotient. We can thus identify an associated eigenproblem. For simplicity of exposition we shall consider the case in which $\dim(Z_1)$ and $\dim(Z_2)$ are finite.

We introduce the "inf-sup" symmetric positive semidefinite (generalized) eigenproblem associated with a parametric bilinear form b: $Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$: Given $\boldsymbol{\mu} \in \mathcal{D}$, find $(\chi, \nu)_i(\boldsymbol{\mu}) \in Z_1 \times \mathbb{R}_{+0}$, $i = 1, \ldots, \dim(Z_1)$, such that

$$(T_{\boldsymbol{\mu}} \chi_i(\mu), T_{\boldsymbol{\mu}} w)_{Z_2} = \nu_i(\boldsymbol{\mu})(\chi_i(\boldsymbol{\mu}), w)_{Z_1}, \qquad \forall\, w \in Z_1 , \tag{1.40}$$

and

$$\|\chi_i(\boldsymbol{\mu})\|_{Z_1} = 1 , \tag{1.41}$$

where $T_{\boldsymbol{\mu}} w$ satisfies (1.27). We order the eigenvalues in ascending order such that $0 \le \nu_1(\boldsymbol{\mu}) \le \nu_2(\boldsymbol{\mu}) \le \ldots \le \nu_{\dim(Z_1)}(\boldsymbol{\mu})$ (recall that $\mathbb{R}_+$ denotes the positive reals and $\mathbb{R}_{+0}$ the non-negative reals). We recall the usual orthogonality relations: $(T_{\boldsymbol{\mu}}\chi_i, T_{\boldsymbol{\mu}}\chi_j)_{Z_2} = \nu_i(\boldsymbol{\mu})\delta_{i\,j}$, $(\chi_i(\boldsymbol{\mu}), \chi_j(\boldsymbol{\mu}))_{Z_1} = \delta_{i\,j}$, $1 \le i, j \le \dim(Z_1)$; here $\delta_{i\,j}$ is the Kronecker-delta symbol.

It follows directly from (1.40), (1.31), and (1.36) that

$$\beta(\boldsymbol{\mu}) = \sqrt{\nu_1(\boldsymbol{\mu})} \tag{1.42}$$

and

$$\gamma(\boldsymbol{\mu}) = \sqrt{\nu_{\dim(Z_1)}(\boldsymbol{\mu})} , \tag{1.43}$$

corresponding to the square root of the extreme eigenvalues. We can further conclude that the infimizer of (1.31) is $\chi_1(\boldsymbol{\mu})$, and that the associated "inner" supremizer is $T_{\boldsymbol{\mu}}\chi_1(\boldsymbol{\mu})$: $\beta^2(\boldsymbol{\mu}) = b(\chi_1(\boldsymbol{\mu}), T_{\boldsymbol{\mu}}\chi_1(\boldsymbol{\mu}); \boldsymbol{\mu})$.

We pause to note that the inf-sup parameter is nothing more than a slightly generalized smallest singular value — a singular value with "attitude." In particular, if $Z_1 \equiv \mathbb{R}^m$ and $Z_2 \equiv \mathbb{R}^n$ with $(\,\cdot\,,\,\cdot\,)_{Z_1}$, $\|\cdot\|_{Z_1}$ and $(\,\cdot\,,\,\cdot\,)_{Z_2}$, $\|\cdot\|_{Z_2}$ the usual Euclidean inner products/norms, and $b(\,\cdot\,,\,\cdot\,;\boldsymbol{\mu})$ is the inner product associated with a matrix $\underline{B}(\boldsymbol{\mu}) \in \mathbb{R}^{n\times m}$, then

$$\beta(\boldsymbol{\mu}) = \inf_{w\in Z_1} \sup_{v\in Z_2} \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m} v_i\, B_{ij}(\boldsymbol{\mu})\, w_j}{\left(\sum\limits_{j=1}^{m} w_j^2\right)^{1/2}\left(\sum\limits_{i=1}^{n} v_i^2\right)^{1/2}} \tag{1.44}$$

is simply the smallest singular value of $\underline{B}(\boldsymbol{\mu})$. (To demonstrate this, we need only note from (1.44) that $(T_{\boldsymbol{\mu}}w)_i = \sum_{j=1}^m B_{ij}(\boldsymbol{\mu})\, w_j$, and hence that the $\nu_i(\boldsymbol{\mu})$ of (1.36) are the eigenvalues of $\underline{B}^{\mathrm{T}}(\boldsymbol{\mu})\underline{B}(\boldsymbol{\mu})$; here $^{\mathrm{T}}$ denotes algebraic transpose.) All our "singular value"/inf-sup results have simple and obvious analogies in the usual linear algebra context [146].

1.3.6 The Coercive Case Revisited

We consider here the relationship between the inf-sup constant and the coercivity constant in the case in which b: $Z \times Z \times \mathcal{D} \to \mathbb{R}$ is coercive — and hence both $\beta(\boldsymbol{\mu})$ and $\alpha(\boldsymbol{\mu})$ are positive and relevant. We immediately note from (1.24) for the choice $v = w$ and (1.13) that

$$\beta(\boldsymbol{\mu}) \geq \alpha(\boldsymbol{\mu}) \ , \tag{1.45}$$

since $v = w$ can never "do better" than the supremizer $v = T_{\boldsymbol{\mu}}w$.

In general, $\beta(\boldsymbol{\mu}) \neq \alpha(\boldsymbol{\mu})$. However, in the case in which b is *symmetric*, $\beta(\boldsymbol{\mu}) = \alpha(\boldsymbol{\mu})$. We present the proof in the case in which $\dim(Z)$ is finite: $\forall\, w \in Z$, $(T_{\boldsymbol{\mu}}\chi_1^{\mathrm{co}}(\boldsymbol{\mu}), T_{\boldsymbol{\mu}}w)_Z = b_{\mathrm{S}}(\chi_1^{\mathrm{co}}(\boldsymbol{\mu}), T_{\boldsymbol{\mu}}w; \boldsymbol{\mu}) = \nu_1^{\mathrm{co}}(\chi_1^{\mathrm{co}}, T_{\boldsymbol{\mu}}w)_Z = \nu_1^{\mathrm{co}}\, b_{\mathrm{S}}(\chi_1^{\mathrm{co}}, w; \boldsymbol{\mu}) = (\nu_1^{\mathrm{co}})^2(\chi_1^{\mathrm{co}}, w)_Z$; hence $\nu_j(\boldsymbol{\mu}) = (\nu_1^{\mathrm{co}})^2(\boldsymbol{\mu})$ for some j, and thus $\beta^2(\boldsymbol{\mu}) = \nu_1(\boldsymbol{\mu}) \leq (\nu_1^{\mathrm{co}}(\boldsymbol{\mu}))^2$; therefore, since b is coercive and thus $\nu_1^{\rm co}(\boldsymbol{\mu}) > 0$, $\beta(\boldsymbol{\mu}) \le \alpha(\boldsymbol{\mu})$; but from (1.45) we must obtain equality $\beta(\boldsymbol{\mu}) = \alpha(\boldsymbol{\mu})$. (Note in the symmetric non-coercive case, we obtain $\beta(\boldsymbol{\mu}) = \min_{j \in 1,\ldots,\dim(Z)} |\nu_j^{\rm co}(\boldsymbol{\mu})|$, as expected from linear algebra.)

1.3.7 Adjoint Operators

We introduce here the operator $T_{\boldsymbol{\mu}}^\dagger$: $Z_2 \to Z_1$ associated with b: $Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$ as

$$T_{\boldsymbol{\mu}}^\dagger v = \arg \sup_{w \in Z_1} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_{Z_1}} \ . \tag{1.46}$$

It is readily shown that

$$(T_{\boldsymbol{\mu}}^\dagger v, w)_{Z_1} = b(w, v; \boldsymbol{\mu}), \qquad \forall\, w \in Z_1 \ ; \tag{1.47}$$

we observe that $T_{\boldsymbol{\mu}}^\dagger$ is linear. It follows directly from (1.27) and (1.47) that

$$(T_{\boldsymbol{\mu}}^\dagger v, w)_{Z_1} = (T_{\boldsymbol{\mu}} w, v)_{Z_2}, \qquad \forall\, w \in Z_1,\ \forall\, v \in Z_2 \ , \tag{1.48}$$

hence the adjoint † notation. Note if b is symmetric, $T_{\boldsymbol{\mu}}^\dagger = T_{\boldsymbol{\mu}}$.

We next define

$$\beta^\dagger(\boldsymbol{\mu}) = \inf_{v \in Z_2} \sup_{w \in Z_1} \frac{b(w, v; \boldsymbol{\mu})}{\|v\|_{Z_2}\, \|w\|_{Z_1}} \ ; \tag{1.49}$$

equivalently,

$$\beta^\dagger(\boldsymbol{\mu}) = \inf_{v \in Z_2} \frac{\|T_{\boldsymbol{\mu}}^\dagger v\|_{Z_1}}{\|v\|_{Z_2}} \ ; \tag{1.50}$$

equivalently, for all $v \in Z_2$ (there exists $w = T_{\boldsymbol{\mu}}^\dagger v$ such that)

$$\beta^\dagger(\boldsymbol{\mu})\, \|v\|_{Z_2}\, \|T_{\boldsymbol{\mu}}^\dagger v\|_{Z_1} \le b(T_{\boldsymbol{\mu}}^\dagger v, v; \boldsymbol{\mu}) \ . \tag{1.51}$$

These relations are relevant to the analysis of *adjoint* problems.

We can also define

$$\gamma^\dagger(\boldsymbol{\mu}) = \sup_{v \in Z_2} \sup_{w \in Z_1} \frac{b(w, v; \boldsymbol{\mu})}{\|v\|_{Z_2}\, \|w\|_{Z_1}} \ ; \tag{1.52}$$

equivalently,

$$\gamma^\dagger(\boldsymbol{\mu}) = \sup_{v \in Z_2} \frac{\|T^\dagger_{\boldsymbol{\mu}} v\|_{Z_1}}{\|v\|_{Z_2}} \ . \tag{1.53}$$

Here $\gamma^\dagger(\boldsymbol{\mu})$ is the adjoint continuity constant.

The case of interest in this book is

$$\dim(Z_1) = \dim(Z_2) \text{ and } \textit{finite} \ ; \tag{1.54}$$

note if $\dim(Z_1) \neq \dim(Z_2)$ then perforce either $\beta(\boldsymbol{\mu})$ or $\beta^\dagger(\boldsymbol{\mu})$ vanishes. Under the hypotheses (1.54) and $\beta(\boldsymbol{\mu}) > 0$ we can readily demonstrate [45] that

$$\beta^\dagger(\boldsymbol{\mu}) = \beta(\boldsymbol{\mu}), \quad \gamma^\dagger(\boldsymbol{\mu}) = \gamma(\boldsymbol{\mu}) \ . \tag{1.55}$$

We sketch here a proof.

We first note from our eigenproblem (1.40) and relation (1.48) that, $\forall v \in Z_2$, $(T^\dagger_{\boldsymbol{\mu}}(T_{\boldsymbol{\mu}}\chi_i(\boldsymbol{\mu})), T^\dagger_{\boldsymbol{\mu}} v)_{Z_1} = (T_{\boldsymbol{\mu}}\chi_i(\boldsymbol{\mu}), T_{\boldsymbol{\mu}}(T^\dagger_{\boldsymbol{\mu}} v))_{Z_2} = \nu_i(\boldsymbol{\mu})(\chi_i(\boldsymbol{\mu}), T^\dagger_{\boldsymbol{\mu}} v)_{Z_1} = \nu_i(\boldsymbol{\mu})(T_{\boldsymbol{\mu}}\chi_i(\boldsymbol{\mu}), v)_{Z_2}$, $1 \le i \le \dim(Z_1)$; it thus follows that the eigenproblem associated with the Rayleigh(-like) quotients (1.50), (1.53) has the $\dim(Z_1)$ — and since $\dim(Z_2) = \dim(Z_1)$, *only* the $\dim(Z_1)$ — eigenpairs $(T_{\boldsymbol{\mu}}\chi_i(\boldsymbol{\mu})/\sqrt{\nu_i(\boldsymbol{\mu})}, \nu_i(\boldsymbol{\mu}))$, $1 \le i \le \dim(Z_1)$ (note if $\beta(\boldsymbol{\mu}) > 0$ then $\nu_i > 0$, $1 \le i \le \dim(Z_1)$); hence $\beta^\dagger(\boldsymbol{\mu}) = \sqrt{\nu_1(\boldsymbol{\mu})} = \beta(\boldsymbol{\mu})$ and $\gamma^\dagger(\boldsymbol{\mu}) = \sqrt{\nu_{\dim(Z_1)}(\boldsymbol{\mu})} = \gamma(\boldsymbol{\mu})$. We also note for future reference that the infimizer in (1.49) is $T_{\boldsymbol{\mu}}\chi_1(\boldsymbol{\mu}) \in Z_2$ with associated supremizer $T^\dagger_{\boldsymbol{\mu}} T_{\boldsymbol{\mu}}\chi_1(\boldsymbol{\mu}) \in Z_1$. (In the symmetric case, from the arguments presented in Section 1.3.6, we know that $T_{\boldsymbol{\mu}}\chi_1$ is in fact colinear with χ_1.)

1.4 Classes of Functions

1.4.1 Field Variables

Scalar and Vector Fields

We first introduce an open bounded domain $\Omega \in \mathbb{R}^d$, $d = 1, 2$, or 3; we shall refer to a typical point in Ω as $\boldsymbol{x} = (x_1, \ldots, x_d)$. We define the canonical basis vectors as $\boldsymbol{e}_i$, $1 \le i \le d$, where (say, for $d = 2$) $\boldsymbol{e}_1 = (1, 0)$, $\boldsymbol{e}_2 = (0, 1)$. (In general the default, if we do not indicate otherwise, shall be two spatial dimensions: $d = 2$.) We consider Lebesgue measure and integration in Ω. We denote the boundary of Ω by $\partial\Omega$; we shall assume that $\partial\Omega$ is (except possibly for the cameo appearance of a well-behaved crack) Lipschitz continuous.

We shall consider both (here, real) scalar-valued and vector-valued field variables; examples of the former include temperature or pressure, while examples of the latter include displacement or velocity. We shall denote by d_v the dimension of the field variable: for scalar valued fields, $d_v = 1$, while for vector-valued fields, $d_v = d$. Thus, given d_v, a typical field variable will be denoted w: $\Omega \to \mathbb{R}^{d_v}$ and written as $w(x) = (w_1(x), \ldots, w_{d_v}(x))$; for convenience of exposition, we shall permit both $w(x)$ and $w_1(x)$ to represent a *scalar* field ($d_v = 1$).

We shall denote a multi-index spatial derivate [158] of a scalar (or one component of a vector) field w as

$$(D^{\boldsymbol{\sigma}} w)(\boldsymbol{x}) = \frac{\partial^{\boldsymbol{\sigma}} w}{\partial x_1^{\sigma_1} \cdots \partial x_d^{\sigma_d}} \ , \tag{1.56}$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$ is an index vector of non-negative integers $\sigma_1, \ldots, \sigma_d$; we denote by $|\boldsymbol{\sigma}| = \sum_{j=1}^{d} \sigma_j$ the *order* of the derivative. We also introduce $I^{d,n}$ as the set of all index vectors $\boldsymbol{\sigma} \in \mathbb{N}_0^d$ such that $|\boldsymbol{\sigma}| \leq n$. (Recall that $\mathbb{N}$ and $\mathbb{N}_0$ shall denote the natural numbers excluding and including zero.) We shall invoke the multi-index notation primarily for $|\boldsymbol{\sigma}| \geq 2$.

Function Spaces

We first introduce the space of continuous functions over $\Omega \subset \mathbb{R}^d$, $C^0(\Omega)$. We can then introduce the spaces $C^m(\Omega)$, $m \in \mathbb{N}_0$,

$$C^m(\Omega) \equiv \{w \mid D^{\boldsymbol{\sigma}} w \in C^0(\Omega), \ \forall \, \boldsymbol{\sigma} \in I^{d,m}\} \ ; \tag{1.57}$$

$C^m(\Omega)$ is the space of functions for which all derivatives $D^{\boldsymbol{\sigma}} w$ of order $|\boldsymbol{\sigma}| \leq m$ exist and are continuous over Ω. We shall denote by $C^\infty(\Omega)$ the space of functions w for which $D^{\boldsymbol{\sigma}} w$ exists and is continuous for any order $|\boldsymbol{\sigma}|$. (Although the domain Ω shall subsequently denote the particular spatial domain over which we define our PDE, at present Ω is any bounded open suitably smooth region in $\mathbb{R}^d$.)

We next introduce the family of Banach spaces $L^p(\Omega)$: for $1 \le p < \infty$,

$$L^p(\Omega) \equiv \left\{ w \text{ measurable} \,\Big|\, \left(\int_\Omega |w|^p \right)^{1/p} < \infty \right\} ; \tag{1.58}$$

the associated $L^p(\Omega)$ norm is

$$\|w\|_{L^p(\Omega)} \equiv \left(\int_\Omega |w|^p \right)^{1/p}, \qquad \forall\, w \in L^p(\Omega) \ . \tag{1.59}$$

We recall that $\|w\|_{L^\infty(\Omega)}$ should be interpreted as the essential supremum,

$$\|w\|_{L^\infty(\Omega)} \equiv \inf_D \sup_{\Omega\backslash D} |w| \tag{1.60}$$

over all sets D of zero measure.

The particular Lebesgue space $p = 2$ is of central importance. We introduce the (now) Hilbert space $L^2(\Omega) \equiv \{w \text{ measurable} \,|\, \int_\Omega w^2 < \infty\}$ equipped with inner product and induced norm

$$(w, v)_{L^2(\Omega)} \equiv \int_\Omega wv, \qquad \forall\, w, v \in L^2(\Omega) \ , \tag{1.61}$$

$$\|w\|_{L^2(\Omega)} \equiv \sqrt{(w, w)_{L^2(\Omega)}}, \qquad \forall\, w \in L^2(\Omega) \ . \tag{1.62}$$

In words, $L^2(\Omega)$ is the space of all functions $w\colon \Omega \to \mathbb{R}$ that are square-integrable over Ω. (It is clear from our definitions that $L^2(\Omega)$ is an inner product space; it can also be shown that this space is complete, and hence a Hilbert space [2].)

We next introduce the family of Hilbert spaces, $H^m(\Omega)$, $m \in \mathbb{N}_0$,

$$H^m(\Omega) \equiv \{w \text{ measurable} \mid D^{\boldsymbol{\sigma}} w \in L^2(\Omega),\ \forall \boldsymbol{\sigma} \in I^{d,m}\} \tag{1.63}$$

with inner product and norm

$$(w, v)_{H^m(\Omega)} = \sum_{\boldsymbol{\sigma} \in I^{d,m}} \int_\Omega D^{\boldsymbol{\sigma}} w \, D^{\boldsymbol{\sigma}} v \ , \tag{1.64}$$

and

$$\|w\|_{H^m(\Omega)} = \sqrt{(w,w)_{H^m(\Omega)}}\ , \tag{1.65}$$

respectively; note that the derivatives in (1.63)–(1.65) are to be interpreted in the distributional sense [79]. These spaces may be generalized in many ways: to fractional and negative m [2]; and to general L^p $(p \neq 2)$ norms — corresponding to the family of Sobolev (Banach) spaces [2]. However, for our purposes the most intuitive case — non-negative integer m and the L^2 norm — shall suffice.

In fact, given our exclusive emphasis on (formulation of) *second-order* PDEs, we shall require mostly $H^0(\Omega) = L^2(\Omega)$ and $H^1(\Omega)$. We have already provided details for the former; we here present the latter. In particular, it follows directly from (1.63) that

$$H^1(\Omega) \equiv \left\{ w \in L^2(\Omega) \;\middle|\; \frac{\partial w}{\partial x_i} \in L^2(\Omega),\ 1 \leq i \leq d \right\} \tag{1.66}$$

equipped with inner product and induced norm

$$\begin{aligned} (w,v)_{H^1(\Omega)} &\equiv \int_\Omega \nabla w \cdot \nabla v + wv, \qquad &\forall\, w,v \in H^1(\Omega), \qquad (1.67)\\ \|w\|_{H^1(\Omega)} &\equiv \sqrt{(w,w)_{H^1(\Omega)}}, \qquad &\forall\, w \in H^1(\Omega)\ ; \qquad (1.68) \end{aligned}$$

we also introduce the H^1 *seminorm*,

$$|w|_{H^1(\Omega)} \equiv \int_\Omega \nabla w \cdot \nabla w, \qquad \forall\, w \in H^1(\Omega)\ . \tag{1.69}$$

Finally, we define the space

$$H_0^1(\Omega) \equiv \{ v \in H^1(\Omega) \mid v|_{\partial\Omega} = 0 \}; \tag{1.70}$$

here $v|_{\partial\Omega}$ denotes the trace of v on the boundary of Ω. We note that for $H_0^1(\Omega)$, thanks to the Poincare-Friedrichs inequality, the seminorm (1.69) in fact constitutes a (alternative, equivalent) norm: for $v \in H_0^1(\Omega)$, $|v|_{H^1(\Omega)} = 0$ implies $v = 0$.

The spaces above are defined for the scalar case, $d_v = 1$. However, in all cases we can construct the corresponding "vector" fields by the Cartesian product recipe of Section 1.1.2.

We consider here just the particularly important cases of $L^2(\Omega)$ and $H^1(\Omega)$: for vector fields $w = (w_1, \ldots, w_{d_v}) \in (L^2(\Omega))^{d_v}$,

$$(L^2(\Omega))^{d_v} \equiv \{w_i \in L^2(\Omega),\ 1 \le i \le d_v\}\ , \tag{1.71}$$

$$(w, v)_{L^2(\Omega)} \equiv \sum_{i=1}^{d_v} (w_i, v_i)_{L^2(\Omega)} \left(= \sum_{i=1}^{d_v} \int w_i v_i \right)\ , \tag{1.72}$$

$$\|w\|_{L^2(\Omega)} \equiv \left(\sum_{i=1}^{d_v} \|w_i\|^2_{L^2(\Omega)} \right)^{1/2} \left(= \left(\sum_{i=1}^{d_v} \int w_i^2 \right)^{1/2} \right)\ ; \tag{1.73}$$

and for vector fields $w = (w_1, \ldots, w_{d_v}) \in (H^1(\Omega))^{d_v}$,

$$(H^1(\Omega))^{d_v} \equiv \{w_i \in H^1(\Omega),\ 1 \le i \le d_v\}\ , \tag{1.74}$$

$$(w, v)_{H^1(\Omega)} \equiv \sum_{i=1}^{d_v} (w_i, v_i)_{H^1(\Omega)} \left(= \sum_{i=1}^{d_v} \int_\Omega \nabla w_i \cdot \nabla v_i + w_i v_i \right)\ , \tag{1.75}$$

$$\|w\|_{H^1(\Omega)} \equiv \left(\sum_{i=1}^{d_v} \|w_i\|^2_{H^1(\Omega)} \right)^{1/2} \left(= \left(\sum_{i=1}^{d_v} \int_\Omega |\nabla w_i|^2 + w_i^2 \right)^{1/2} \right)\ . \tag{1.76}$$

We shall let the arguments of the inner products and norms distinquish between the scalar and vector cases; note in particular that (1.71)–(1.76) are consistent for $d_v = 1$ (or for $d_v = d$).

1.4.2 Parametric Functions

Parameter Domains and Grids

We first recall our *closed*, *bounded*, and *suitably regular* parameter domain $\mathcal{D} \subset \mathbb{R}^P$, a typical point in which shall be denoted $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)$. It shall also prove convenient to introduce $\mathcal{D}_{\text{box}} \subset \mathbb{R}^P$ as the smallest parallel-"P"ped such that $\mathcal{D} \subset \mathcal{D}_{\text{box}}$: $\mathcal{D}_{\text{box}} \equiv [\mu_1^{\min}, \mu_1^{\max}] \times \cdots \times [\mu_P^{\min}, \mu_P^{\max}]$, where

$$\mu_p^{\min} = \min_{\boldsymbol{\mu} \in \Omega} \mu_p, \quad \mu_p^{\max} = \max_{\boldsymbol{\mu} \in \Omega} \mu_p, \qquad 1 \le p \le P\ ; \tag{1.77}$$

we also denote $\mu^{\min} \equiv \min_{p \in \{1,\ldots,P\}} \mu_p^{\min}$ and $\mu^{\max} \equiv \max_{p \in \{1,\ldots,P\}} \mu_p^{\max}$. It shall often prove worthwhile to consider a logarithmic transformation: for $\mu^{\min} > 0$, we introduce

$$\hat{\mu}_p = \ln \mu_p, \qquad 1 \le p \le P\ ; \tag{1.78}$$

we denote by $\widehat{\mathcal{D}}$ and $\widehat{\mathcal{D}}_{\text{box}}$ $(\equiv [\ln \mu_1^{\min}, \ln \mu_1^{\max}] \times \cdots \times [\ln \mu_P^{\min}, \ln \mu_P^{\max}])$ the image of $\mathcal{D}$ and $\mathcal{D}_{\text{box}}$ under the log transformation (1.78).

In the construction, analysis, and assessment of our RB approximations and *a posteriori* error estimators we shall often need various grids — finite subsets of $\mathcal{D}$ or $\mathcal{D}_{\text{box}}$. Most commonly we shall invoke Monte-Carlo samples $G^{\text{lin}}_{[\text{MC};m]}$ and $G^{\text{ln}}_{[\text{MC};m]}$ (note ln and log shall refer to the logarithm base e and base 10, respectively) for given sample size $m \in \mathbb{N}$. To construct $G^{\text{lin}}_{[\text{MC};m]}$ we draw points uniformly over $\mathcal{D}_{\text{box}}$,

$$\mu_p = \mu_p^{\min} + \text{rand} \times (\mu_p^{\max} - \mu_p^{\min}), \qquad 1 \le p \le P \ ,$$

and reject $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P) \notin \mathcal{D}$; here rand is a random variable uniformly distributed over $[0, 1]$. To construct $G^{\text{ln}}_{[\text{MC};m]}$ we draw points

$$\mu_p = \mu_p^{\min} \exp\left\{ \text{rand} \times \ln\left(\frac{\mu_p^{\max}}{\mu_p^{\min}}\right)\right\}, \qquad 1 \le p \le P \ ,$$

and reject $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P) \notin \mathcal{D}$; this procedure in effect creates a uniform distribution over $\widehat{\mathcal{D}}$. Note we may consider $G^{\text{ln}}_{[\text{MC};m]}$ only if $\mu^{\min} > 0$.

We also introduce one-dimensional deterministic grids $G^{\text{lin}}_{[z_1,z_2;m]}$, $G^{\text{ln}}_{[z_1,z_2;m]}$: for $z_2 \in \mathbb{R} > z_1 \in \mathbb{R}$ (> 0 in the logarithmic case) and $m \in \mathbb{N}$,

$$\begin{aligned} G^{\text{lin}}_{[z_1,z_2;m]} &= \left\{ z_1 + \frac{i-1}{m-1}(z_2 - z_1),\ 1 \le i \le m \right\} && (1.79) \\ G^{\text{ln}}_{[z_1,z_2;m]} &= \left\{ z_1 \exp\left\{ \frac{i-1}{m-1} \ \ln\left(\frac{z_2}{z_1}\right)\right\},\ 1 \le i \le m \right\} \ ; && (1.80) \end{aligned}$$

note $\hat{z}_i = \ln(z_i)$ is *equi-distributed* for $G^{\text{ln}}_{[z_1,z_2;m]}$. We also define grids based on the Chebyshev Gauss-Lobatto points

$$\begin{aligned} G^{\text{lin, Cheb}}_{[z_1,z_2;m]} &= \left\{ z_1 + \frac{1}{2}\left(1 - \cos \pi \left(\frac{i-1}{m-1}\right)\right)(z_2 - z_1),\ 1 \le i \le m \right\}, && (1.81) \\ G^{\text{ln, Cheb}}_{[z_1,z_2;m]} &= \left\{ z_1 \exp\left\{ \frac{1}{2}\left(1 - \cos \pi \left(\frac{i-1}{m-1}\right)\right) \ln\left(\frac{z_2}{z_1}\right)\right\},\ 1 \le i \le m \right\} \ ; && (1.82) \end{aligned}$$

which shall prove useful in several comparisons. Note that we can combine our one-dimensional grids in tensor product form to create multi-dimensional grids: for example, $G^{\text{lin}}_{[\mu_1^{\min},\mu_1^{\max};m]} \times \cdots \times G^{\text{lin}}_{[\mu_P^{\min},\mu_P^{\max};m]}$ is a grid of m^P points over $\mathcal{D}_{\text{box}}$.

Parametric Scalar and Vector Fields

We now define a *parametric* scalar (respectively, vector) field w as the application w: $\Omega \times \mathcal{D} \to \mathbb{R}^{d_v}$ for $d_v = 1$ (respectively, $d_v = d$); we shall denote this field as $w(\boldsymbol{x};\boldsymbol{\mu})$.

We shall denote the multi-index *parametric* (or "sensitivity") derivative of a scalar (or one component of a vector) parametric field $w(\boldsymbol{x};\boldsymbol{\mu})$ as

$$(D_{\boldsymbol{\sigma}} w)(\boldsymbol{x};\boldsymbol{\mu}) = \frac{\partial^{\boldsymbol{\sigma}} w}{\partial \mu_1^{\sigma_1} \cdots \partial \mu_P^{\sigma_P}} \tag{1.83}$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_P)$ is an index vector of non-negative integers $\sigma_1, \ldots, \sigma_P$; we denote by $|\boldsymbol{\sigma}| = \sum_{j=1}^{P} \sigma_j$ the order of the derivative. (Of course our definition of parameter derivatives here parallels the definition of spatial derivatives in Section 1.4.1; subscript $_{\boldsymbol{\sigma}}$ denotes the former and superscript $^{\boldsymbol{\sigma}}$ denotes the latter. We shall have no direct need for space-parameter cross derivatives.) As before, $I^{P,n}$ denotes the set of all index vectors $\boldsymbol{\sigma} \in \mathbb{N}_0^P$ such that $|\boldsymbol{\sigma}| \le n$.

We shall say that a parametric scalar (or one component of a vector) field w: $\Omega \times \mathcal{D} \to \mathbb{R}$ is "*separable*" over Ω if, for some finite M,

$$w(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{j=1}^{M} h^j(\boldsymbol{x})\, g^j(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{x} \in \Omega,\ \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \tag{1.84}$$

for h^j: $\Omega \to \mathbb{R}$, g^j: $\mathcal{D} \to \mathbb{R}$, $1 \le j \le M$. We shall further say that w is $\boldsymbol{x}$*-affine separable* over Ω if additionally each $h^j(\boldsymbol{x})$ is affine in $\boldsymbol{x}$:

$$h^j(\boldsymbol{x}) = C_0^j + \sum_{i=1}^{d} C_i^j x_i, \qquad C_i^j \in \mathbb{R},\ 0 \le i \le d,\ 1 \le j \le M\ . \tag{1.85}$$

The restrictions (1.84),(1.85) shall place an important role in defining admissible geometric parametrizations.

Parametric Function Spaces

Although on occasion, in particular as regards convergence theory, it will be important to "think" of $w(\boldsymbol{x};\boldsymbol{\mu})$ as $\boldsymbol{x} \in \Omega \rightarrow w(\boldsymbol{x};\,\cdot\,) \in Z(\mathcal{D})$ — where $Z(\mathcal{D})$ is a function space over $\mathcal{D}$ — more often we shall think of $w(\boldsymbol{x};\boldsymbol{\mu})$ as $\boldsymbol{\mu} \in \mathcal{D} \rightarrow w(\,\cdot\,;\boldsymbol{\mu}) \in Z(\Omega)$ — where $Z(\Omega)$ is a function space over Ω. In the latter case, we shall often abbreviate $w(\,\cdot\,;\boldsymbol{\mu})$ by $w(\boldsymbol{\mu})$.

Given a (say, scalar) function space $Z(\Omega)$ — for example, $Z(\Omega) = H^1(\Omega)$ of (1.66) — we shall define

$$C^m(\mathcal{D};Z(\Omega)) \equiv \left\{D_{\boldsymbol{\sigma}} w(\,\cdot\,;\boldsymbol{\mu}) \in Z(\Omega),\ \forall\, \boldsymbol{\sigma} \in I^{P,m},\ \forall \boldsymbol{\mu} \in \mathcal{D}\right\} \ . \tag{1.86}$$

(More precisely, we should require that the derivatives of w in $\boldsymbol{\mu}$ exist and are furthermore continuous as measured in the Z norm.) For instance, if $m = 1$ (and $Z(\Omega) = H^1(\Omega)$),

$$C^1(\mathcal{D};H^1(\Omega)) \equiv \left\{ w(\,\cdot\,;\boldsymbol{\mu}) \in H^1(\Omega) \text{ and } \frac{\partial w}{\partial \mu_p}(\,\cdot\,;\boldsymbol{\mu}) \in H^1(\Omega),\ 1 \le p \le P,\ \forall \boldsymbol{\mu} \in \mathcal{D} \right\} \ . \tag{1.87}$$

Note functions $C^m(\mathcal{D};H^1(\Omega))$ for larger m are very smooth in parameter but not (necessarily) very smooth in space.

1.5 The Complex Case

Our discussion above focuses on *real* vector spaces. However, we shall also on occasion require complex vector spaces — most notably in Part III in the context of acoustics problems — and we thus discuss here the necessary extensions. In general, we treat the complex case as an "overload" of the real case, with the obvious extension/identification (according to the development of this section) if $\mathbb{R}$ is replaced with $\mathbb{C}$.

We first recall the usual notations. For $y \in \mathbb{C}$ (a complex number), $y = \operatorname{Re} y + \mathrm{i} \operatorname{Im} y$, where Re (respectively, Im) refers to the real (respectively, imaginary) part, and $\mathrm{i} = \sqrt{-1}$. We denote the complex conjugate of y as $\overline{y} = \operatorname{Re} y - \mathrm{i} \operatorname{Im} y$, and the modulus of y as $|y| = ((\operatorname{Re} y)^2 + (\operatorname{Im} y)^2)^{1/2} = (y\,\overline{y})^{1/2}$. We also recall that y can be represented as $y = |y| \mathrm{e}^{\mathrm{i}\varphi}$ for

$\varphi = \tan^{-1}(\operatorname{Im} y / \operatorname{Re} y)$.

We can readily generate a complex inner product space $Z_{\mathbb{C}}$ from a corresponding real inner product space $Z_{\mathbb{R}}$ [108]. We first extend the underlying vector space as $Z_{\mathbb{C}} = Z_{\mathbb{R}} \times Z_{\mathbb{R}}$: for any two members of $Z_{\mathbb{C}}$, $w_1 = (\operatorname{Re} w_1 \in Z_{\mathbb{R}},\ \operatorname{Im} w_1 \in Z_{\mathbb{R}})$ and $w_2 = (\operatorname{Re} w_2 \in Z_{\mathbb{R}},\ \operatorname{Im} w_2 \in Z_{\mathbb{R}})$, $w_1 + w_2 \equiv (\operatorname{Re} w_1 + \operatorname{Re} w_2, \operatorname{Im} w_1 + \operatorname{Im} w_2)$; for any complex number $\alpha \in \mathbb{C}$, $\alpha w_1 \equiv (\operatorname{Re}\alpha \operatorname{Re} w_1 - \operatorname{Im}\alpha \operatorname{Im} w_1, \operatorname{Im}\alpha \operatorname{Re} w_1 + \operatorname{Re}\alpha \operatorname{Im} w_1)$. Finally, our notion of basis (1.1) directly extends to the complex case, except that now $\alpha_j \in \mathbb{C}$, $1 \leq j \leq \dim(Z)$.

We then introduce our inner product as $(\,\cdot\,,\,\cdot\,)_{Z_{\mathbb{C}}}$: $Z_{\mathbb{C}} \times Z_{\mathbb{C}} \rightarrow \mathbb{C}$ as $(w_1, w_2)_{Z_{\mathbb{R}}} = ((\operatorname{Re} w_1, \operatorname{Re} w_2)_{Z_{\mathbb{R}}} + (\operatorname{Im} w_1, \operatorname{Im} w_2)_{Z_{\mathbb{R}}}, (\operatorname{Im} w_1, \operatorname{Re} w_2)_{Z_{\mathbb{R}}} - (\operatorname{Re} w_1, \operatorname{Im} w_2)_{Z_{\mathbb{R}}})$; note that now $(w_2, w_1) = \overline{(w_1, w_2)}$. As usual, our inner product induces a (well-defined) norm $\|w\|_{Z_{\mathbb{C}}}$ which is *real-valued*; furthermore, for our definitions, $\|w\|_{Z_{\mathbb{C}}} = \||w_1|\|_{Z_{\mathbb{R}}}$. The Cauchy-Schwarz inequality directly applies, but of course with $|\cdot|$ in (1.2) now interpreted as complex modulus.

We say that a functional g: $Z_{\mathbb{C}} \rightarrow \mathbb{C}$ is an antilinear (sloppily, linear, understanding from $\mathbb{C}$ the context) if, for any $\alpha \in \mathbb{C}$, $w, v \in Z_{\mathbb{C}}$, $g(\alpha w + v) = \overline{\alpha}\, g(w) + g(v)$. Our definition of dual norm, (1.8), still applies, though now we must consider $|g(v)|$ (complex modulus) in the numerator; (1.9) and (1.10) also still apply — and note that (even in the complex case) there is *no* complex modulus in (1.9).

We similarly extend the notion of a bilinear form b. A form b: $Z_{\mathbb{C}} \times Z_{\mathbb{C}} \rightarrow \mathbb{C}$ (or over $V_{\mathbb{C}} \times W_{\mathbb{C}}$) is sesquilinear (sloppily, bilinear, understanding from $\mathbb{C}$ the context) if, for given $w \in Z_{\mathbb{C}}$, $b(w, v)$ is antilinear in v, and for given $v \in Z_{\mathbb{C}}$, $\overline{b(w, v)}$ is antilinear in w. We say that b is symmetric or *Hermitian if* $b(w, v) = \overline{b(v, w)}$, $\forall\, w, v \in Z_{\mathbb{C}}$; we define the Hermitian part of b as $b_{\mathrm{H}}(w, v) = \frac{1}{2}(b(w, v) + \overline{b(v, w)}\,)$, $\forall\, w, v \in Z_{\mathbb{C}}$.

The discussion of parametric linear and bilinear forms readily extends to the complex case; note we assume without loss of generality that our parameter remains real, $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$. The notion of affine parameter dependence (1.18), (1.19) is directly applicable to the complex case:

now, however, $\Theta_g^q\colon\ \mathcal{D} \to \mathbb{C}$, $1 \le q \le Q_q$, and the g^q, $1 \le q \le Q_g$ are bounded antilinear functionals over $Z_{\mathbb{C}}$; similarly, $\Theta_b^q\colon\ \mathcal{D} \to \mathbb{C}$, $1 \le q \le Q_b$, and the b^q, $1 \le q \le Q_b$, are sesquilinear forms over $Z_{\mathbb{C}} \times Z_{\mathbb{C}}$.

The inf-sup continuity notions developed in Section 1.3 for the real case, $b\colon\ W \times V \times \mathcal{D} \to \mathbb{R}$, W and V real spaces, directly extend to the complex case, $b\colon\ W \times V \times \mathcal{D} \to \mathbb{C}$, W and V complex spaces: we simply consider $|b(w,v;\boldsymbol{\mu})|$ in (1.24) and (1.35). Most importantly, the definition of $T_{\boldsymbol{\mu}} w$ in the complex case remains exactly as in the real case (1.27): no complex modulus is introduced. As a result neither the expressions for $\beta(\boldsymbol{\mu})$ and $\gamma(\boldsymbol{\mu})$ in terms of $T_{\boldsymbol{\mu}} w$ — (1.31) and (1.35), respectively — nor the (now Hermitian) inf-sup eigenproblem (1.40), (1.41) and associated identifications, (1.42), (1.43), require any modification.

Finally, our discussion of functions in Section 1.4 admits direct extension to the complex case. For example, given a complex scalar field $w\colon\ \Omega \to \mathbb{C}$, we can readily define $D^{\boldsymbol{\sigma}} w\colon\ \Omega \to \mathbb{C}$ as $D^{\boldsymbol{\sigma}} \operatorname{Re} w + \mathrm{i}\, D^{\boldsymbol{\sigma}} \operatorname{Im} w$ and subsequently — following our "complexification" recipe above — identify

$$H^1(\Omega) \equiv \left\{ \int_\Omega |w|^2 < \infty,\ \int_\Omega \left| \frac{\partial w}{\partial x_i} \right|^2 < \infty,\ 1 \le i \le d \right\}$$

with inner product and norm

$$\begin{aligned} (w,v)_{H^1(\Omega)} &\equiv \int_\Omega \nabla w \cdot \nabla \bar{v} + \int_\Omega w \bar{v}\ , \\ \|w\|_{H^1(\Omega)} &\equiv \left(\int_\Omega |\nabla w|^2 + \int_\Omega |w|^2 \right)^{1/2}\ ; \end{aligned}$$

here $|\cdot|$ refers to the complex modulus. Our parametric function spaces (1.86) admit similar generalization.

Part I

Parametrically Coercive and Compliant Affine Linear Elliptic Problems

Notice

Chapter 2

Abstract Formulation

2.1 Exact Statement

In what follows, the mysterious $^{\mathrm{e}}$ refers to "exact" — associated with the exact solution to our problem for the prescribed mathematical model. (Uncertainties in the mathematical model are briefly considered in Part VIII.)

2.1.1 Space X^{e}

In this section we define the function spaces associated with our PDE field variable.

We first recall our (suitably regular) physical domain $\Omega \in \mathbb{R}^d$ with boundary $\partial\Omega$ of Section 1.4.1; recall that $d = 1$, 2, or 3 is the spatial dimension. In this Part of the book, we shall consider only real-valued field variables. However, we shall already consider here both scalar-valued (e.g., temperature in Poisson problems) and vector-valued (e.g., displacement in linear elasticity problems) field variables w: $\Omega \rightarrow \mathbb{R}^{d_v}$: we recall that d_v denotes the dimension of the field variable; for scalar-valued fields, $d_v = 1$, while for vector-valued fields, $d_v = d$. We also introduce (boundary measurable) segments of $\partial\Omega$, Γ_i^D, $1 \leq i \leq d_v$, over which we shall ultimately impose Dirichlet — in our context, essential — boundary conditions on the components of the field variable.

We next introduce the scalar spaces $Y_i^{\rm e}$, $1 \leq i \leq d_v$,

$$Y_i^{\rm e} \equiv Y_i^{\rm e}(\Omega) \equiv \{v \in H^1(\Omega) \mid v|_{\Gamma_i^D} = 0\}, \qquad 1 \leq i \leq d_v \ ; \tag{2.1}$$

in general $H_0^1(\Omega) \subset Y_i^{\rm e} \subset H^1(\Omega)$, and for $\Gamma_i^D = \partial\Omega$, $Y_i^{\rm e} = H_0^1(\Omega)$. We then construct the space in which our vector-valued field variable shall reside as the Cartesian product $X^{\rm e} = Y_1^{\rm e} \times \ldots Y_{d_v}^{\rm e}$; a typical element of $X^{\rm e}$ shall be denoted $w = (w_1, \ldots, w_{d_v})$. (Not all problems and in particular domains/boundary conditions will admit this Cartesian product form; however, $Y^{\rm e}$ provides sufficient scope for our expositional purposes here. Note that neither our formulation nor the software provided is in fact restricted to the Cartesian product case.) We equip $X^{\rm e}$ with an inner product $(w, v)_{X^{\rm e}}$, $\forall\, w, v, \in X^{\rm e}$, and induced norm $\|w\|_{X^{\rm e}} = \sqrt{(w,w)_{X^{\rm e}}}$, $\forall\, w \in X^{\rm e}$: any inner product which induces a norm equivalent to the $(H^1(\Omega))^{d_v}$ norm is permissible; we shall propose particular candidates below.

We next recall our (suitably regular) closed parameter domain $\mathcal{D} \in \mathbb{R}^P$, a typical parameter (or input) point, or vector, or P-tuple, in which shall be denoted $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_P)$. We may then define our parametric field variable as $u \equiv (u_1, \ldots, u_{d_v})$: $\mathcal{D} \to X^{\rm e}$; here, $u(\boldsymbol{\mu})$ denotes the field for parameter value $\boldsymbol{\mu} \in \mathcal{D}$. (It shall also prove convenient on occasion to view the field variable fully as u: $\Omega \times \mathcal{D} \to \mathbb{R}$; in this case, $u(\boldsymbol{x}; \boldsymbol{\mu})$ denotes the value of the field at point $\boldsymbol{x} \in \Omega$ for parameter value $\boldsymbol{\mu} \in \mathcal{D}$.) Note that in the scalar case we shall denote the field either as $u(\boldsymbol{\mu})$ or as $u_1(\boldsymbol{\mu})$; the latter permits general formulas relevant to both the scalar and vector case.

2.1.2 Parametric Weak Form

We shall first briefly describe the general problem to be addressed in Part II and Part III; we then impose the restrictions for the class of problems to be addressed here in Part I. We are given affine parametric linear forms f and ℓ that are bounded over $X^{\rm e}$, and an affine parametric bilinear form a that is inf-sup stable (with constant $\beta_0 > 0$) and continuous (with constant γ_0)

over X^{e}. Then, given $\boldsymbol{\mu} \in \mathcal{D}$, we find $u^{\mathrm{e}}(\boldsymbol{\mu}) \in X^{\mathrm{e}}$ such that

$$a(u^{\mathrm{e}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall\, v \in X^{\mathrm{e}}\ , \tag{2.2}$$

and evaluate

$$s^{\mathrm{e}}(\boldsymbol{\mu}) = \ell(u^{\mathrm{e}}(\boldsymbol{\mu}); \boldsymbol{\mu})\ . \tag{2.3}$$

Here s^{e} is our output of interest, $s^{\mathrm{e}}\colon\ \mathcal{D} \to \mathbb{R}$ is the input (parameter)-output relationship, and ℓ is the linear "output" functional which links the input to the output through the field variable. (In Part II and Part III, we shall consider multiple outputs as well as quadratic output functionals.)

We recall the interpretation of "affine in parameter" from Section 1.2.5:

$$\ell(v; \boldsymbol{\mu}) \;=\; \sum_{q=1}^{Q_\ell} \Theta_\ell^q(\boldsymbol{\mu})\, \ell^q(v), \qquad \forall\, v \in X^{\mathrm{e}},\ \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \tag{2.4}$$

$$f(v; \boldsymbol{\mu}) \;=\; \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu})\, f^q(v), \qquad \forall v \in X^{\mathrm{e}},\ \forall \boldsymbol{\mu} \in \mathcal{D}\ , \tag{2.5}$$

and

$$a(w, v; \boldsymbol{\mu}) \;=\; \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(w, v), \qquad \forall\, w, v \in X^{\mathrm{e}}, \forall \boldsymbol{\mu} \in \mathcal{D}\ , \tag{2.6}$$

for finite — and, as we shall see, preferably modest — Q_ℓ, Q_f, and Q_a. We of course implicitly assume that the Θ_ℓ^q for $1 \le q \le Q_\ell$, Θ_f^q for $1 \le q \le Q_f$, and Θ_a^q for $1 \le q \le Q_a$ are simple algebraic expressions that can be readily evaluated in $O(1)$ operations. (This is universally true in all the examples in this book, though we can certainly envision situations where the $\Theta_a^q(\boldsymbol{\mu})$ are the result of extensive computation — and perhaps even in need of RB approximation! Such excitement.)

We shall impose two restrictions on the class of problems that we shall treat in Part I. First, we shall presume that our problem is "compliant" (a term that originates in the solid mechanics literature [140], to denote the displacement associated with an applied load). A compliant problem satisfies two conditions on (2.2),(2.3): (i) $\ell(\,\cdot\,; \boldsymbol{\mu}) = f(\,\cdot\,; \boldsymbol{\mu})$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$

— the output functional and "load/source" functional are identical, and (*ii*) a is symmetric. Together, these two assumptions greatly simplify the formulation (no adjoint is required), the *a priori* convergence theory for the output, and the *a posteriori* error estimation for the output. Though very restrictive, there are certainly interesting problems which in fact satisfy the "compliance" requirements.

Second, we shall presume that our bilinear form a is parametrically coercive. To wit, as described in Section 1.2.6 and specialized here (thanks to compliance) to the case of symmetric a, we require that (*i*) the $\Theta_a^q(\boldsymbol{\mu})$, $1 \le q \le Q_a$, are non-negative for all $\boldsymbol{\mu} \in \mathcal{D}$, and (*ii*) the a^q, $1 \le q \le Q_a$, are symmetric semipositive definite (SSPD). This assumption of parametric coercivity shall greatly simplify the stability lower bound required for our *a posteriori* error estimation theory. Though restrictive, there are again interesting problems which do satisfy the "parametric coercivity" requirements, both in the elliptic but also the parabolic context.

To avoid any confusion, we restate our problem for this Part of the book: Given $\boldsymbol{\mu} \in \mathcal{D}$, we find $u^{\mathrm{e}}(\boldsymbol{\mu}) \in X^{\mathrm{e}}$ such that

$$a(u^{\mathrm{e}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall\, v \in X^{\mathrm{e}} \ , \tag{2.7}$$

and evaluate

$$s^{\mathrm{e}}(\boldsymbol{\mu}) = f(u^{\mathrm{e}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \ . \tag{2.8}$$

Here f is a bounded parametric linear form with affine parameter dependence, (2.5); and a is a continuous, parametrically coercive bilinear form with affine parameter dependence, (2.6).

It follows from our assumptions and the Lax-Milgram theorem [125] that (2.7), (2.8) admits a unique solution.

2.1.3 Inner Products and Norms

Given that a is coercive, we may introduce the usual energy inner product and the induced energy norm as

$$(((w,v)))_{\boldsymbol{\mu}} \;=\; a(w,v;\boldsymbol{\mu}), \qquad \forall\, w,v \in X^{\mathrm{e}} , \tag{2.9}$$

$$|||w|||_{\boldsymbol{\mu}} \;\equiv\; \sqrt{a(w,w;\boldsymbol{\mu})}, \qquad \forall\, w \in X^{\mathrm{e}} , \tag{2.10}$$

respectively; note that these quantities are *parameter-dependent*. It is clear that (2.9) constitutes a well-defined inner product — and (2.10), as required, an induced norm equivalent to the $H^1(\Omega)$ norm (1.76) — thanks to our coercivity and continuity assumptions on a.

We can now specify the inner product and norm associated to X^{e}. In particular, we shall choose an energy inner product and norm associated with a specific parameter value $\overline{\boldsymbol{\mu}} \in \mathcal{D}$:

$$(w,v)_{X^{\mathrm{e}}} \;\equiv\; (((w,v)))_{\overline{\boldsymbol{\mu}}} \;(= a(w,v;\overline{\boldsymbol{\mu}}))\,, \qquad \forall\, w,v \in X^{\mathrm{e}} , \tag{2.11}$$

$$\|w\|_{X^{\mathrm{e}}} \;\equiv\; |||w|||_{\overline{\boldsymbol{\mu}}} \;\left(= \sqrt{a(w,w;\overline{\boldsymbol{\mu}})}\right), \qquad \forall\, w \in X^{\mathrm{e}} . \tag{2.12}$$

We address in Chapter 4 how we might choose $\overline{\boldsymbol{\mu}}$ and how this choice affects our numerical results; we note already here that the choice of norm (and hence of $\overline{\boldsymbol{\mu}}$) *does not* affect the RB output prediction but *does* affect the sharpness of the RB *a posteriori* output error bound. We also consider in Section 4.5 a "multi-inner-product" extension.

Recalling (1.13), we introduce the coercivity constant of a over X^{e} as

$$\alpha^{\mathrm{e}}(\boldsymbol{\mu}) \equiv \inf_{w \in X^{\mathrm{e}}} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|^2_{X^{\mathrm{e}}}} , \tag{2.13}$$

which is positive for all $\boldsymbol{\mu} \in \mathcal{D}$. Similarly, from (1.14), we introduce the continuity constant of a over X^{e} as

$$\gamma^{\mathrm{e}}(\boldsymbol{\mu}) \equiv \sup_{w \in X^{\mathrm{e}}} \sup_{v \in X^{\mathrm{e}}} \frac{a(w,v;\boldsymbol{\mu})}{\|w\|_{X^{\mathrm{e}}}\|v\|_{X^{\mathrm{e}}}} , \tag{2.14}$$

which is finite for all $\boldsymbol{\mu} \in \mathcal{D}$.

2.2 Examples

In general, we shall consider examples — instantiations of our abstractions — that are intended to be representative of larger classes of problems (e.g., conduction, acoustics, linear elasticity). In Part I we shall consider just two model problems: a (steady) heat conduction problem with conductivities as parameters; and a linear elasticity problem with Young's moduli as parameters. In fact, (steady, and also unsteady) heat conduction and diffusion problems can very often be modeled as parametrically coercive problems; we consider several additional parametrically coercive conduction examples — including the important case of simple (piecewise dilation) *geometric* variation/parameters — within the more general framework presented in Part II. (Also in Part II we present much more extensive examples in linear elasticity; in general — for more complex constitutive relations and even *simple* geometric variations — linear elasticity problems are coercive but *not* parametrically coercive.)

Note in our Part I discussion it might appear that the prerequisites — such as affine parameter dependence — for RB treatment can only be verified *a posteriori* — that there is no way to *a priori* frame the *general* class of problems, and associated parametric dependencies, amenable to the RB approach. In fact, this is not the case: In Part II we describe the broad family of PDE operators and in particular geometric and "coefficient" parametric variations to which the RB method can be applied; we reserve this discussion for Part II since, in general, most of these problems will not honor the "parametrically coercive and compliant" restrictions of Part I. Indeed, the *purpose* of Part I is to introduce all the RB concepts in a particularly simple context.

2.2.1 Example 1 (Ex1): ThermalBlock

We consider steady heat conduction [94] in a two-dimensional domain, or "block," $\Omega =]0,1[\, \times\,]0,1[$, shown in Figure 2.1. The block comprises B_1 (in the x_1-direction) $\times$ B_2 (in the x_2-direction) rectangular subblocks/subdomains Ω_i, $i = 1, \ldots, B_1 B_2$, each of dimension

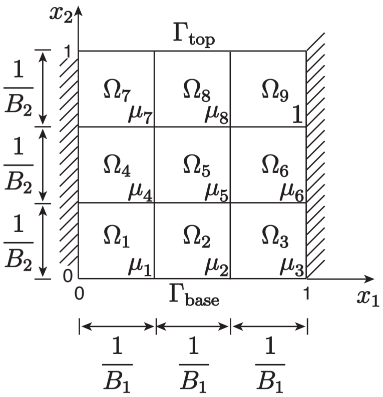

Figure 2.1: ThermalBlock Geometry.

$1/B_1 \times 1/B_2$. For subblock $i = B_1B_2$, corresponding to the subdomain $\Omega_{B_1B_2}$ in the upper right corner of the domain, the thermal conductivity is unity (our reference); for subblocks $i = 1, \ldots, B_1B_2 - 1$, corresponding to the subdomains Ω_i, $i = 1, \ldots, B_1B_2 - 1$, the normalized thermal conductivity is denoted κ_i.

We consider $P = B_1B_2 - 1$ parameters: the conductivities. Our parameter vector is thus given by $\boldsymbol{\mu} \equiv (\mu_1, \ldots, \mu_P) \equiv (\kappa_1, \ldots, \kappa_{B_1B_2-1})$. We choose for our parameter domain $\mathcal{D} = \mathcal{D}_{\text{box}} = [\mu_1^{\min}, \mu_1^{\max}] \times \cdots \times [\mu_P^{\min}, \mu_P^{\max}]$; we shall take the $\mu_p^{\min} = \mu^{\min}$, $1 \leq p \leq P$, and $\mu_p^{\max} = \mu^{\min}$, $1 \leq p \leq P$; furthermore, we shall select a “symmetric” interval $\mu^{\min} = 1/\sqrt{\mu_r}$, $\mu^{\max} = \sqrt{\mu_r}$ (for $1 < \mu_r < \infty$) such that $\mu^{\max}/\mu^{\min} = \mu_r$.

Our (scalar) field variable is the temperature: the temperature satisfies Laplace’s equation in Ω; continuity of temperature and heat flux (the product of the conductivity and the gradient of the temperature) across subblock interfaces [63]; zero Neumann (zero flux, or insulated) conditions on the side boundaries; zero Dirichlet (temperature) conditions on the top boundary $\Gamma_{\text{top}} \equiv \Gamma^D$; and unity Neumann (imposed unity heat flux into the domain) conditions on the bottom boundary, or “base,” Γ_{base}.

The output of interest is the average temperature over the base Γ_{base}. Note that here and in all examples in the book we presume a non-dimensional form in which all unnecessary (i.e., redundant) parameters have been removed.

We recall from Section 2.1.1 that the function space associated with this set of boundary conditions is given by $X^{\mathrm{e}} \equiv \{v \in H^1(\Omega) \mid v|_{\Gamma^D} = 0\}$: the Dirichlet interface and boundary conditions are essential; the Neumann interface and boundary conditions are natural. We can then define our source (and also output) functional

$$f(v;\boldsymbol{\mu}) \equiv \int_{\Gamma_{\mathrm{base}}} v, \qquad \forall\, v \in X^{\mathrm{e}}\ , \tag{2.15}$$

and our bilinear form

$$a(w,v;\boldsymbol{\mu}) \equiv \sum_{p=1}^{P} \mu_p \int_{\Omega_i} \nabla w \cdot \nabla v + \int_{\Omega_{P+1}} \nabla w \cdot \nabla v, \qquad \forall\, w, v \in X^{\mathrm{e}}\ . \tag{2.16}$$

(Recall that the conductivity of block $i = B_1 B_2 = P+1$ — our reference value — is unity.) Finally, our weak form is then given by the abstract statement of Section 2.1.2, ((2.7),(2.8)).

Armed with our bilinear form, we can now specify our inner product according to the recipe (2.11) as

$$(w,v)_{X^{\mathrm{e}}} \equiv \sum_{p=1}^{P} \overline{\mu}_p \int_{\Omega_i} \nabla w \cdot \nabla v + \int_{\Omega_{P=1}} \nabla w \cdot \nabla v, \qquad \forall\, w, v \in X^{\mathrm{e}}\ , \tag{2.17}$$

for a given value $\overline{\boldsymbol{\mu}} \in \mathcal{D}$. In our case we shall take $\overline{\mu}_i = 1$, $1 \leq i \leq P$, corresponding to the "logarithmic center" of the parameter domain. This choice will be justified in Chapter 4.

We can now readily verify our hypotheses. First, it is standard to confirm [158] that f is indeed bounded. Second, we readily confirm by inspection that a is symmetric, and by simple application of the Cauchy Schwarz inequality, that a is coercive,

$$0 < \frac{1}{\sqrt{\mu_r}} \leq \mathrm{Min}\,(\mu_1/\overline{\mu}_1, \ldots, \mu_P/\overline{\mu}_P, 1) \leq \alpha^{\mathrm{e}}(\boldsymbol{\mu})\ , \tag{2.18}$$

and continuous,

$$\gamma^{\mathrm{e}}(\boldsymbol{\mu}) \leq \mathrm{Max}\,(\mu_1/\overline{\mu}_1, \ldots, \mu_P/\overline{\mu}_P, 1) \leq \sqrt{\mu_r} < \infty\ ; \tag{2.19}$$

here Min ( ) (respectively, Max ( )) returns the smallest (respectively, largest) of its arguments. (More detailed coercivity and continuity calculations will be provided in Chapter 4, Section 4.5.) Third, f is clearly affine in the parameter — in fact, f does not explicitly depend

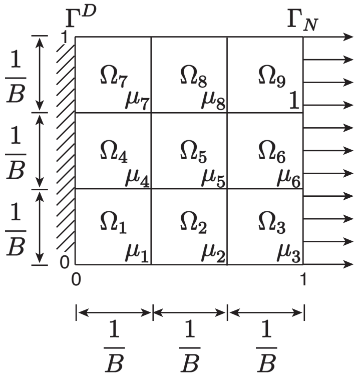

Figure 2.2: Multi-material ElasticBlock.

on the parameter, and hence $Q_f = 1$ with $\Theta^1_f(\boldsymbol{\mu}) = 1$ and $f^1 = f$; and a is affine in the parameter for $Q_a = P + 1$ with $\Theta^q_a(\boldsymbol{\mu}) = \mu_q$, $1 \leq q \leq P$, $\Theta^{P+1}_a = 1$, and

$$a^q(w,v) \equiv \int_{\Omega_i} \nabla w \cdot \nabla v, \qquad 1 \leq i \leq P+1 \ . \tag{2.20}$$

Fourth, and finally, a is parametrically coercive: $\Theta^q_a(\boldsymbol{\mu}) > 0$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$, $1 \leq q \leq Q_a$, and $a^q(w,w) \geq 0$, $\forall\, w \in X^{\rm e}$, $1 \leq q \leq Q_a$.

2.2.2 Example 2 (Ex2): ElasticBlock

We consider a linear elasticity [44, 78] example in the two-dimensional domain, or "material block," $\Omega =]0,1[\, \times \,]0,1[$, shown in Figure 2.2. The block is comprised of B^2 square isotropic subblocks/subdomains Ω_i, $i = 1,\ldots,B^2$, each of sidelength $1/B$. For subblock $i = B^2$, corresponding to the subdomain Ω_{B^2} in the upper right corner of the domain, the Young's modulus is unity (our reference); for subblocks $i = 1,\ldots,B^2-1$, corresponding to the subdomains Ω_i, $i = 1,\ldots,B^2-1$, the normalized Young's modulus is E_i. The Poisson's ratio in all subblocks is $\nu = 0.30$.

We consider $P = B_1B_2 - 1$ parameters: the Young's moduli. Our parameter vector is thus given by $\boldsymbol{\mu} \equiv (\mu_1,\ldots,\mu_P) \equiv (E_1,\ldots,E_{B_1B_2-1})$. We choose for our parameter domain $\mathcal{D} = \mathcal{D}_{\rm box} = [\mu_1^{\min},\mu_1^{\max}] \times \cdots \times [\mu_P^{\min},\mu_P^{\max}]$; we shall take the $\mu_p^{\min} = \mu^{\min}$, $1 \leq p \leq P$, and $\mu_p^{\max} = \mu^{\min}$, $1 \leq p \leq P$; furthermore, we shall select a "symmetric" interval $\mu^{\min} = 1/\sqrt{\mu_r}$,

$\mu^{\max} = \sqrt{\mu_r}$ (for $1 < \mu_r < \infty$) such that $\mu^{\max}/\mu^{\min} = \mu_r$.

Our (vector) field variable $u(\boldsymbol{\mu}) = (u_1(\boldsymbol{\mu}), u_2(\boldsymbol{\mu}))$ is the displacement: the displacement satisfies the plane-strain Linear Elasticity equations in Ω; continuity of displacement and stress across subblock interfaces; zero Neumann (load-free) conditions on the top and bottom boundaries; zero Dirichlet (dispacement) conditions on the left boundary $\Gamma_1^D = \Gamma_2^D = \Gamma^D$ — the "structure" is clamped; and inhomogeneous Neumann conditions on the right boundary Γ_N corresponding to unity tension and zero shear.

The output of interest is the integrated horizontal (x_1-)displacement over the loaded boundary Γ_N; this corresponds to the eponymous compliant situation.

We recall from Section 2.1.1 that the function space associated with this set of boundary conditions is given by $X^{\mathrm{e}} \equiv \{v \in (H^1(\Omega))^2 \mid v|_{\Gamma^D} = 0\}$: the Dirichlet interface and boundary conditions are essential; the Neumann interface and boundary conditions are natural. We can then define our load (and also output) functional

$$f(v; \boldsymbol{\mu}) \equiv \int_{\Gamma_N} v_1, \qquad \forall\, v \in X^{\mathrm{e}} \ , \tag{2.21}$$

and our bilinear form as

$$a(w, v; \boldsymbol{\mu}) \equiv \sum_{p=1}^{P} \mu_p \int_{\Omega^P} \frac{\partial v_i}{\partial x_j}\, C_{ijkl}\, \frac{\partial w_k}{\partial x_l} + \int_{\Omega^{P+1}} \frac{\partial v_i}{\partial x_j}\, C_{ijkl}\, \frac{\partial w_k}{\partial x_l}, \quad \forall\, w, v \in X^{\mathrm{e}} \ . \tag{2.22}$$

For our isotropic material, the elasticity tensor is given by

$$C_{ijkl} = \lambda^1 \delta_{ij}\delta_{kl} + \lambda^2 \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \ , \tag{2.23}$$

where

$$\lambda^1 = \frac{\nu}{(1+\nu)(1-2\nu)} \ , \tag{2.24}$$

$$\lambda^2 = \frac{1}{2(1+\nu)} \ , \tag{2.25}$$

(for $\nu = 0.30$) are the Lamé constants for plane strain. The weak form is then given by ((2.7), (2.8)).

The inner product is specified by the recipe (2.11),

$$(w,v)_{X^{\mathrm{e}}} \equiv \sum_{p=1}^{P} \overline{\mu}_p \int_{\Omega^p} \frac{\partial v_i}{\partial x_j}\, C_{ijkl}\, \frac{\partial w_k}{\partial x_l} + \int_{\Omega^{P+1}} \frac{\partial v_i}{\partial x_j}\, C_{ijkl}\, \frac{\partial w_k}{\partial x_l}, \quad \forall\, w, v \in X^{\mathrm{e}} \ , \tag{2.26}$$

for a given $\overline{\boldsymbol{\mu}} \in \mathcal{D}$; we choose $\overline{\mu}_p = 1$, $1 \le p \le P$.

We can now readily verify our hypotheses. First, it is standard to confirm that f is indeed bounded. Second, we readily confirm by inspection that a is symmetric, and we further verify by application of the Korn inequality [48] and Cauchy Schwarz inequality that a is coercive and continuous, respectively. Third, f is clearly affine in the parameter — in fact, f does not explicitly depend on the parameter, and hence $Q_f = 1$ with $\Theta_f^1(\boldsymbol{\mu}) = 1$ and $f^1 = f$; and a is affine for $Q_a = B^2 = P + 1$ with $\Theta_a^q(\boldsymbol{\mu}) = \mu_q$, $1 \le q \le P$, $\Theta_a^{P+1} = 1$, and

$$a^q(w,v) = \int_{\Omega^q} \frac{\partial v_i}{\partial x_j}\, C_{ijkl}\, \frac{\partial w_k}{\partial x_l}, \quad 1 \le q \le P+1 \ . \tag{2.27}$$

Fourth, and finally, a is parametrically coercive: $\Theta_a^q(\boldsymbol{\mu}) > 0$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$, $1 \le q \le Q_a$, and $a^q(w,w) \ge 0$, $\forall\, w \in X^{\mathrm{e}}$, $1 \le q \le Q_a$; as regards the latter, note we only require *semi*positive-definiteness, and hence the rigid-body modes are not a concern.

2.3 "Truth" Approximation

In this section we develop the (somewhat Orwellianly named) "truth" approximation. We shall build our Reduced Basis (RB) approximation on, and measure the error in the reduced basis approximation relative to, this "truth" approximation. (Historically, the reduced basis method has always relied on an underlying discrete model, either a directly lumped (algebraic) model [6, 95, 99] or an approximation to an infinite-dimensional "exact" PDE [50]. We pursue here the latter.)

2.3.1 Approximation Spaces and Bases

We now introduce a family of *conforming* approximation spaces $X^{\mathcal{N}} \subset X^{\mathrm{e}}$ of dimension $\dim(X^{\mathcal{N}}) = \mathcal{N}$; note $\mathcal{N}$ is not only the dimension of the space but also the label for a particular approximation in a specified sequence. Within our Cartesian product formulation for vector-valued field problems we first construct conforming scalar approximation spaces $Y_i^{\mathcal{N}_i} \subset Y_i^{\mathrm{e}}$ of dimension $\dim(Y_i^{\mathcal{N}_i}) = \mathcal{N}_i$, $1 \leq i \leq d_v$. We then form our vector approximation space as the product of these d_v scalar approximation spaces: $X^{\mathcal{N}} = Y_1^{\mathcal{N}_1} \times \cdots \times Y_{d_v}^{\mathcal{N}_{d_v}}$ and $\mathcal{N} = \sum_{i=1}^{d_v} \mathcal{N}_i$.

We now associate to our space a set of basis functions $\varphi_k^{\mathcal{N}} \in X^{\mathcal{N}}$, $1 \leq k \leq \mathcal{N}$; by construction, any member of $X^{\mathcal{N}}$ can be represented by a unique linear combination of the $\varphi_k^{\mathcal{N}}$, $1 \leq k \leq \mathcal{N}$. Within our Cartesian product formulation for vector-valued field problems we first associate to our scalar approximation spaces $Y_i^{\mathcal{N}_i}$ the basis functions $(\phi_i^{\mathcal{N}_i})_{k'}$, $1 \leq k' \leq \mathcal{N}_i$ for $1 \leq i \leq d_v$. We then form our vector basis as the "sum" of the scalar bases: $\varphi^{\mathcal{N}}_{\mathrm{Ind}\,(i,k')} = (\phi_i^{\mathcal{N}_i})_{k'} \boldsymbol{e}_i$, $1 \leq k' \leq \mathcal{N}_i$, $1 \leq i \leq d_v$; here Ind is a (any) mapping from the double index onto the single index (e.g., $\mathrm{Ind}\,(i,k') = k' + \sum_{i'=1}^{i-1} \mathcal{N}_{i'}$).

Finally, we choose the inner products and norms with which to equip $X^{\mathcal{N}}$. Here we simply inherit the inner products and norms associated with the exact space:

$$(w,v)_{X^{\mathcal{N}}} \equiv (w,v)_{X^{\mathrm{e}}} \equiv a(w,v;\overline{\boldsymbol{\mu}}), \qquad \forall\, w,v \in X^{\mathcal{N}}\ , \tag{2.28}$$

and

$$\|w\|_{X^{\mathcal{N}}} \equiv \|w\|_{X^{\mathrm{e}}} \equiv \sqrt{a(w,w;\overline{\boldsymbol{\mu}})}, \qquad \forall w \in X^{\mathcal{N}}\ . \tag{2.29}$$

We note that the definitions of these inner products and induced norms is in fact independent of $\mathcal{N}$ (in the here-assumed absence of quadrature errors) — only the class of admissible functions grows as we enlarge our approximation space.

From definition (1.13) we introduce the coercivity constant of a over $X^{\mathcal{N}}$ as

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) \equiv \inf_{w \in X^{\mathcal{N}}} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|^2_{X^{\mathcal{N}}}}\ ; \tag{2.30}$$

similarly, from the definition (1.14) we introduce the continuity constant of a over $X^{\mathcal{N}}$ as

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) \equiv \sup_{w \in X^{\mathcal{N}}} \sup_{v \in X^{\mathcal{N}}} \frac{a(w,v;\boldsymbol{\mu})}{\|w\|_{X^{\mathcal{N}}} \|v\|_{X^{\mathcal{N}}}}\ . \tag{2.31}$$

We shall shortly infer various properties of $\alpha^{\mathcal{N}}(\boldsymbol{\mu})$ and $\gamma^{\mathcal{N}}(\boldsymbol{\mu})$.

We shall require that our family of truth subspaces $X^{\mathcal{N}}$ satisfies the approximation condition

$$\max_{\boldsymbol{\mu}\in\mathcal{D}} \inf_{w\in X^{\mathcal{N}}} \|u(\boldsymbol{\mu}) - w\|_{X^{\rm e}} \to 0 \text{ as } \mathcal{N} \to \infty \ . \tag{2.32}$$

In words, (2.32) states that, for any $\varepsilon > 0$, there exists an $\mathcal{N}$ such that the error in the best fit to $u(\boldsymbol{\mu})$ in $X^{\mathcal{N}}$ is less than or equal to ε *for all* $\boldsymbol{\mu}$ in $\mathcal{D}$. In our examples, these approximation spaces will most often be linear or quadratic (or bilinear or biquadratic) finite element (FE) spaces defined over suitable triangulations of Ω. Furthermore, we shall typically consider associated nodal bases with compact support in Ω. From the perspective of this book, only certain features of these approximation spaces and associated bases — apart from the usual efficiency considerations — are important: we highlight these as we proceed.

2.3.2 Galerkin Projection

We can now present our family of approximations to the exact problem. Given $\boldsymbol{\mu} \in \mathcal{D}$, find $u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}}$ such that

$$a(u^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall\, v \in X^{\mathcal{N}} \ , \tag{2.33}$$

and then evaluate

$$s^{\mathcal{N}}(\boldsymbol{\mu}) = f(u^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \ . \tag{2.34}$$

This represents a standard Galerkin projection.

We implicitly assume (and it is indeed the case for our two examples of Section 2.2) that we commit no variational crimes in our Galerkin approximation (2.33),(2.34). For example, we assume that we represent the exact geometry within our truth approximation (which in turn largely implies polygonal domains Ω): if not, we would need in Section 2.1 $\Omega^{\rm e}$ and in the current section $\Omega^{\mathcal{N}}$. Similarly, we assume that all quadratures are exact: if not, we would need in Section 2.1 $f^{\rm e}$ and $a^{\rm e}$ and in the current section $f^{\mathcal{N}}$ and $a^{\mathcal{N}}$. These variational crimes, if present, simply represent additional contributions to the error between the exact solution and the "truth" upon which we shall build the RB approximation: little modification to the framework is required. We thus choose *not* to consider explicitly any variational transgression in (most of) this book; however, both the methodology developed and the software provided are non-judgemental — accepting the user's definition of truth.

Our Galerkin approximation must satisfy certain conditions over $X^{\mathcal{N}}$ — in particular, the same conditions imposed in Section 2.1 on the exact formulation over X^{e} — on which the subsequent RB approximation and *a posteriori* error estimation will depend. For the particular class of problems of interest in Part I — and our conforming approximation subspaces and crime-free projection — the Galerkin formulation in fact directly inherits and even improves upon all the good properties of the exact formulation. The dual norm of f over $X^{\mathcal{N}}$ ($\subset X^{\mathrm{e}}$) is bounded by the dual norm of f over X^{e}; the Galerkin recipe of course preserves symmetry; a is coercive over $X^{\mathcal{N}}$ with (since $X^{\mathcal{N}} \subset X$)

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) \geq \alpha^{\mathrm{e}}(\boldsymbol{\mu}), \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ ;$$

a is continuous over $X^{\mathcal{N}}$ with (since $X^{\mathcal{N}} \subset X$)

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) \leq \gamma^{\mathrm{e}}(\boldsymbol{\mu}), \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ ;$$

our affine expansions for f and a are of course still valid for w, v restricted to $X^{\mathcal{N}}$ ($\subset X^{\mathrm{e}}$); and a clearly still satisfies the two conditions for parametric coercivity — note positive semi-definiteness follows directly from (again) $X^{\mathcal{N}} \subset X^{\mathrm{e}}$. Thus, for any $\mathcal{N}$ and associated $X^{\mathcal{N}}$, our Galerkin approximation preserves the "parametrically coercive and compliant affine" property necessary for the development of Part I.

It directly follows from our hypothesis (2.32) that our Galerkin approximation is convergent: as $\mathcal{N} \to \infty$ — recall this refers to a particular suite of approximations — $u^{\mathcal{N}}(\boldsymbol{\mu}) \to u^{\mathrm{e}}(\boldsymbol{\mu})$ and (since f is bounded) $s^{\mathcal{N}} \to s^{\mathrm{e}}$. Thus, for sufficiently large $\mathcal{N}$ (and in the absence of precision issues) we can approximate $u^{\mathrm{e}}(\boldsymbol{\mu})$ and $s^{\mathrm{e}}(\boldsymbol{\mu})$ arbitrarily closely. For future reference we make this precise; if we define

$$\varepsilon^{\mathcal{N}} = \max_{\boldsymbol{\mu} \in \mathcal{D}} \|u(\boldsymbol{\mu}) - u^{\mathcal{N}}(\boldsymbol{\mu})\|_{X^{\mathrm{e}}} \ ,$$

then $\varepsilon^{\mathcal{N}} \to 0$ as $\mathcal{N} \to \infty$.

2.3.3 Algebraic Equations

The development of the algebraic equations induced by the approximation (2.33),(2.34) and our choice of basis of Section 2.3.1 is standard. In particular, if we expand our solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ as

$$u^{\mathcal{N}}(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{j=1}^{\mathcal{N}} u_j^{\mathcal{N}}(\boldsymbol{\mu})\, \varphi_j^{\mathcal{N}}(\boldsymbol{x}) \ , \tag{2.35}$$

then

$$\underline{u}^{\mathcal{N}}(\boldsymbol{\mu}) \equiv \left[u_1^{\mathcal{N}}(\boldsymbol{\mu})\ u_2^{\mathcal{N}}(\boldsymbol{\mu})\ \ \cdots\ \ u_{\mathcal{N}}^{\mathcal{N}}(\boldsymbol{\mu}) \right]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{N}} \tag{2.36}$$

satisfies

$$\underline{A}^{\mathcal{N}}(\boldsymbol{\mu}) \underline{u}^{\mathcal{N}}(\boldsymbol{\mu}) = \underline{F}^{\mathcal{N}}(\boldsymbol{\mu}) \ ; \tag{2.37}$$

the output of interest can then be expressed as

$$s^{\mathcal{N}}(\boldsymbol{\mu}) = \left(\underline{F}^{\mathcal{N}}(\boldsymbol{\mu}) \right)^{\mathrm{T}} \underline{u}^{\mathcal{N}}(\boldsymbol{\mu}) \ . \tag{2.38}$$

Here superscript T refers to the usual algebraic transpose.

The elements of the stiffness matrix $\underline{A}^{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ are given by

$$A_{i\,j}^{\mathcal{N}}(\boldsymbol{\mu}) = a(\varphi_j^{\mathcal{N}}, \varphi_i^{\mathcal{N}}; \boldsymbol{\mu}), \qquad 1 \leq i, j \leq \mathcal{N} \ ; \tag{2.39}$$

the elements of the load/source vector (and, for this compliant case, output vector) $\underline{F}^{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}}$ are given by

$$F_i^{\mathcal{N}}(\boldsymbol{\mu}) = f(\varphi_i^{\mathcal{N}}; \boldsymbol{\mu}), \qquad 1 \leq i \leq \mathcal{N} \ . \tag{2.40}$$

Of course, by virtue of our assumptions on a, the stiffness matrix $\underline{A}^{\mathcal{N}}(\boldsymbol{\mu})$ is symmetric and positive definite.

We now invoke the affine assumption on f, (2.5), and a, (2.6), to express our stiffness matrix and load/output vector in a form that will subsequently prove quite useful within the Offline-Online RB context. In particular, it follows directly from (2.6) and (2.39) that

$$\underline{A}^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \underline{\mathbb{A}}^{\mathcal{N}q} \ , \tag{2.41}$$

for $\underline{\mathbb{A}}^{\mathcal{N}q} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $1 \le q \le Q_a$, given by

$$\mathbb{A}_{i\,j}^{\mathcal{N}q} = a^q(\varphi_j^{\mathcal{N}}, \varphi_i^{\mathcal{N}}), \qquad 1 \le i, j \le \mathcal{N},\ 1 \le q \le Q_a \ . \tag{2.42}$$

Similarly, it follows directly from (2.5) and (2.40) that

$$\underline{F}^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) \underline{\mathbb{F}}^{\mathcal{N}q} \ , \tag{2.43}$$

for $\underline{\mathbb{F}}^{\mathcal{N}q} \in \mathbb{R}^{\mathcal{N}}$, $1 \le q \le Q_f$, given by

$$\mathbb{F}_i^{\mathcal{N}q} = f^q(\varphi_i^{\mathcal{N}}), \qquad 1 \le i \le \mathcal{N},\ 1 \le q \le Q_f \ . \tag{2.44}$$

Note that the $\underline{\mathbb{A}}^{\mathcal{N}q}$, $1 \le q \le Q_a$, and $\underline{\mathbb{F}}^{\mathcal{N}q}$, $1 \le q \le Q_f$, are all *parameter-independent*.

For completeness we also introduce here another (parameter-independent) matrix $\underline{\mathbb{X}}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ — associated with our inner product — that shall prove quite useful in particular in the *a posteriori* error estimation context:

$$\mathbb{X}_{i\,j}^{\mathcal{N}} = (\varphi_j^{\mathcal{N}}, \varphi_i^{\mathcal{N}})_{X^{\mathcal{N}}}, \qquad 1 \le i, j \le \mathcal{N} \ . \tag{2.45}$$

For any given two members of $X^{\mathcal{N}}$,

$$w = \sum_{j=1}^{\mathcal{N}} w_j\, \varphi_j^{\mathcal{N}} \ , \tag{2.46}$$

and

$$v = \sum_{j=1}^{\mathcal{N}} v_j\, \varphi_j^{\mathcal{N}} \ , \tag{2.47}$$

the $X^{\mathcal{N}}$-inner product can be calculated as

$$\begin{aligned} (w, v)_{X^{\mathcal{N}}} &= \left(\sum_{j=1}^{\mathcal{N}} w_j\, \varphi_j^{\mathcal{N}}, \sum_{i=1}^{\mathcal{N}} v_i\, \varphi_i^{\mathcal{N}} \right)_{X^{\mathcal{N}}} \\ &= \sum_{j=1}^{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} w_j v_i\, (\varphi_j^{\mathcal{N}}, \varphi_i^{\mathcal{N}})_{X^{\mathcal{N}}} \ = \ \underline{w}^{\mathrm{T}} \underline{\mathbb{X}}^{\mathcal{N}} \underline{v} \ , \end{aligned} \tag{2.48}$$

where

$$\underline{w} \equiv [w_1,\ w_2,\ \cdots\ w_{\mathcal{N}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{N}} \tag{2.49}$$

and

$$\underline{v} \equiv [v_1,\ v_2,\ \cdots\ v_{\mathcal{N}}]^{\mathrm{T}} \in \mathbb{R}^{\mathcal{N}} \tag{2.50}$$

are the nodal basis coefficient vectors.

This Galerkin approximation will be invoked frequently within the Offline stage of the RB methodology. We shall refer to "$\underline{A}^{\mathcal{N}}$-solve($K$)" (or "$\underline{A}^{\mathcal{N}}$-solve" if $K = 1$) as the operations to solve K linear systems with matrix $\underline{A}^{\mathcal{N}}(\boldsymbol{\mu})$ (or $(\underline{A}^{\mathcal{N}})^{\mathrm{T}}(\boldsymbol{\mu})$) for some given value of $\boldsymbol{\mu}$ (and of course K different right-hand sides) — for example, (2.37) requires $\underline{A}^{\mathcal{N}}$-solve(1) operations; we shall refer to "$\underline{\mathbb{X}}^{\mathcal{N}}$-solve($K$)" as the operations to solve K linear systems with (parameter-independent) matrix $\underline{\mathbb{X}}^{\mathcal{N}}$ (and of course K different right-hand sides); we shall refer to "$\underline{A}^{\mathcal{N}}$-matvec" as the operations to evaluate a matrix-vector product $\underline{A}^{\mathcal{N}}(\boldsymbol{\mu})\underline{w}^{\mathcal{N}}$ (or $\underline{A}^{\mathcal{N}q}(\boldsymbol{\mu})\underline{w}^{\mathcal{N}}$ for $q \in \{1, \ldots, Q_a\}$) or $(\underline{A}^{\mathcal{N}})^{\mathrm{T}}(\boldsymbol{\mu})\underline{w}^{\mathcal{N}}$ for some given $\underline{w}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$; and we shall refer to "$X^{\mathcal{N}}$-inprod" as the operations — clearly $O(\mathcal{N})$ in number — required to evaluate an inner product $(\underline{w}^{\mathcal{N}})^{\mathrm{T}}\underline{v}^{\mathcal{N}}$ for given $\underline{w}^{\mathcal{N}}, \underline{v}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$. As we shall see, there will be a proliferation of $\underline{A}^{\mathcal{N}}$-matvec's, and hence we shall strongly prefer "truth" approximation spaces and bases which engender very sparse stiffness matrices $\underline{A}^{\mathcal{N}}(\boldsymbol{\mu})$ — or at least permit rapid evaluation procedures (for example, tensor product techniques in the spectral context).

2.3.4 Choice of $\mathcal{N}_{\mathrm{t}}$

We would of course prefer to base the RB approach directly on the exact solution, but this is not in general possible. As indicated earlier, we shall thus build the RB approximation on, and measure the reduced basis error relative to, a particular "truth" Galerkin approximation corresponding to the choice $\mathcal{N} = \mathcal{N}_{\mathrm{t(ruth)}}$: we find $u^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu}) \in X^{\mathcal{N}_{\mathrm{t}}}$ such that

$$a(u^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall\, v \in X^{\mathcal{N}_{\mathrm{t}}}\ , \tag{2.51}$$

and then evaluate the truth output

$$s^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) = f(u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \ . \tag{2.52}$$

We must anticipate that $\mathcal{N}_\mathrm{t}$ will be very large.

In general, $\mathcal{N}_\mathrm{t}$ must be chosen rather large to achieve reasonable engineering accuracies $\varepsilon^{\mathcal{N}_\mathrm{t}}$: many problems of interest exhibit scale and (three-dimensional) geometric complexity. Furthermore, $\mathcal{N}_\mathrm{t}$ must be chosen "worst-case" over $\mathcal{D}$: Offline the RB approximation must be built on a unique truth representation for all $\boldsymbol{\mu} \in \mathcal{D}$; and Online, we do not have access to either an (adaptive) FE error estimation context or human intervention — the "truth" approximation (and hence $\mathcal{N}_\mathrm{t}$) is "frozen." Finally, $\mathcal{N}_\mathrm{t}$ must be chosen conservatively: in particular in the *real-time* context, errors can have serious and immediate (adverse) consequences.

We must thus formulate a reduced basis approach that is (a) numerically stable as $\mathcal{N}_\mathrm{t} \to \infty$, and (b) computationally efficient as $\mathcal{N}_\mathrm{t} \to \infty$. As regards (a) it is crucial, for example, that we choose the correct norms consistent with the exact infinite-dimensional formulation. All norms are of course "equivalent" for a finite-dimensional space: however, the equivalence constants are dependent on the dimension of the space and not necessarily bounded as $\mathcal{N}_\mathrm{t} \to \infty$; a discrete (Euclidean) ℓ_2 or even continuous L^2 dual norm — rather than the correct $H^1(\Omega)$ (equivalent) dual norm — for the residual will lead ultimately to ill-posed and very poor *a posteriori* error estimators. As regards (b), it is crucial to ensure that the operation count (and storage) in the Online stage — for calculation of the reduced basis output prediction and associated *a posteriori* output error bound — is *independent* of $\mathcal{N}_\mathrm{t}$. In this way at least the most crucial performance indicator — marginal response time for input-output evaluation in the Online Stage — will be insensitive to the richness of the truth approximation. (Of course, the operation count for the Offline stage will depend on $\mathcal{N}_\mathrm{t}$.)

In summary, in the formulation we will present, the "user" *can* directly and efficiently monitor and control the accuracy of the very fast reduced basis output relative to the "truth" (FE) output; but the user can *not* (at least in the Online stage) rigorously assess or modify the accuracy of the "truth" output relative to the exact output. However, at least the Online stage of the RB formulation is numerically/mathematically *and* computationally stable as $\mathcal{N}_\mathrm{t} \to \infty$, and hence the "truth" approximation may be chosen *very conservatively* — such that the "exact-truth" error $\varepsilon^{\mathcal{N}_\mathrm{t}}$ is arguably negligibly small compared to the desired accuracy and the (as we shall see, *controllable*) "truth-RB" error. We do not find this state of affairs completely satisfactory (despite Orwellian attempts at spinning the nomenclature), but at present it is the best we can offer. So there.

Finally, we close this section with some nomenclature housekeeping. In particular, we have introduced earlier the "exact" superscript e so that now we can suppress the $\mathcal{N}_\mathrm{t}$ superscript — to significantly simplify the presentation of the RB methodology — yet not risk confusion between the "truth" approximation and the "exact" solution. Thus in what follows the superscript e shall continue to refer to the exact solution; however, *no* superscript shall now refer to the "truth" approximation (except on occasions in which we wish to recall or re-emphasize the $\mathcal{N}_\mathrm{t}$ dependence, or specify a particular "truth" approximation in the examples); and, as we shall see, subscript N shall refer to the reduced basis approximation. Thus X (wherever it appears), $u(\boldsymbol{\mu})$, and $s(\boldsymbol{\mu})$ shall now be understood as $X^{\mathcal{N}_\mathrm{t}}$, $u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu})$, and $s^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu})$, respectively; the basis functions φ_k, $1 \leq k \leq \mathcal{N}_\mathrm{t}$, shall be understood as $\varphi_k^{\mathcal{N}_\mathrm{t}}$, $1 \leq k \leq \mathcal{N}_\mathrm{t}$; $\underline{u}$, $\underline{A}$, and $\underline{F}$ shall be understood as $\underline{u}^{\mathcal{N}_\mathrm{t}}$, $\underline{A}^{\mathcal{N}_\mathrm{t}}$, and $\underline{F}^{\mathcal{N}_\mathrm{t}}$; $\underline{\mathbb{A}}^q$, $1 \leq q \leq Q_a$, $\underline{\mathbb{F}}^q$, $1 \leq q \leq Q_f$, and $\underline{\mathbb{X}}$ shall be understood as $\underline{\mathbb{A}}^{\mathcal{N}_\mathrm{t} q}$, $1 \leq q \leq Q_a$, $\underline{\mathbb{F}}^{\mathcal{N}_\mathrm{t} q}$, $1 \leq q \leq Q_f$, and $\underline{\mathbb{X}}^{\mathcal{N}}$; and similarly for any future "truth" quantities introduced in particular in the context of *a posteriori* error estimation.

Chapter 3

Reduced Basis Approximation

3.1 Overview

As described in the Preface, the Reduced Basis (RB) approach derives from the Opportunities (I) and (II). In particular, although $u(\boldsymbol{\mu})$ ($\equiv u^{\mathcal{N}_{\rm t}}(\boldsymbol{\mu})$) is a member of the space X ($\equiv X^{\mathcal{N}_{\rm t}}$) of typically very high dimension $\mathcal{N}_{\rm t}$, in fact $u(\boldsymbol{\mu})$ perforce resides on the parametrically induced manifold $\mathcal{M}$ ($\equiv \mathcal{M}^{\mathcal{N}_{\rm t}} \equiv \{u^{\mathcal{N}_{\rm t}}(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{D}\}) \equiv \{u(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{D}\}$ of typically quite low dimension. It is thus wasteful to express $u(\boldsymbol{\mu})$ as an arbitrary member of (the very general space) X; rather — presuming $\mathcal{M}$ is sufficiently smooth, a point to which we shall return — we should represent $u(\boldsymbol{\mu})$ in terms of elements of the *ad hoc* space span$\{\mathcal{M}\}$ — Opportunity (I). (In this book, *ad hoc* has a positive connotation.)

The (Lagrange) RB recipe is very simply stated: for any $\boldsymbol{\mu} \in \mathcal{D}$ we approximate $u(\boldsymbol{\mu})$ by a linear combination of N — typically relatively few — precomputed solutions or "snapshots" (on $\mathcal{M}^{\mathcal{N}_{\rm t}}$) $u(\boldsymbol{\mu}^1) \equiv u^1, \ldots, u(\boldsymbol{\mu}^N) \equiv u^N$. (We present below a more general framework that also includes both the Taylor [102, 118] and Hermite [66] RB spaces. However, we shall focus on Lagrange spaces which we contend are perhaps better suited to higher dimensional and more global parameters spaces.) Of course we immediately incur an initial cost of at least N $\underline{A}$-solve operations (see Section 2.3.3 for nomenclature), and thus the RB approach is clearly ill-suited to the single-query or few-query situation; however, in the real-time and many-query context we readily accept this Offline investment in exchange for future asymptotic or deployed/Online reductions in marginal cost — Opportunity (II).

In fact, the recipe is less simple than it first appears, and raises many questions:

1. Given a set of parameters $\boldsymbol{\mu}^n \in \mathcal{D}$, $1 \leq n \leq N$, and associated precomputed solutions $u(\boldsymbol{\mu}^n)$, $1 \leq n \leq N$, how can we best combine these "snapshots" — or more generally, any set of N basis functions — to approximate $u(\boldsymbol{\mu})$ and subsequently $s(\boldsymbol{\mu})$ for any given $\boldsymbol{\mu} \in \mathcal{D}$? We address this "projection" question in Section 3.2.2.

2. How can we ensure a well-conditioned RB algebraic system — in particular given the inevitable asymptotic (with increasing N) colinearity of the Lagrange snapshots? We discuss the necessary constructions and implications in Sections 3.2.1 and 3.2.3, respectively.

3. How can we effect the RB projection such that the Online operation count and storage is independent of $\mathcal{N}_{\mathrm{t}}$? We consider this computational issue in Section 3.3.

4. How should we optimally, or even just "reasonably," choose the parameter points $\boldsymbol{\mu}^n$, $1 \leq n \leq N$, and associated snapshots on $\mathcal{M}$ — or more generally, the best N-dimensional subspace of X — to provide most rapid convergence of the RB approximation to $u(\boldsymbol{\mu})$ and hence $s(\boldsymbol{\mu})$ over the entire parameter domain $\mathcal{D}$? We address this "optimal sampling" question in Section 3.4.

5. Under what hypotheses can we expect the (Lagrange) RB approximation to converge rapidly to $u(\boldsymbol{\mu})$ and $s(\boldsymbol{\mu})$? We discuss this question in Sections 3.2.2 and 3.5; at present, we can only identify the central issues and provide limited theoretical results — substantiated (throughout the book) by significant empirical evidence.

6. Why is RB approximation better than simply connecting the dots — interpolating $s^{\mathcal{N}_{\mathrm{t}}}$: $\mathcal{D} \rightarrow \mathbb{R}$? We consider this question in Section 3.5 as regards approximation, and again in Chapter 4 in the context of *a posteriori* error estimation.

Note that in this Part of the book we address these questions in the context of parametrically coercive compliant problems; however, in many cases the responses remain relevant in the more general settings described later in the book.

3.2 Galerkin Approximation

3.2.1 Spaces and Bases

RB Spaces

We first specify the maximum dimension of the RB spaces, $N_{\max}$. (We shall generally presume that $N_{\max}$ is less than the dimension of span$\{\mathcal{M}\}$.) We then introduce a set of linearly independent functions

$$\xi^n \in X, \qquad 1 \leq n \leq N_{\max} \ , \tag{3.1}$$

in terms of which we define our RB approximation spaces

$$X_N = \mathrm{span}\{\xi^n, 1 \leq n \leq N\}, \qquad 1 \leq N \leq N_{\max} \ . \tag{3.2}$$

We presume — in order that our approximation qualify as "reduced basis" — that the ξ^n are somehow related to the manifold $\mathcal{M}$. By construction we obtain

$$X_N \subset X, \quad \dim(X_N) = N, \qquad 1 \leq N \leq N_{\max} \ , \tag{3.3}$$

and furthermore

$$X_1 \subset X_2 \cdots X_{N_{\max}-1} \subset X_{N_{\max}} \ (\subset X) \ . \tag{3.4}$$

We shall extensively exploit the "nested" or hierarchical property, (3.4), to reduce both the Offline and Online operations and storage.

On occasion we shall consider non-hierarchical RB spaces which we shall denote as X_N^{nh}, $1 \leq n \leq N_{\max}$:

$$X_N^{\mathrm{nh}} \subset X, \quad \dim(X_N^{\mathrm{nh}}) = N, \qquad 1 \leq N \leq N_{\max} \ ; \tag{3.5}$$

these spaces do not lead to particularly efficient RB approximation — in particular as regards Offline effort and Online storage — and will be invoked primarily in the context of theoretical discussions.

In the Lagrange RB recipe, we first introduce a set of parameter points

$$\boldsymbol{\mu}^n \in \mathcal{D}, \qquad 1 \le n \le N_{\max} \ , \tag{3.6}$$

in terms of which we define our associated RB samples

$$S_N \equiv \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N\}, \qquad 1 \le N \le N_{\max} \ ; \tag{3.7}$$

note that these samples are nested — $S_1 \subset S_2 \cdots \subset S_{N_{\max}-1} \subset S_{N_{\max}} \subset \mathcal{D}$. We then introduce our "snapshots"

$$u^n \equiv u(\boldsymbol{\mu}^n), \qquad 1 \le n \le N_{\max} \ , \tag{3.8}$$

(of course in the case of vector-valued fields, $u(\boldsymbol{\mu}^n) = (u_1(\boldsymbol{\mu}^n), \ldots, u_{d_v}(\boldsymbol{\mu}^n))$, $1 \le n \le N_{\max}$) in terms of which we define our Lagrange RB spaces

$$W_N \equiv \mathrm{span}\{u(\boldsymbol{\mu}^n),\ 1 \le n \le N\}, \qquad 1 \le N \le N_{\max} \ ; \tag{3.9}$$

note that (because the samples S_N are nested) these spaces are hierarchical — $W_1 \subset W_2 \cdots W_{N_{\max}-1} \subset W_{N_{\max}}$ ($\subset X \equiv X^{\mathcal{N}_{\rm t}}$). In some rare cases (primarily for theoretical purposes) we shall consider non-hierarchical Lagrange spaces: we shall denote the corresponding samples and spaces as $S_N^{\rm nh}$ and $W_N^{\rm nh}$, respectively.

The Lagrange spaces W_N [118] are a special (but especially important) example of our more general hierarchical spaces X_N for the particular case in which $\xi^n = u(\boldsymbol{\mu}^n)$, $1 \le n \le N_{\max}$. The Taylor and Hermite RB spaces are also (or can also) be particular cases of the general hierarchical spaces X_N: we can generate the Taylor RB spaces [102, 118] by choosing the ξ^n as the field *and* field sensitivity derivatives — derivatives of $u(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$ (see Sections 1.4.2 and 3.5.1) — at a particular parameter point in $\mathcal{D}$; we can generate the "Hermite" RB spaces [66] — a composite of the Lagrange and Taylor ideas — by choosing the ξ^n as the field and sensitivity derivatives at *several* points in $\mathcal{D}$. We shall be particularly focused on Lagrange RB spaces, as these spaces most readily extend to more parameters and larger parameter domains; however, most aspects of our formulation — in particular the Offline-Online decompositions and *a posteriori* error estimation procedures — apply to any hierarchical RB approximation spaces.

In theory we may choose for our Lagrange sample points (3.6) any set of parameter values that induce a linearly independent set of snapshots (3.7). In actual practice, the RB samples S_N and associated Lagrange RB spaces W_N — as well as $N_{\max}$ to achieve the desired error tolerance — are determined by the adaptive procedure described in Section 3.4. This procedure effectively ensures linear dependence (given our assumption $N_{\max} < \dim(\text{span}\{\mathcal{M}\})$. However, if in fact the reduced basis spaces are well chosen then the snapshots *should* approach linear dependence as N increases [57]. (Of course, the converse is not true: the very poorly chosen Lagrange sample $\boldsymbol{\mu}^1 \approx \boldsymbol{\mu}^2 \approx \cdots \approx \boldsymbol{\mu}^{N_{\max}}$ also generates nearly linearly dependent snapshots.) To wit, if the space W_N can already provide a good approximation to any member of $\mathcal{M}$, then the next snapshot $u(\boldsymbol{\mu}^{N+1})$ will perforce contain "much" of $u(\boldsymbol{\mu}^1), \ldots, u(\boldsymbol{\mu}^N)$. We therefore pursue Gram-Schmidt orthogonalization in the $(\cdot,\cdot)_X$ inner product to create a well-conditioned set of basis functions.

Orthogonal RB Basis

In particular, given the ξ^n, $1 \le n \le N_{\max}$, of (3.1) ($u(\boldsymbol{\mu}^n)$, $1 \le n \le N_{\max}$, of (3.8) in the Lagrange case), we construct the basis set $\{\zeta^n\}$, $1 \le n \le N_{\max}$, as

```
$\zeta^1 = \xi^1 / \|\xi^1\|_X;$
for $n = 2\colon N_{\max}$
    $z^n = \xi^n - \sum_{m=1}^{n-1} (\xi^n, \zeta^m)_X \, \zeta^m;$    (3.10)
    $\zeta^n = z^n / \|z^n\|_X;$
end.
```

(In the case of vector-valued fields, $\zeta^n = (\zeta_1^n, \ldots, \zeta_{d_v}^n)$, $1 \leq n \leq N_{\max}$.) As a result of this process we obtain the orthogonality condition

$$(\zeta^n, \zeta^m)_X = \delta_{n\,m}, \qquad 1 \leq n, m \leq N_{\max} \ , \tag{3.11}$$

where $\delta_{n\,m}$ is the Kronecker-delta symbol. The orthogonality condition (3.11) is imperative in ensuring a well-conditioned reduced basis algebraic system. (In fact, (3.11) only obtains in infinite precision: in finite precision, in particular for the higher modes and larger $N_{\max}$, (3.11) will be violated. We can slightly improve the result by full orthogonalization [92, 146]; however, typically the forward process (3.10) suffices and is more intuitive in subsequent adaptive sampling procedures.)

We can express our reduced basis spaces as

$$X_N = \text{span}\{\zeta^n,\ 1 \leq n \leq N\}, \qquad 1 \leq N \leq N_{\max} \ . \tag{3.12}$$

Equivalently, any $w_N \in X_N$ can be expressed as

$$w_N = \sum_{n=1}^{N} w_{N\,n}\, \zeta^n \tag{3.13}$$

for unique

$$w_{N\,n} \in \mathbb{R}, \qquad 1 \leq n \leq N \ . \tag{3.14}$$

Note that, due to our orthogonalization, $w_{N\,n}$ is not (even in the Lagrange case, $X_N = W_N$) the coefficient of the n^{th} snapshot; rather, $w_{N\,n}$ is the coefficient of the "new" contribution of the n^{th} snapshot.

Algebraic Representation of RB Basis

Before proceeding we restate the orthogonalization process above in algebraic terms. We first express our functions in terms of the truth FE approximation basis functions φ_i, $1 \leq i \leq \mathcal{N}_{\rm t}$. In particular,

$$\xi^n(x) = \sum_{i=1}^{\mathcal{N}_{\rm t}} \xi_i^n\, \varphi_i(x), \qquad 1 \leq n \leq N_{\max} \ ; \tag{3.15}$$

we shall denote the vectors of FE basis coefficients as

$$\underline{\xi}^n \equiv [\xi_1^n \ldots \xi_{\mathcal{N}_\mathrm{t}}^n]^\mathrm{T} \in \mathbb{R}^{\mathcal{N}_\mathrm{t}}, \qquad 1 \leq n \leq N_\mathrm{max} \ . \tag{3.16}$$

Similarly, the orthogonal RB basis functions can be expressed as

$$\zeta^n(x) = \sum_{i=1}^{\mathcal{N}_\mathrm{t}} \zeta_i^n \, \varphi_i(x) \ ; \tag{3.17}$$

we denote the vectors of basis coefficients as

$$\underline{\zeta}^n \equiv [\zeta_1^n \ldots \zeta_{\mathcal{N}_\mathrm{t}}^n]^\mathrm{T} \in \mathbb{R}^{\mathcal{N}_\mathrm{t}}, \qquad 1 \leq n \leq N_\mathrm{max} \ . \tag{3.18}$$

We can now succinctly describe the orthogonalization process.

To wit, our algorithm (3.10) can now be expressed as

```
$\underline{\zeta}^1 = \underline{\xi}^1 / \sqrt{(\underline{\xi}^1)^\mathrm{T} \, \mathbb{X} \, \underline{\xi}^1}$
for $n = 2 \colon N_\mathrm{max}$
        $\underline{z}^n = \underline{\xi}^n - \sum_{m=1}^{n-1} ((\underline{\xi}^n)^\mathrm{T} \, \mathbb{X} \, \underline{\zeta}^m) \, \underline{\zeta}^m;$        (3.19)
        $\underline{\zeta}^n = \underline{z}^n / \sqrt{(\underline{z}^n)^\mathrm{T} \, \mathbb{X} \, \underline{z}^n};$
end.
```

Recall that $\mathbb{X}$ is the "inner product" (and induced norm) matrix defined in Chapter 2, (2.45).

Finally, we introduce "basis" matrices $\underline{\mathbb{Z}}_N^{\mathcal{N}_\mathrm{t}} \equiv$ (in our shorthand) $\underline{\mathbb{Z}}_N \in \mathbb{R}^{\mathcal{N}_\mathrm{t} \times N}$, $1 \leq N \leq N_\mathrm{max}$:

$$\mathbb{Z}_{N\, j\, n} = \zeta_j^n, \qquad 1 \leq j \leq \mathcal{N}_\mathrm{t}, \ 1 \leq n \leq N, \ 1 \leq N \leq N_\mathrm{max} \ . \tag{3.20}$$

In essence, the n^th column of $\mathbb{Z}_{N_\mathrm{max}}$ contains the vector of FE basis coefficients associated with the n^th RB basis function. Note that, thanks to the nested/hierarchical nature of our RB spaces, $\underline{\mathbb{Z}}_1$ is a principal submatrix of $\underline{\mathbb{Z}}_2 \cdots$ is a principal submatrix of $\underline{\mathbb{Z}}_{N_\mathrm{max}}$; clearly, we shall store only $\underline{\mathbb{Z}}_{N_\mathrm{max}}$ and then extract the $\underline{\mathbb{Z}}_N$, $1 \leq N \leq N_\mathrm{max}$, submatrices as necessary. We can express the orthogonality condition (3.11) as

$$\mathbb{Z}_{N_\mathrm{max}}^\mathrm{T} \, \mathbb{X} \, \mathbb{Z}_{N_\mathrm{max}} = \mathbb{I}_{N_\mathrm{max}} \ , \tag{3.21}$$

where $\mathbb{I}_M$ is the Identity matrix in $\mathbb{R}^{M \times M}$.

3.2.2 Projection

Not surprisingly, given our hypotheses on a and f $(= \ell)$, standard Galerkin projection is the best discretization choice. We look for $u_{X_N}(\boldsymbol{\mu})$ $(\equiv u^{\mathcal{N}_t}_{X_N}(\boldsymbol{\mu})) \in X_N$ such that

$$a(u_{X_N}(\mu), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \qquad \forall v \in X_N \ ; \tag{3.22}$$

we then evaluate

$$s_{X_N}(\boldsymbol{\mu}) = f(u_{X_N}(\boldsymbol{\mu}); \mu) \ . \tag{3.23}$$

(In the case of vector-valued fields, $u_{X_N}(\boldsymbol{\mu}) \equiv (u_1(\boldsymbol{\mu}), \ldots, u_{d_v}(\boldsymbol{\mu}))_{X_N}$.) In the case of the Lagrange space — $X_N = W_N$ — we shall sometimes explicitly label $u_{X_N}(\boldsymbol{\mu}), s_{X_N}(\boldsymbol{\mu})$ as $u_{W_N}(\boldsymbol{\mu}), s_{W_N}(\boldsymbol{\mu})$. Furthermore, we shall often abbreviate $u_{X_N}(\boldsymbol{\mu}), s_{X_N}(\boldsymbol{\mu})$ or $u_{W_N}(\boldsymbol{\mu}), s_{W_N}(\boldsymbol{\mu})$ as simply $u_N(\boldsymbol{\mu}), s_N(\boldsymbol{\mu})$ in situations in which no ambiguity will arise.

On occasion, and again primarily for theoretical purposes, we will consider non-hierarchical RB approximations. The statement (3.22),(3.23) shall still apply, however now the weak statement is defined over $X^{\rm nh}_N$ (or $W^{\rm nh}_N$ in the case of Lagrange spaces). The corresponding RB field variable and output shall be denoted $u_{X^{\rm nh}_N}(\boldsymbol{\mu}), s_{X^{\rm nh}_N}$ and $u_{W^{\rm nh}_N}(\boldsymbol{\mu}), s_{W^{\rm nh}_N}$ in the case of general non-hierarchical spaces and Lagrange non-hierarchical spaces, respectively. We do not propose such RB approximations as practical numerical approaches due to the increased Offline and Online computational effort — and reduced flexibility — associated with these non–hierarchical approximations. Unless otherwise explicitly stated, our various formulations — in particular computational procedures — are restricted to *hierarchical* approximation spaces.

It is clear from our coercivity and continuity hypotheses on a, our conforming reduced basis space $X_N \subset X$, and our assumption of linear independence of snapshots, that (3.22) admits a unique solution. We can readily demonstrate the usual Galerkin optimality results in

Proposition 3A. *For any $\boldsymbol{\mu} \in \mathcal{D}$ and $u_N(\boldsymbol{\mu})$ and $s_N(\boldsymbol{\mu})$ satisfying (3.22)–(3.23),*

$$|||u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} = \inf_{w_N \in X_N} |||u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \ , \tag{3.24}$$

$$\|u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X \leq \sqrt{\frac{\gamma^\mathrm{e}(\boldsymbol{\mu})}{\alpha^\mathrm{e}(\boldsymbol{\mu})}} \inf_{w_N \in X_N} \|u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - w_N\|_X \ , \tag{3.25}$$

and furthermore

$$\begin{aligned} s^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) &= |||u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||^2_{\boldsymbol{\mu}} \\ &= \inf_{w_N \in X_N} |||u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})|||^2_{\boldsymbol{\mu}} \ , \end{aligned} \tag{3.26}$$

as well as

$$0 < s^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) \leq \gamma^\mathrm{e}(\boldsymbol{\mu}) \inf_{w_N \in X_N} \|u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\|^2_X \ . \tag{3.27}$$

Here $\alpha^\mathrm{e}(\boldsymbol{\mu})$ and $\gamma^\mathrm{e}(\boldsymbol{\mu})$ are the coercivity and continuity constants defined in (2.13) and (2.14), respectively.

Proof. The proof is standard [41], but as the result is central we recall the main ingredients. First, since our reduced basis space is conforming, $X_N \subset X^{\mathcal{N}_\mathrm{t}}$, we obtain Galerkin orthogonality: $a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = 0$, $\forall\, v \in X_N$; here $e(\boldsymbol{\mu}) \equiv u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$ is the error in the reduced basis field approximation. It follows (recall a is symmetric and coercive) that $u_N(\boldsymbol{\mu})$ is in fact the projection of $u^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu})$ in the $a(\cdot, \cdot; \boldsymbol{\mu}) \equiv (((\cdot, \cdot)))_{\boldsymbol{\mu}}$ inner product: the energy norm result (3.24) directly follows. To obtain the X-norm result (3.25) we then apply the energy-norm bound (3.24) and coercivity and continuity. To prove the output results (3.26) we invoke compliance and Galerkin orthogonality — $s^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) = f(e(\boldsymbol{\mu}); \boldsymbol{\mu}) = a(u^{\mathcal{N}_\mathrm{t}}, e(\boldsymbol{\mu}); \boldsymbol{\mu}) = a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu})$ — and then appeal to the energy-norm bound (3.24). Finally, (3.27) follows from (3.26) and continuity. ■

We note that $s_N(\boldsymbol{\mu})$ is a lower bound — in fact, since $s_N(\boldsymbol{\mu}) = a(u_N, (\boldsymbol{\mu}), u_N(\boldsymbol{\mu}); \boldsymbol{\mu})$, a *positive* lower bounds — for $s^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu})$, and that the error in the output is effectively the *square* of the error in the field variable. Note also that our proof in fact also applies to non-hierarchical RB approximation spaces.

3.2.3 Algebraic Equations

We now apply the standard "variational" procedure to determine the linear algebraic set of equations associated with our Galerkin procedure (3.22) and basis functions (3.9). In particular, we first expand

$$u_N(\boldsymbol{\mu}) = \sum_{j=1}^{N} u_{N\,j}(\boldsymbol{\mu})\,\zeta^j \ . \tag{3.28}$$

We next insert (3.28) in (3.22) and choose $v = \zeta^i$, $1 \leq i \leq N$, as our test functions. We thus obtain the set of linear algebraic equations

$$\sum_{j=1}^{N} a(\zeta^j, \zeta^i; \boldsymbol{\mu})\, u_{N\,j}(\boldsymbol{\mu}) = f(\zeta^i; \boldsymbol{\mu}), \qquad 1 \leq i \leq N \ , \tag{3.29}$$

for the reduced basis coefficients $u_{N\,j}$, $1 \leq j \leq N$. (Note in the case of vector-valued fields, the single coefficient $u_{N\,j}$ multiplies all d_v components of the j^{th} basis function ζ^j.) The output can then be expressed as

$$s_N(\boldsymbol{\mu}) = \sum_{j=1}^{N} u_{N\,j}(\boldsymbol{\mu})\, f(\zeta^j; \boldsymbol{\mu}) \ . \tag{3.30}$$

We now express these operations in matrix form.

We first introduce the vector of RB coefficients,

$$\underline{u}_N(\boldsymbol{\mu}) \equiv [u_{N\,1}\, u_{N\,2} \ldots u_{N\,N}]^{\mathrm{T}} \in \mathbb{R}^N \ . \tag{3.31}$$

It then follows from (3.29) that $\underline{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ satisfies

$$\underline{A}_N(\boldsymbol{\mu})\, \underline{u}(\boldsymbol{\mu}) = \underline{F}_N(\boldsymbol{\mu}) \ , \tag{3.32}$$

where the stiffness matrix $\underline{A}_N(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ and "load" or "source" (and "output") vector $\underline{F} \in \mathbb{R}^N$ are given by

$$A_{N\,i\,j}(\boldsymbol{\mu}) = a(\zeta^j, \zeta^i; \boldsymbol{\mu}), \qquad 1 \leq i, j \leq N \ , \tag{3.33}$$

and

$$F_{N\,i}(\boldsymbol{\mu}) = f(\zeta^i; \boldsymbol{\mu}), \qquad 1 \leq i \leq N \ , \tag{3.34}$$

respectively. Finally, the output can now be expressed as

$$s_N(\boldsymbol{\mu}) = \underline{F}_N^{\mathrm{T}}(\boldsymbol{\mu})\, \underline{u}_N(\boldsymbol{\mu}) \ . \tag{3.35}$$

We recall that $^{\mathrm{T}}$ denotes algebraic transpose.

It immediately follows from our assumption of linear independence of the snapshots that the stiffness matrix $\underline{A}_N(\boldsymbol{\mu})$ is symmetric and positive definite. In fact, we can be more precise about the conditioning of our system in

Proposition 3B. *The condition number of $\underline{A}_N(\boldsymbol{\mu})$ is bounded from above by $\gamma^{\mathrm{e}}(\boldsymbol{\mu})/\alpha^{\mathrm{e}}(\boldsymbol{\mu})$, the ratio of the continuity and coercivity constants for the continuous problem.*

Proof. We recall that the condition number of a symmetric positive definite matrix is the ratio of the maximum to minimum eigenvalues of the matrix. To obtain a lower bound for the smallest eigenvalue of $\underline{A}_N(\boldsymbol{\mu})$ we appeal to coercivity and orthogonality (3.11) to note that

$$\begin{aligned}\frac{\underline{w}_N^{\mathrm{T}}\, \underline{A}_N(\boldsymbol{\mu})\, \underline{w}_N}{\underline{w}_N^{\mathrm{T}}\underline{w}_N} &= \frac{a\left(\sum_{n=1}^{N} w_{N\,n}\, \zeta^n,\ \sum_{m=1}^{N} w_{N\,m}\, \zeta^m; \boldsymbol{\mu}\right)}{\underline{w}_N^{\mathrm{T}}\, \underline{w}_N} \\ &\geq \alpha^{\mathrm{e}}(\boldsymbol{\mu})\, \frac{\sum_{n=1}^{N}\sum_{m=1}^{N} w_{N\,n}\, w_{N\,m}(\zeta^n, \zeta^m)_X}{\underline{w}_N^{\mathrm{T}}\, \underline{w}_N} = \alpha^{\mathrm{e}}(\boldsymbol{\mu}), \qquad \forall\, w_N \in \mathbb{R}^N \ ;\end{aligned} \tag{3.36}$$

we then invoke the Rayleigh quotient to conclude that the smallest eigenvalue of $\underline{A}_N(\boldsymbol{\mu})$ is greater than $\alpha^{\mathrm{e}}(\boldsymbol{\mu})$. Similarly, to obtain an upper bound for the largest eigenvalue of $\underline{A}_N(\boldsymbol{\mu})$ we apply continuity and orthogonality to write

$$\begin{aligned}\frac{\underline{w}_N^{\mathrm{T}}\, \underline{A}_N(\boldsymbol{\mu})\, \underline{w}_N}{\underline{w}_N^{\mathrm{T}}\, \underline{w}_N} &= \frac{a\left(\sum_{n=1}^{N} w_{N\,n}\, \zeta^n,\ \sum_{m=1}^{N} w_{N\,m}\, \zeta^m; \boldsymbol{\mu}\right)}{\underline{w}_N^{\mathrm{T}}\, \underline{w}_N} \\ &\leq \gamma^{\mathrm{e}}(\boldsymbol{\mu})\, \frac{\sum_{n=1}^{N}\sum_{m=1}^{N} w_{N\,n}\, w_{N\,m}(\zeta^n, \zeta^m)_X}{\underline{w}_N^{\mathrm{T}}\, \underline{w}_N} = \gamma^{\mathrm{e}}(\boldsymbol{\mu}), \qquad \forall\, w_N \in \mathbb{R}^N \ ;\end{aligned} \tag{3.37}$$

we then invoke the Rayleight quotient to conclude that the largest eigenvalue of $\underline{A}_N(\boldsymbol{\mu})$ is less than $\gamma^{\mathrm{e}}(\boldsymbol{\mu})$. The desired result directly follows. ■

Thus, despite the near linear dependence of the original snapshots, orthogonalization *in the correct inner product* recovers a very well-conditioned reduced basis system. (The latter suggests that, for very large N, iterative RB solution strategies might be of interest; however, the large N limit is obviously of very limited interest.)

3.3 Offline-Online Computational Procedure

3.3.1 Strategy

The reduced basis system (3.32) is clearly of small size — an $N \times N$ set of linear equations. Hence for any new $\boldsymbol{\mu} \in \mathcal{D}$ — *once the RB stiffness matrix* $\underline{A}_N(\boldsymbol{\mu})$ *of (3.32) is formed* — $\underline{u}_N(\boldsymbol{\mu})$ and subsequently $s_N(\boldsymbol{\mu})$ can be obtained from (3.32) in $O(N^3)$ operations and (3.35) in $O(N)$ operations, respectively. However, the formation of the reduced basis stiffness matrix $\underline{A}_N(\boldsymbol{\mu})$ *ostensibly* requires N $\underline{A}^{\mathcal{N}_{\mathrm{t}}}$-matvec and N^2 $X^{\mathcal{N}_{\mathrm{t}}}$-inprod operations — or at least $O(N^2\mathcal{N}_{\mathrm{t}})$ operations even for a sparse finite element system. If we permit this outrage, the Online stage operation count will *not* be independent of $\mathcal{N}_{\mathrm{t}}$, and the RB approach will be only marginally better than classical approaches even in the many-query and real-time contexts.

However, we can in fact restore "Online $\mathcal{N}_{\mathrm{t}}$ independence" by appeal to our assumption of affine parameter dependence, (2.5)–(2.6). It follows directly from application of (2.6) to (3.33) and (2.5) to (3.34) that our stiffness matrix and load vector can be expressed as

$$a(\zeta^n, \zeta^m; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\, a^q(\zeta^n, \zeta^m), \qquad 1 \leq m, n \leq N \ , \tag{3.38}$$

and

$$f(\zeta^n; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu})\, f^q(\zeta^n), \qquad 1 \leq n \leq N \ , \tag{3.39}$$

respectively. (We already identified an analogous decomposition in the finite element context, (2.41)–(2.43).) The Offline-Online decomposition is now clear.

Offline we form the *parameter-independent* matrices $\underline{\mathbb{A}}_N^q \in \mathbb{R}^{N\times N}$, $1 \le q \le Q_a$,

$$\underline{\mathbb{A}}^q_{N\,n\,m} = a^q(\zeta^n, \zeta^m), \qquad 1 \le n, m \le N,\ 1 \le q \le Q_a \ , \tag{3.40}$$

and *parameter-independent* vectors $\underline{\mathbb{F}}_N^q \in \mathbb{R}^N$, $1 \le q \le Q_f$,

$$\underline{\mathbb{F}}^q_{N\,n} = f^q(\zeta^n), \qquad 1 \le n \le N,\ 1 \le q \le Q_f \ ; \tag{3.41}$$

the operation count (provided in detail below) will be $\mathcal{N}_\mathrm{t}$-*dependent* — and hence very *expensive*. *Online*, for any given $\boldsymbol{\mu} \in \mathcal{D}$, we then assemble the RB stiffness matrix and load vector as

$$\underline{A}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta^q_a(\boldsymbol{\mu})\, \underline{\mathbb{A}}^q_N \ , \tag{3.42}$$

and

$$\underline{F}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta^q_f(\boldsymbol{\mu})\, \underline{\mathbb{F}}^q_N \ ; \tag{3.43}$$

the operation count (provided in detail below) and storage will now be $\mathcal{N}_\mathrm{t}$-*independent* — and hence very *inexpensive*.

Before discussing the detailed operation count it shall prove convenient to express our RB matrices and vectors in terms of the corresponding truth FE matrices and vectors: the former are linked to the latter via the basis matrices $\underline{\mathbb{Z}}_N$, $1 \le N \le N_\mathrm{max}$, of (3.20). In particular, from (3.33) we obtain

$$a(\zeta^n, \zeta^m; \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}_\mathrm{t}} \sum_{j=1}^{\mathcal{N}_\mathrm{t}} \zeta^m_i \, a(\varphi_i, \varphi_j; \boldsymbol{\mu})\, \zeta^n_j, \qquad 1 \le n, m \le N \ , \tag{3.44}$$

which from (2.39) and (3.20) can be expressed as

$$\underline{A}_N(\boldsymbol{\mu}) = \underline{\mathbb{Z}}_N^\mathrm{T}\, \underline{A}(\boldsymbol{\mu})\, \underline{\mathbb{Z}}_N \ ; \tag{3.45}$$

similarly, from (3.40), (2.42), and (3.20) we obtain

$$\underline{\mathbb{A}}^q_N = \underline{\mathbb{Z}}_N^T\, \underline{\mathbb{A}}^q\, \underline{\mathbb{Z}}_N, \qquad 1 \le q \le Q_a \ . \tag{3.46}$$

The load vector permits an analogous treatment:

$$\underline{F}_N(\boldsymbol{\mu}) = \underline{\mathbb{Z}}_N^T\, \underline{F}(\boldsymbol{\mu})\, \underline{\mathbb{Z}}_N \ , \tag{3.47}$$

and

$$\underline{\mathbb{F}}^q_N = \underline{\mathbb{Z}}_N^T \, \mathbb{F}^q \, \underline{\mathbb{Z}}_N, \qquad 1 \le q \le Q_f \ . \tag{3.48}$$

All these quantities are defined for $1 \le N \le N_{\max}$.

3.3.2 Operation Count and Storage

We now can succinctly describe the Offline and Online stages and provide associated operation counts (or at least operations) and storage. In the Offline stage, we must first compute the matrix $Z_{N_{\max}} \in \mathbb{R}^{\mathcal{N}_{\mathrm{t}} \times N_{\max}}$ — $N_{\max}$ $\underline{A}$-solve operations (recall $\underline{A}$ is shorthand for the FE stiffness matrix $\underline{A}^{\mathcal{N}_{\mathrm{t}}}$). Next, we must form the RB parameter-independent matrices $\underline{\mathbb{A}}^q_{N_{\max}}$, $1 \le q \le Q_a$, from (3.46) and vectors $\underline{\mathbb{F}}^q_{N_{\max}}$, $1 \le q \le Q_f$, from (3.48) — $Q_a N_{\max}$ $\underline{A}$-matvec, $Q_a N_{\max}^2$ $X^{\mathcal{N}_{\mathrm{t}}}$-inprod, and $Q_f N_{\max}$ $X^{\mathcal{N}_{\mathrm{t}}}$-inprod operations, respectively.

The link between the Offline and Online stages is the "permanent storage" of quantities computed in the Offline stage and then invoked in the Online stage. The items that must be "permanently" stored, in essence the Online storage, are the $\underline{\mathbb{A}}^q_{N_{\max}}$, $1 \le q \le Q_a$ — $Q_a N_{\max}^2$ words — and the $\underline{\mathbb{F}}^q_{N_{\max}}$, $1 \le q \le Q_f$ — $Q_f N_{\max}$ words: note the Online storage is independent of $\mathcal{N}_{\mathrm{t}}$. It is crucial to note that, just as the RB spaces are hierarchical, so too are the reduced basis matrices (and vectors): for $1 \le N \le N_{\max}$, the $\underline{\mathbb{A}}^q_N$, $1 \le q \le Q_a$, are principal $N \times N$ submatrices of the $N_{\max} \times N_{\max}$ matrix $\underline{\mathbb{A}}^q_{N_{\max}}$, and the $\underline{\mathbb{F}}^q_N$, $1 \le q \le Q_f$, are principal N subvectors of the $N_{\max}$ vector $\underline{\mathbb{F}}^q_{N_{\max}}$. Thus we need only store $\underline{\mathbb{A}}^q_{N_{\max}}$, $1 \le q \le Q_a$, and $\underline{\mathbb{F}}^q_{N_{\max}}$, $1 \le q \le Q_f$, and then simply extract (in the Online stage) the necessary submatrices and subvectors for the desired N (related to the particular accuracy of interest). The hierarchical structure greatly reduces the requisite storage — a full factor of $N_{\max}$. (The hierarchical structure can also play a role in efficient Online adaptivity, as described in Chapters 4 and 5.)

In the Online stage, we need only assemble $\underline{A}_N(\boldsymbol{\mu})$ from (3.42) and $\underline{F}(\boldsymbol{\mu})$ from (3.43) — $Q_a N^2$ and $Q_f N$ operations, respectively. Subsequently we solve the RB linear algebraic system (3.32) — $O(N^3)$ operations: note that the RB matrix is in general full. (The latter is one of many incentives for effective sampling and *a posteriori* error estimation: given the rapid increase in computational effort with N, we must find the smallest — or least a small — N that satisfies our error tolerance. This is discussed further in Section 3.4 and Chapter 4.) Finally, we evaluate the output from (3.35) — N operations. *The critical point is that the operation count and storage of the Online stage is independent of* $\mathcal{N}_{\mathrm{t}}$. This provides the extremely rapid evaluation $\boldsymbol{\mu} \to s_N(\boldsymbol{\mu})$ — the greatly reduced marginal cost — so crucial in the real-time and many-query contexts. This $\mathcal{N}_{\mathrm{t}}$-independence of the Online operation count and storage also permits us to choose our "truth" approximation very conservatively with no penalty in terms of Online/deployed performance.

3.4 Sampling Strategy

Given hierarchical RB spaces X_N, $1 \leq N \leq N_{\max}$, we can now efficiently — thanks to the Offline-Online decomposition of Section 3.3 — and computationally stably — thanks to the orthonormal basis and Proposition 3B of Section 3.2.3 — determine the "best" combination of snapshots — thanks to our Galerkin projection and Proposition 3A of Section 3.2.2. However, it remains to determine good RB spaces.

We shall see in Section 3.5 that, for parametrically (and no doubt also more generally) coercive problems, it is possible for one-dimensional parameter domains ($P = 1$) to determine generic *a priori* "quasi–hierarchical" (Lagrange) spaces that provide (if not optimal at least) very rapid convergence. However, in higher parameter dimensions (and for noncoercive problems), no such recipes are available: note that tensor product approaches are prohibitively expensive typically even for $P = 3$ and certainly for larger P, yielding Offline and even Online complexity and storage that increases exponentially with P. More generally (and even for $P = 1$), we prefer *ad hoc* or "adaptive" and truly hierarchical spaces which — just as the associated RB approximation — are automatically tailored to the particular problem of interest. The latter can yield approximations that are quite efficient even for modestly large P.

For our discussions below we shall first introduce a "train" sample $\Xi_{\text{train}} \equiv \{\boldsymbol{\mu}^1_{\text{train}}, \ldots, \boldsymbol{\mu}^{n_{\text{train}}}_{\text{train}}\}$ $\subset \mathcal{D}$ consisting of n_{train} distinct parameter points in $\mathcal{D}$: Ξ_{train} shall be a finite dimensional surrogate for $\mathcal{D}$; clearly, at least for larger P, we must anticipate that n_{train} should be quite large — in our applications, easily as large as or larger than 10^6. We also require two quantities related to error control and the size of our RB spaces: $\varepsilon_{\text{tol,min}}$, the smallest anticipated truth-RB error tolerance over $\mathcal{D}$ (in norms to be specified); and $\overline{N}_{\max}$, an upper limit to the maximum dimension of the hierarchical RB spaces.

3.4.1 Kolmogorov N-Width

Before embarking on more practical sampling procedures it is useful to establish a benchmark — even if abstract and not particularly computable — relative to which we can measure progress. In particular, related to Question 4 of 3.1, we can ask "What is the *best* N-dimensional subspace to approximate $u(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{D}$?" We thus define the Kolmogorov "N-width" [69, 74, 116] $\overline{\overline{\varepsilon}}{}^{\text{Kol}}_N$:

$$\overline{\overline{\varepsilon}}{}^{\text{Kol}}_N \equiv \sup_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \inf_{w_N \in X^{\text{Kol}}_N} \|u(\boldsymbol{\mu}) - w_N\|_X \ , \tag{3.49}$$

where the optimal "Kolmogorov spaces" X^{Kol}_N are given by

$$X^{\text{Kol}}_N = \arg \inf_{\text{spaces } X^{\text{nh}}_N \subset X \text{ of dimension } N} \left(\sup_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \inf_{w_N \in X^{\text{nh}}_N} \|u(\boldsymbol{\mu}) - w_N\|_X \right) \ . \tag{3.50}$$

(If we wish to ensure that strictly speaking the X^{Kol}_N qualify as "proper" RB spaces, we can replace X in (3.50) with span$\{\mathcal{M}\}$. However, it is demonstrated in [88] that the resulting convergence rate is little degraded — not surprising since we anticipate that the objective function in (3.50) will naturally prefer span$\{\mathcal{M}\}$.) Roughly, the Kolmogorov N-width indicates how well we can hope to approximate our field variable given the freedom to choose any sequence of "RB" approximation subspaces; note in general the X^{Kol}_N will not be hierarchical.

More precisely, we note that the Kolmogorov N-width is (here) defined relative to the "$L^\infty(\Xi_{\text{train}})$" norm in parameter of the $X(\Omega)$ norm of the "best fit" to the field variable.

The former is clearly (desirably) strong — worst-case analysis; the latter is directly related to the actual RB error from Proposition 3A, (3.25). However, the method is not of any direct practical value. First, in terms of deliverables, the optimal spaces X_N^{Kol} can not be assumed to be hierarchical. Second, in terms of (Offline) computational expense, (*i*) the optimization is combinatorially difficult, and in any event (*ii*) $O(n_{\text{train}})$ $\underline{A}^{\mathcal{N}_{\text{t}}}$-solve are required.

3.4.2 A POD Approach

The Proper Orthogonal Decomposition (POD) or Karhunen-Loève (KL) [72, 80] approach to sampling [75, 77, 97, 142] is immensely popular in a variety of contexts — turbulent flows [85], fluid-structure interaction [46], non-linear structural mechanics [73], turbo-machinery flows [156] — most notably in time-domain Reduced Order Modeling (ROM) [8, 14, 31, 38, 39, 40, 93, 114, 126, 127, 128, 129, 143, 151, 152]. The technique can also be applied within the parametric context [31, 40, 59, 86], as we now describe.

In particular, we now consider a slightly different space optimization problem, in which we effectively replace the $L^\infty(\Xi_{\text{train}})$ norm of (3.49) with the weaker (discrete) $L^2(\Xi_{\text{train}})$ norm:

$$\overline{\overline{\varepsilon}}_N^{\text{POD}} \equiv \sqrt{\frac{1}{n_{\text{train}}} \sum_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \inf_{w_N \in X_N^{\text{POD}}} \|u(\boldsymbol{\mu}) - w_N\|_X^2} \ , \tag{3.51}$$

where the optimal "POD spaces" X_N^{POD} are given by

$$X_N^{\text{POD}} = \arg \inf_{\text{spaces } X_N \subset \text{ span } \{u(\boldsymbol{\mu}_{\text{train}}^n),\, 1 \le n \le n_{\text{train}}\}} \left(\frac{1}{n_{\text{train}}} \sum_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \inf_{w_N \in X_N} \|u(\boldsymbol{\mu}) - w_N\|_X^2 \right) . \tag{3.52}$$

(Clearly the POD spaces are proper RB spaces defined on span$\{\mathcal{M}\}$. Note, however, that the POD spaces are not in general Lagrange, as snapshots can be mixed — arguably an *advantage*.) The remarkably beautiful result is that, unlike for the Kolmogorov N-width optimization, the POD optimization yields hierarchical spaces at non-combinatorial (Offline) cost.

In particular, we can construct the POD spaces through an equivalent symmetric positive semidefinite eigenproblem [37, 58]. (We apply here the method of "snapshots" since, although

n_{train} will be large, we also anticipate the limit $\mathcal{N}_{\text{t}} \to \infty$.) In particular, we first form the correlation matrix $\underline{C}^{\text{POD}} \in \mathbb{R}^{n_{\text{train}} \times n_{\text{train}}}$ given by

$$C^{\text{POD}}_{i\,j} = \frac{1}{n_{\text{train}}} \left(u(\boldsymbol{\mu}^i_{\text{train}}), u(\boldsymbol{\mu}^j_{\text{train}}) \right)_X , \qquad 1 \leq i, j \leq n_{\text{train}} \ , \tag{3.53}$$

which can of course be readily expressed in terms of the FE basis coefficients of the $u(\boldsymbol{\mu}^{\bullet}_{\text{train}})$, $\underline{u}(\boldsymbol{\mu}^{\bullet}_{\text{train}})$, as

$$C^{\text{POD}}_{i\,j} = \frac{1}{n_{\text{train}}} \left(\underline{u}(\boldsymbol{\mu}^i_{\text{train}}) \right)^{\text{T}} \, \underline{\mathbb{X}} \, \underline{u}(\boldsymbol{\mu}^j_{\text{train}}) \ . \tag{3.54}$$

We then look for the eigenpairs ($\underline{\psi}^{\text{POD},k} \in \mathbb{R}^{n_{\text{train}}}, \lambda^{\text{POD},k} \in \mathbb{R}_{+0}$), $1 \leq k \leq n_{\text{train}}$, satisfying

$$\underline{C}^{\text{POD}} \, \underline{\psi}^{\text{POD},k} = \lambda^{\text{POD},k} \underline{\psi}^{\text{POD},k}, \quad (\underline{\psi}^{\text{POD},k})^{\text{T}} \, \underline{\mathbb{X}} \, \underline{\psi}^{\text{POD},k} = 1 \ . \tag{3.55}$$

We arrange the eigenvalues in *descending* order: $\lambda^{\text{POD},1} \geq \lambda^{\text{POD},2} \geq \cdots \lambda^{\text{POD},n_{\text{train}}} \geq 0$.

We now identify $\Psi^{\text{POD},k} \in X$, $1 \leq k \leq n_{\text{train}}$, as

$$\Psi^{\text{POD},k} \equiv \sum_{m=1}^{n_{\text{train}}} \psi^{\text{POD},k}_m \, u(\boldsymbol{\mu}^m_{\text{train}}), \qquad 1 \leq k \leq n_{\text{train}} \ ; \tag{3.56}$$

we further define $N_{\max}$ as the smallest N such that

$$\left(\overline{\overline{\varepsilon}}{}^{\text{POD}}_N \equiv \right) \sqrt{\sum_{k=N+1}^{n_{\text{train}}} \lambda^{\text{POD},k}} \leq \varepsilon_{\text{tol,min}} \ . \tag{3.57}$$

We then construct our POD RB spaces as [58]

$$X^{\text{POD}}_N = \text{span}\{\Psi^{\text{POD},n},\, 1 \leq n \leq N\}, \qquad 1 \leq N \leq N_{\max} \ ; \tag{3.58}$$

in other words, $X^{\text{POD}}_N = X_N$ (our general hierarchical spaces) for the particular choice $\xi^n = \Psi^{\text{POD},n}$, $1 \leq n \leq N$.

We furthermore note from the usual mutual orthogonality properties of symmetric eigenproblems, and our particular normalization in (3.55), that

$$(\Psi^{\text{POD},n}, \Psi^{\text{POD},m})_X = \delta_{n\,m}, \qquad 1 \leq n, m \leq n_{\text{train}} \ , \tag{3.59}$$

and hence ($\Psi^{\mathrm{POD},n} \equiv$) $\xi^n = \zeta^n$, $1 \leq n \leq N_{\max}$: we automatically obtain the orthonormalization of Section 3.2.1. It follows that the $\mathbb{Z}_N \in \mathbb{R}^{\mathcal{N}_{\mathrm{t}} \times N}$ matrices are given by

$$\mathbb{Z}_{N\,j\,n} = \sum_{m=1}^{n_{\mathrm{train}}} \psi_m^{\mathrm{POD},n}\, u_j(\boldsymbol{\mu}_{\mathrm{train}}^m), \qquad 1 \leq j \leq \mathcal{N}_{\mathrm{t}},\ 1 \leq n \leq N,\ 1 \leq N \leq N_{\max}\ , \tag{3.60}$$

where the $u_j(\boldsymbol{\mu}_{\mathrm{train}}^\bullet)$ are the FE basis coefficients of the $u(\boldsymbol{\mu}_{\mathrm{train}}^\bullet)$. We can then directly apply the discrete equations and Offline-Online procedures of Section 3.3.2.

From the point of view of deliverables, the POD improves upon the Kolmogorov framework by providing *hierarchical* spaces — at the only slight disadvantage of a slightly weaker norm over Ξ_{train}; this constitutes a significant advance. Unfortunately, from the perspective of Offline expense, although the POD procedure is no longer combinatorial in nature — a major improvement relative to Kolmogorov — the POD remains extremely expensive: we must still perform $n_{\mathrm{train}} \underline{A}^{\mathcal{N}_{\mathrm{t}}}$-solve and $n_{\mathrm{train}}^2 X^{\mathcal{N}_{\mathrm{t}}}$-inprod operations just to form $\underline{C}^{\mathrm{POD}}$; and we must subsequently solve the rather large eigenproblem (3.55) (though typically the larger eigenvalues are well separated).

Not surprisingly, the POD has found most application in the time-domain [8, 154, 155, 157] — a single dimension — in which n_{train} typically remains quite small. (Furthermore, the interactions between the solution at different times is nicely captured by the global nature of the POD optimization; in this context, the greedy approach described below is not as successful, and hence in the parabolic context we will consider both greedy [53, 56] and combined greedy-POD [60] concepts.) Application in the parameter domain is more rare [31, 40, 59, 86], in particular for larger P for which n_{train} must be quite large and hence the Offline POD expense prohibitive. (The latter can be somewhat reduced by a clustering pre-processing of the snapshots — for example, centroidal Voronoi tesselations [33, 47] — to remove redundant information from Ξ_{train}.)

3.4.3 A Greedy Approach: $\boldsymbol{W_N^*}$

For the greedy approach we will describe shortly we shall need a sharp, rigorous, and efficient bound $\Delta_{X_N}(\boldsymbol{\mu})$ for the RB error $\|u(\boldsymbol{\mu}) - u_{X_N}(\boldsymbol{\mu})\|_X$, where u_{X_N} is our RB approximation associated with the space X_N. To quantity sharp and rigorous, we introduce the effectivity

$$\eta_N(\boldsymbol{\mu}) \equiv \frac{\Delta_{X_N}(\boldsymbol{\mu})}{\|u(\boldsymbol{\mu}) - u_{X_N}(\boldsymbol{\mu})\|_X} \ ; \tag{3.61}$$

we then require

$$1 \leq \eta_N(\boldsymbol{\mu}) \leq \eta_{\max,\mathrm{UB}}, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}, \quad 1 \leq N \leq N_{\max} \ , \tag{3.62}$$

where $\eta_{\max,\mathrm{UB}}$ is finite and independent of N. In essence, the left inequality insists that $\Delta_{X_N}(\boldsymbol{\mu})$ is never less than the true error — *rigor*; the right inequality insists that $\Delta_{X_N}(\boldsymbol{\mu})$ is not too much larger than the true error — *sharpness*. To quantify "*efficient*," we require that in the limit of many evaluations the marginal cost (and hence asymptotic average) cost to evaluate $\boldsymbol{\mu} \to \Delta_{X_N}(\boldsymbol{\mu})$ is *independent of* $\mathcal{N}_\mathrm{t}$. In Chapter 4 we shall develop error estimators Δ_{X_N} that satisfy all these requirements.

Our greedy procedure is intimately connected to (and thus this subsection is restricted to) the Lagrange RB approximation subspace W_N. We shall denote the particular "optimal" (nested) samples and (hierarchical) spaces selected by our greedy algorithm as

$$S_N^* \equiv \{\boldsymbol{\mu}^{1*}, \ldots, \boldsymbol{\mu}^{N*}\}, \qquad 1 \leq N \leq N_{\max} \ , \tag{3.63}$$

and

$$X_N^* \ (\equiv W_N^*) = \mathrm{span}\{u(\boldsymbol{\mu}^{1*}), \ldots, u(\boldsymbol{\mu}^{N*})\}, \qquad 1 \leq N \leq N_{\max} \ , \tag{3.64}$$

respectively. The corresponding "optimal" RB approximation will thus be denoted, at least in this section, as $u_{X_N^*}(\boldsymbol{\mu})$. (Clearly the "optimal" approximation still depends on, and we will specify in all cases, the choice of Ξ_{train} and other algorithmic design variables.)

We presume that we are given some initial $N_0 \in [1, \ldots, \overline{N}_{\max}]$, where $\overline{N}_{\max}$ is an upper bound for $N_{\max}$; we are also given an initial sample $S_{N_0}^* = \{\boldsymbol{\mu}^{1*}, \ldots, \boldsymbol{\mu}^{N_0*}\}$ and associated Lagrange space $X^*_{N_0} = W^*_{N_0} = \text{span}\{u(\boldsymbol{\mu}^{n*}),\ 1 \le n \le N_0\}$; finally, we specify our train sample Ξ_{train} and tolerance $\varepsilon_{\text{tol,min}}$. (Often we shall set $N_0 = 1$ and choose $\boldsymbol{\mu}^{1*}$ (say) randomly; however, the flexibility of an "arbitrary" initialization will permit more advanced sampling sequences, as we discuss below.) We denote the resulting greedy algorithm Greedy(N_0, $S^*_{N_0}$, Ξ_{train}, $\varepsilon_{\text{tol,min}}$).

The algorithm proceeds as follows (note we provisionally set $N_{\max} = \overline{N}_{\max}$):

```
for $N = N_0 + 1 \colon \overline{N}_{\max}$
        $\boldsymbol{\mu}^{N*} = \arg \max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_{X^*_{N-1}}(\boldsymbol{\mu})$;
        $\varepsilon^*_{N-1} = \Delta_{X^*_{N-1}}(\boldsymbol{\mu}^{N*})$;
        if $\varepsilon^*_{N-1} \le \varepsilon_{\text{tol,min}}$
                $N_{\max} = N - 1$;
                exit;
        end;
        $S^*_N = S^*_{N-1} \cup \boldsymbol{\mu}^{N*}$;
        $X^*_N = X^*_{N-1} + \text{span}\{u(\boldsymbol{\mu}^{N*})\}$;
end.
```
(3.65)

We also introduce

$$\overline{\varepsilon}^*_N = \max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \|u(\boldsymbol{\mu}) - u_{X_N}(\boldsymbol{\mu})\|_X, \qquad 1 \le N \le N_{\max}\ , \tag{3.66}$$

which measures the maximum *true* error (not the error *bound*) for the sequence of greedy spaces; in actual practice, we never compute $\overline{\varepsilon}^*_N$ — except in the theoretical context to understand the performance of greedy algorithm relative to other approaches. (Note by "exit" we refer to early termination of the algorithm — when the space is rich enough to achieve the desired minimum error tolerance.)

From the point of view of deliverables, the greedy algorithm provides both hierarchical spaces and in the stronger $L^\infty(\Xi_{\text{train}})$ norm in parameter — two very important attributes. Also, as we discuss below, the low cost of the greedy formulation permits a very exhaustive search — large n_{train} — with corresponding high quality approximation spaces in particular for $P > 1$. The greedy approach is admittedly a short-horizon heuristic that will be sub-optimal with respect to the global (albeit non-hierarchical) Kolmogorov framework; however, it is demonstrated in [30] that, if $\overline{\overline{\varepsilon}}_N^{\text{Kol}}$ *decreases exponentially and sufficiently rapidly with* N, then $\overline{\varepsilon}_N^*$ will also *decrease exponentially with* N.

We shall substantiate the good convergence properties of the greedy spaces in numerous numerical exercises. Typically, or at least in the few test cases we shall report (see Section 3.5.2, Numerical Results), the short-horizon greedy and global POD approaches will in fact perform commensurately if measured in comparable norms; as might be expected, each is slightly better in the "native" norm over Ξ_{train} which defines the respective objective function — $L^2(\Xi_{\text{train}})$ for POD, and $L^\infty(\Xi_{\text{train}})$ for greedy. The success of the greedy approach certainly originates at least in part from the absence of interactions between the RB approximations for different parameter values; in the time-domain context, there are of course interactions between different times and as a result the greedy algorithm [56] may not perform as well as the more global POD optimization procedures [60]. We return to this point in Part IV.

From the perspective of Offline cost, the greedy approach is much more efficient than either the Kolmogorov or POD approaches. This permits either less Offline expense or (typically) much larger n_{train} and hence improved RB approximation spaces and ultimately RB convergence; the effect is particularly pronounced for $P > 1$. Relative to the Kolmogorov framework, the *sequential* greedy "relaxation" is of algebraic rather than of combinatorial complexity. Relative to the POD framework, the greedy approach replaces most FE "truth" computations with inexpensive error bound evaluations: *we compute truth solutions/snapshots not for all the points in* Ξ_{train}, *as in the POD context, but only for the "winning candidates"* $\boldsymbol{\mu}^{n*}$, $1 \leq n \leq N_{\max}$; since $N_{\max} \ll n_{\text{train}}$, the computational savings can be very large. (It is perhaps also possible, but less obvious how, to incorporate inexpensive error bounds within the POD context.)

As we shall elaborate further in Chapter 4, the greedy operation count is roughly (assuming $Q_f \ll Q_a$)

$$\begin{aligned} N_{\max}\, \underline{A}^{\mathcal{N}_{\rm t}}\text{-solve} + N_{\max}\, Q_a\, \underline{A}^{\mathcal{N}_{\rm t}}\text{-matvec} + N_{\max}^2\, Q_a\, X^{\mathcal{N}_{\rm t}}\text{-inprod} & \\ + \underline{\mathbb{X}}\text{-solve}(N_{\max} Q_a) + (N_{\max}^2 Q_a^2)\, X^{\mathcal{N}_{\rm t}}\text{-inprod} & \\ + n_{\rm train}\, O(N_{\max}^4 + N_{\max}^3\, Q_a^2)\ ; & \end{aligned} \tag{3.67}$$

the first line relates to RB formation, the second line to error bound formation, and the third line to the $N_{\max}$ searches over $\Xi_{\rm train}$. (There is also a "short-term" memory requirement of $O(N_{\max} Q_a \mathcal{N}_{\rm t})$ in addition to the "permanent" storage required by the Online stage.) In the preceding discussion, we have focused on the $N_{\max}\, \underline{A}^{\mathcal{N}_{\rm t}}$-solve and $(N_{\max}^2 Q^2) X^{\mathcal{N}_{\rm t}}$-inprod of the greedy (with error bounds) *versus* the $n_{\rm train}\, \underline{A}^{\mathcal{N}_{\rm t}}$-solve and $n_{\rm train}^2\, X^{\mathcal{N}_{\rm t}}$-inprod of the POD approach. However, the $n_{\rm train}\, O(N_{\max}^4 + N_{\max}^3\, Q_a^2)$ of the greedy, related to searches over $\Xi_{\rm train}$, can also be problematic. There are several ways to mitigate this effect.

First, and perhaps easiest [137, 138] is to first run the greedy with $N_0 = 1$ and a relatively *coarse* train sample $\Xi_{\rm train}^{\rm coarse}$ — Greedy(N_0, $\boldsymbol{\mu}^{1*}$, $\Xi_{\rm train}^{\rm coarse}$, $\varepsilon_{\rm tol,min}$) — to obtain $N_{\max}^{\rm coarse}$ and $S^*_{N_{\max}^{\rm coarse}}$, $X^*_{N_{\max}^{\rm coarse}}$ $(= W^*_{N_{\max}^{\rm coarse}})$. We then again run the greedy but now initialized with $S^*_{N_{\max}^{\rm coarse}}$ *and* the desired *fine* train sample $\Xi_{\rm train}^{\rm fine}$ — Greedy($N_{\max}^{\rm coarse}$, $S^*_{N_{\max}^{\rm coarse}}$, $\Xi_{\rm train}^{\rm fine}$, $\varepsilon_{\rm tol,min}$) — to obtain the "final" $N_{\max}^{\rm fine}$ and $S_{N_{\max}^{\rm fine}}$, $X^*_{N_{\max}^{\rm fine}}$ $(= W^*_{N_{\max}^{\rm fine}})$. The hope is to reduce the number of greedy cycles on the fine train sample: in theory, the coarse sample does the work — many cycles — and the final sample does the confirmation — a few cycles. Second, it may be possible to even further reduce the cost associated with the determination of the largest error bound over $\Xi_{\rm train}$ (or in fact $\mathcal{D}$) by considering more sophisticated non-ennumerative optimization procedures [32, 20]; however, the error and error bound are highly oscillatory over $\mathcal{D}$, and hence multi-start (or other global) strategies may limit the possible gains.

3.4.4 A Greedy Approach: $W_N^{\rm out,*}$

We now present a form of the greedy approach particularly well-suited to the coercive compliant case: in particular, we shall replace the error bound for the X norm of Section 3.4.3 with the

the error bound for the energy norm. We also now include in our formulation positive functions $\omega_N \colon \mathcal{D} \to \mathbb{R}$, $1 \le N \le \overline{N}_{\max}$, that permit non-uniform weighting or "importance" over $\mathcal{D}$.

We shall now need a sharp, rigorous, and efficient bound $\Delta^{\text{en}}_{X_N}(\boldsymbol{\mu})$ for the RB error $|||u(\boldsymbol{\mu}) - u_{X_N}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$, where u_{X_N} is our RB approximation associated with the space X_N; we note from Proposition 3A, (3.26), that $\Delta^{\text{en}}_{X_N}(\boldsymbol{\mu})$ is also an upper bound for $\sqrt{s(\boldsymbol{\mu}) - s_{X_N}(\boldsymbol{\mu})}$. To quantity sharp and rigorous, we introduce the effectivity

$$\eta^{\text{en}}_N(\boldsymbol{\mu}) \equiv \frac{\Delta^{\text{en}}_{X_N}(\boldsymbol{\mu})}{|||u(\boldsymbol{\mu}) - u_{X_N}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}} \quad \left(= \frac{\Delta^{\text{en}}_{X_N}(\boldsymbol{\mu})}{\sqrt{s(\boldsymbol{\mu}) - s_{X_N}(\boldsymbol{\mu})}} \right) \ . \tag{3.68}$$

We then require

$$1 \le \eta^{\text{en}}_N(\boldsymbol{\mu}) \le \eta^{\text{en}}_{\max,\text{UB}}, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D},\ 1 \le N \le N_{\max} \ , \tag{3.69}$$

where $\eta^{\text{en}}_{\max,\text{UB}}$ is finite and independent of N. As before, we shall further require that in the limit of many evaluations the marginal cost (and hence asymptotic average) cost to evaluate $\boldsymbol{\mu} \to \Delta^{\text{en}}_{X_N}(\boldsymbol{\mu})$ is independent of $\mathcal{N}_{\text{t}}$. In Chapter 4 we shall develop error estimators $\Delta^{\text{en}}_{X_N}$ that satisfy all these requirements.

We shall denote the particular optimal (nested) samples and (hierarchical) spaces selected by the greedy algorithm as

$$S^{\text{out},*}_N = \{\boldsymbol{\mu}^{1\,\text{out},*}, \dots, \boldsymbol{\mu}^{N\,\text{out},*}\}, \qquad 1 \le N \le N_{\max} \ , \tag{3.70}$$

and

$$X^{\text{out},*}_N \ (= W^{\text{out},*}_N) = \text{span}\{u(\boldsymbol{\mu}^{1\,\text{out},*}), \dots, u(\boldsymbol{\mu}^{N\,\text{out},*})\}, \qquad 1 \le N \le N_{\max} \ , \tag{3.71}$$

respectively, The corresponding "optimal" RB approximation will thus be denoted, at least in this section, as $u_{X^{\text{out},*}_N}$. Our algorithm $\text{Greedy}^{\text{out}}(N_0, S^{\text{out},*}_{N_0}, \Xi_{\text{train}}, \varepsilon_{\text{tol,min}}, \omega_N)$ then preceeds as follows (we provisionally set $N_{\max} = \overline{N}_{\max}$):

```
for $N = N_0 + 1 \colon \overline{N}_{\max}$
    $\boldsymbol{\mu}^{N\,\mathrm{out},*} = \arg\ \max_{\boldsymbol{\mu}\in\Xi_{\mathrm{train}}}((\omega_{N-1}(\boldsymbol{\mu}))^{-1}\,\Delta^{\mathrm{en}}_{X^{\mathrm{out},*}_{N-1}}(\boldsymbol{\mu}));$
    $\varepsilon^{\mathrm{out},*}_{N-1} = (\omega_{N-1}(\boldsymbol{\mu}))^{-1}\,\Delta^{\mathrm{en}}_{X^{\mathrm{out},*}_{N-1}}(\boldsymbol{\mu}^{N\,\mathrm{out},*});$
    if $\varepsilon^{\mathrm{out},*}_{N-1} \le \varepsilon_{\mathrm{tol,min}}$
        $N_{\max} = N - 1;$
        exit;
    $S^{\mathrm{out},*}_N = S^{\mathrm{out},*}_{N-1} \cup \boldsymbol{\mu}^{N\,\mathrm{out},*};$
    $X^{\mathrm{out},*}_N = X^{\mathrm{out},*}_{N-1} + \mathrm{span}\{u(\boldsymbol{\mu}^{N\,\mathrm{out},*})\};$
end.
```

We also define

$$\overline{\varepsilon}^{\mathrm{out},*}_N = (\omega_{N-1}(\boldsymbol{\mu}))^{-1} \max_{\boldsymbol{\mu}\in\Xi_{\mathrm{train}}} |||u(\boldsymbol{\mu}) - u_{X_N}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}, \qquad 1 \le N \le N_{\max}\ , \tag{3.72}$$

for purposes of theoretical comparisons.

From the perspective of Offline computational effort, this “output-oriented” optimization differs little from the X norm-based optimization of Section 3.4.3: the operation counts and storage for the two algorithms are identical. However, from the perspective of deliverables, we can expect improved results. In particular, it follows from Proposition 3A, (3.24), that we now directly control the RB error — since in general the RB approximation (here $u_{X^{\mathrm{out},*}_N}(\boldsymbol{\mu})$) *is* the projection of $u(\boldsymbol{\mu})$ in the energy inner product. In fact, it further follows from Proposition 3A, (3.26), that we will now even directly control the error in the RB *output* prediction: if we choose $\omega_N(\boldsymbol{\mu}) = 1$, we control the absolute error in the output; if we choose $\omega_N(\boldsymbol{\mu}) \equiv s_{X^{\mathrm{out},*}_{N-1}}(\boldsymbol{\mu})$, we control the *relative* error in the output — recall that $s_N(\boldsymbol{\mu})$ is, in general, a lower bound for $s(\boldsymbol{\mu})$. In short, we should expect that the resulting “output”-optimized spaces, $W^{\mathrm{out},*}_N$, should provide even more rapid convergence that the X norm-optimized spaces, W^*_N, since our optimization objective is more closely related to the quantity of interest. (Furthermore, as we shall see in Chapter 4, the bound for the error in the energy norm is sharper than the bound for the error in the X norm — $\eta^{\mathrm{en}}_{\mathrm{max,UB}} \leq \eta_{\mathrm{max,UB}}$ — and hence we might expect a more accurate optimization process.)

3.5 Approximation Theory

We now have at least some limited confidence that, if a (very) good low-dimensional approximation space exists, then a (at least) good Lagrange RB approximation space can be found — and constructed with reasonable computational cost by the greedy algorithm. However, we still do not have any useful, verifiable conditions that indicate *when* such a low-dimensional approximation space does indeed exist. A complete *a priori* framework must include not only the optimality results of Proposition 3A but also an approximation theory that provides (upper bounds for) "best fit" convergence rates in terms of the given data for the problem. Such a framework can also provide insights into possible best sample distributions or parameter transformations.

It was recognized in the early theoretical work on Taylor RB approximations [6] that *smoothness* in parameter — smoothness of the parametric manifold $\mathcal{M}$ — is the essential ingredient. (There are some trivial cases — we shall discuss one example below — in which dim(span$\{\mathcal{M}\}$) is small and independent of $\mathcal{N}_{\mathrm{t}}$, and the RB approximation will converge very rapidly for purely "algebraic" reasons; however these results are not of general interest.) In particular, and in contrast to convergence requirements for FE discretizations, *smoothness in the spatial coordinate is not the crucial element*. The early theoretical work on RB Taylor approximations [118, 102] demonstrates exponential convergence $u_N \rightarrow u$ in some small region about the parameter point of expansion for sufficiently smooth parametric (coefficient) dependence. Early results for the more difficult case of RB Lagrange approximations [6] are less complete and — because of norm-equivalence arguments not valid in the infinite–dimensional case — implicitly limited to finite-dimensional (non-PDE) systems; nevertheless, the fundamental link between smoothness in parameter and convergence of the RB approximation was clearly established.

We would like to develop an *a priori* approximation theory for the Lagrange case in which the requisite smoothness can be proven, the norms are consistent with the infinite-dimensional framework (in anticipation that $\mathcal{N}_{\mathrm{t}} \to \infty$), the exponents can be sharply bounded, the constants are independent of N (and perhaps also $\mathcal{N}_{\mathrm{t}}$), and the results are uniform over the entire (finite) parameter domain $\mathcal{D}$. In what follows we present some first very limited steps in this direction for a particular class of very simple model problems. However, we first discuss more generally the smoothness of $\mathcal{M}$.

3.5.1 Smoothness of $\boldsymbol{\mathcal{M}}$: Parametric (or Sensitivity) Derivatives

In this section we proceed formally; however, it is possible to develop the results more rigorously by explicitly considering the appropriate limiting process. In either the former or the latter the essential hypotheses — coercivity, continuity, and smoothness of the parameter functions — are the same. In this section our emphasis is simply on demonstrating the smoothness in parameter. However, the parametric (or sensitivity) derivatives — already introduced in general multi-index form in Section 1.4.2 — can also serve many other functions and indeed are important in their own right: as basis functions within the Taylor [102, 112, 118] and Hermite [66] RB frameworks; in objective gradients for design, optimization, and parameter estimation studies; and in fact even in *a posteriori* error estimators for the non-affine case (Part V). (We may thus pursue an *a priori* theory *for* the convergence of the sensitivity derivatives and not just an *a priori* theory *by* consideration of sensitivity derivatives. We return to the former in a later chapter: we shall see that even in the Lagrange RB context, the sensitivity derivatives will converge quite rapidly.)

To begin, we define $(\partial u/\partial \mu_i)$: $\mathcal{D} \to \mathbb{R}$, $1 \leq i \leq P$, as the derivative of $u(\boldsymbol{x};\boldsymbol{\mu})$ with respect to the i^{th} component of the parameter $\boldsymbol{\mu}$; we will write either $(\partial u/\partial \mu_i)(\boldsymbol{\mu})$ or, if we wish to emphasize the dependence on the spatial coordinate, $(\partial u/\partial \mu_i)(\boldsymbol{x};\boldsymbol{\mu})$. We shall assume that our functions Θ_a^q, $1 \le q \le Q_a$, and Θ_f^q, $1 \le q \le Q_f$, are all $C^1(\mathcal{D})$ (continuously differentiable over $\mathcal{D}$). It is then standard to (formally) derive the equation for $(\partial u/\partial \mu_i)(\boldsymbol{\mu})$ by differentiation of (2.2) [3]: given $\boldsymbol{\mu} \in \mathcal{D}$ (and $u(\boldsymbol{\mu}) \in X$), $(\partial u/\partial \mu_i)(\boldsymbol{\mu}) \in X$ satisfies

$$a\left(\frac{\partial u}{\partial \mu}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = -\sum_{q=1}^{Q_a} \left(\frac{\partial \Theta_a^q}{\partial \mu_i}\right)(\boldsymbol{\mu}) a^q(u(\boldsymbol{\mu}), v) + \sum_{q=1}^{Q_f} \left(\frac{\partial \Theta_f^q}{\partial \mu_i}\right)(\boldsymbol{\mu}) f^q(v), \ \forall v \in X, \ 1 \le i \le P\,, \tag{3.73}$$

where we have invoked our affine parameter dependence, (2.5) and (2.6). It directly follows from our coercivity and continuity assumptions on a and f (and differentiability assumptions on the Θ_a^q, $1 \le q \le Q_a$, and Θ_f^q, $1 \le q \le Q_f$) that (3.73) admits a unique and stable solution: it is a simple matter to bound $\|(\partial u/\partial \mu_i)\|_X$, $1 \le i \le P$. (In fact, parametric coercivity provides additional properties both for the sensitivity derivatives and the output variation: we will explore these further in the context of inverse problems.)

We can now readily derive from (3.73), assuming that our parameter functions $\Theta_a^q(\boldsymbol{\mu})$, $1 \le q \le Q_a$, and $\Theta_f^q(\boldsymbol{\mu})$, $1 \le q \le Q_f$, are $C^2(\mathcal{D})$, the equation for $(\partial^2 u/\partial \mu_i\, \partial \mu_j)$, $1 \le i, j \le P$: this equation will be very similar to (3.73), except with a greater proliferation of terms on the right-hand side. It is clear that if our parameter functions $\Theta_a^q(\boldsymbol{\mu})$, $1 \le q \le Q_a$, and $\Theta_f^q(\boldsymbol{\mu})$, $1 \le q \le Q_f$, are in fact $C^\infty(\mathcal{D})$ — very often the case in practice — then we can continue this differentiation process indefinitely, and the sensitivity derivatives of all order are well defined and bounded in X: $u \in C^\infty(\mathcal{D}; X)$ (see Section 1.4.2). As we shall discuss further below, even if the parametric derivatives remain bounded for any finite order, the magnitude of the parametric derivatives (in the X norm) will typically increase relatively rapidly with increasing order — with factorial and exponential terms; the rate at which the derivatives grow will of course be important in any theoretical analysis, as we shall observe in Section 3.5.2. (We work here with the "truth" approximation; however, similar results apply to the originating exact/infinite-dimensional statement.)

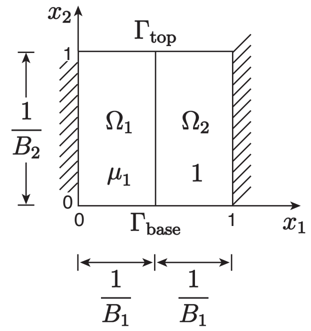

Figure 3.1: ThermalBlock for the particular case $B_1 = 2$, $B_2 = 1$.

3.5.2 *A Priori* Theory for ThermalBlock ($B_1 = 2, B_2 = 1$)

The analysis in this section is a variation — in some cases a specialization, in other cases a generalization — of the development in [90, 91]. However, the example is important in that it illustrates many of the key issues in RB approximation, and also motivates general concepts (such as logarithmic mappings) that will serve throughout our development. Hence we provide most of the details, even those that appear in somewhat similar form elsewhere.

Preliminaries

We shall consider here the $P = 1$ ThermalBlock composite corresponding to $B_1 = 2$, $B_2 = 1$ (hence $P = 1$) as shown in Figure 3.1; see Section 2.2.1 for the complete detailed definition and interpretation of this problem. (Note that for $B_1 = 1$, $B_2 = 2$, the solution is linear in x_2 and independent of x_1 in each block: $\dim(\mathcal{M}) = 2$, and the RB will reproduce the FE truth and exact solution for any $N \geq 2$.) We recall that for $B_1 = 2$, $B_2 = 1$, our subdomains are given by $\Omega^1 =]0, \frac{1}{2}[\times]0, 1[$ (with conductivity μ_1) and $\Omega^2 =]\frac{1}{2}, 1[\times]0, 1[$ (with conductivity 1) such that $\overline{\Omega} = [0, 1]^2 = \overline{\Omega}_1 \cup \overline{\Omega}_2$. Our function space X is a FE truth approximation subspace of $X^{\text{e}} = \{v \in H^1(\Omega)\, |v|_{\Gamma^D} = 0\}$, where $\Gamma^D = \Gamma_{\text{top}}$ of Figure 3.1.

We further recall that (for $P = 1$) $\boldsymbol{\mu} = \mu_1 \in \mathcal{D} \equiv [\mu_1^{\min}, \mu_1^{\max}] \subset \mathbb{R}_+$ for $1/\mu_1^{\min} = \mu_1^{\max}/1 = \sqrt{\mu_{\text{r}}}$; the extent of the parameter domain is thus given by $\mu_1^{\max}/\mu_1^{\min} = \mu_{\text{r}}$. It shall also prove convenient to introduce a mapped parameter $\hat{\mu} \in \widehat{\mathcal{D}}$, where $\hat{\mu} = \tau(\mu_1)$ for

$$\tau(z) \equiv \ln(z), \qquad \forall\, z \in \mathcal{D} \ , \tag{3.74}$$

and hence $\widehat{\mathcal{D}} = [\ln \mu_1^{\min}, \ln \mu_1^{\max}]$. This logarithmic mapping is important not just for this simple model problem but also more generally for many coercive problems with (multi-) parameter domains of significant extent.

Our problem statement is then: given $\mu_1 \in \mathcal{D}$, find $u(\mu_1) \in X$ that satisfies

$$\mu_1 a^1(u(\mu_1), v) + a^2(u(\mu_1), v) = f(v), \qquad \forall\, v \in X \ , \tag{3.75}$$

and evaluate

$$s(\mu_1) = f(u(\mu_1)) \ . \tag{3.76}$$

Here

$$a^q(w, v) = \int_{\Omega^q} \nabla w \cdot \nabla v, \quad \forall\, w, v \in X, \quad 1 \leq q \leq 2 \ , \tag{3.77}$$

and

$$f(v) = \int_{\Gamma_{\text{base}}} v, \qquad \forall\, v \in X \ ; \tag{3.78}$$

recall that in Ex1 our linear form is parameter-independent.

We choose for our inner product $a(\,\cdot\,,\,\cdot\,; \mu_{1\,\text{ref}} = 1)$, and hence

$$(w, v)_X = a^1(w, v) + a^2(w, v), \qquad \forall\, w, v \in X \ , \tag{3.79}$$

or

$$(w, v)_X \equiv \int_{\Omega} \nabla w \cdot \nabla v, \qquad \forall\, w, v \in X \ . \tag{3.80}$$

We observe that $\|\cdot\|_X = |\cdot|_{H^1(\Omega)}$; the H^1-seminorm is in fact a *norm* over X thanks to the non-zero Direchlet segment $\Gamma^D = \Gamma_{\text{top}}$.

We next introduce an equivalent parameter-mapped problem: given $\hat{\mu} \in \widehat{\mathcal{D}}$, find $\hat{u}(\hat{\mu}) \in X$ such that

$$e^{\hat{\mu}}\, a^1(\hat{u}(\hat{\mu}), v) + a^2(\hat{u}(\hat{\mu}), v) = f(v), \qquad \forall\, v \in X \ , \tag{3.81}$$

and evaluate

$$\hat{s}(\hat{\mu}) = f(\hat{u}(\hat{\mu})) \ . \tag{3.82}$$

It will also prove convenient to write (3.81) as

$$(\hat{u}(\hat{\mu}), v)_X + (e^{\hat{\mu}} - 1)\, a^1(\hat{u}(\hat{\mu}), v) = f(v), \qquad \forall\, v \in X \ . \tag{3.83}$$

Clearly, $u(\mu_1) = \hat{u}(\tau(\mu_1))$ and $\hat{u}(\hat{\mu}) = u(\tau^{-1}(\hat{\mu}))$.

We next introduce a useful symmetric positive semidefinite generalized eigenproblem: find $(\Upsilon_i^{\mathcal{N}_{\rm t}}, \lambda_i^{\mathcal{N}_{\rm t}}) \in (X, \mathbb{R}_{+0})$, $1 \le i \le \mathcal{N}_{\rm t}$, such that

$$a^1(\Upsilon_i^{\mathcal{N}_{\rm t}}, v) = \lambda_i^{\mathcal{N}_{\rm t}} (\Upsilon_i^{\mathcal{N}_{\rm t}}, v)_X, \ \forall\, v \in X, \ \text{and} \ \|\Upsilon_i^{\mathcal{N}_{\rm t}}\|_X = 1 \ . \tag{3.84}$$

We order our eigenvalues as $0 \le \lambda_i^{\mathcal{N}_{\rm t}} \le \ldots \le \lambda_{\mathcal{N}_{\rm t}}^{\mathcal{N}_{\rm t}} \le 1$; we define $\Lambda \equiv [0,1]$ as the range of the eigenvalues. (It is a simple matter to demonstrate that $\lambda_{\mathcal{N}_{\rm t}}^{\mathcal{N}_{\rm t}} = \max_{w \in X}(\int_{\Omega_1} |\nabla w|^2 / \int_{\Omega_1 \cup \Omega_2} |\nabla w|^2)$ ≤ 1 and that furthermore $\lambda_{\mathcal{N}_{\rm t}}^{\mathcal{N}_{\rm t}} \to 1$ as $\mathcal{N}_{\rm t} \to \infty$.) From the usual arguments we conclude that

$$(\Upsilon_i^{\mathcal{N}_{\rm t}}, \Upsilon_j^{\mathcal{N}_{\rm t}})_X = \delta_{i\,j}, \qquad 1 \le i, j \le \mathcal{N}_{\rm t}, \tag{3.85}$$

and

$$a^1(\Upsilon_i^{\mathcal{N}_{\rm t}}, \Upsilon_j^{\mathcal{N}_{\rm t}}) = \lambda_i^{\mathcal{N}_{\rm t}} \delta_{i\,j}, \qquad 1 \le i, j \le \mathcal{N}_{\rm t} \ , \tag{3.86}$$

and that furthermore

$$X = \text{span}\{\Upsilon_i, \ 1 \le i \le \mathcal{N}_{\rm t}\} \ ; \tag{3.87}$$

the Υ_i constitute an orthonormal basis for X.

It is then simple to derive from (3.83) and (3.85), (3.86) that

$$\hat{u}(\hat{\mu}) \ (\equiv \hat{u}^{\mathcal{N}_{\rm t}}(\hat{\mu})) = \sum_{i=1}^{\mathcal{N}_{\rm t}} f_i^{\mathcal{N}_{\rm t}} \ \Upsilon_i^{\mathcal{N}_{\rm t}} \ g(\hat{\mu}, \lambda_i^{\mathcal{N}_{\rm t}}) \ , \tag{3.88}$$

where $f_i^{\mathcal{N}_{\rm t}} = f(\Upsilon_i^{\mathcal{N}_{\rm t}})$, $1 \le i \le \mathcal{N}_{\rm t}$, and g: $\widehat{\mathcal{D}} \times \Lambda \to \mathbb{R}_+$ is given by

$$g(z, \sigma) = \frac{1}{1 - \sigma + \sigma e^z} \ . \tag{3.89}$$

We also observe that

$$|||\hat{u}(\hat{\mu})|||_{\tau^{-1}(\hat{\mu})} \equiv \left(\sum_{i=1}^{\mathcal{N}_{\rm t}} (f_i^{\mathcal{N}_{\rm t}})^2 \, g(\hat{\mu}, \lambda_i) \right)^{1/2} , \tag{3.90}$$

since

$$(\Upsilon_i^{\mathcal{N}_{\rm t}}, \Upsilon_j^{\mathcal{N}_{\rm t}})_X + (e^{\hat{\mu}} - 1)\, a^1(\Upsilon_i^{\mathcal{N}_{\rm t}}, \Upsilon_j^{\mathcal{N}_{\rm t}}) = \frac{\delta_{i\,j}}{g(\hat{\mu}, \lambda_i)}, \qquad 1 \leq i, j \leq \mathcal{N}_{\rm t} ,$$

from our orthogonality relations.

Before proceeding to the main result we require the preparatory

Lemma 3C. *For $j \in \mathbb{N}_0$,*

$$\sup_{z \in \widehat{\mathcal{D}}} \sup_{\sigma \in \Lambda} \frac{1}{g(z, \sigma)} \; |(\partial_j g)(z, \sigma)| \leq 2^j j! , \tag{3.91}$$

where $(\partial_j g)(z, \sigma)$ refers to the $j^{\rm th}$-derivative of g with respect to the first argument.

Proof. It is readily shown [90] that

$$(\partial_j g)(z, \sigma) = \sum_{k=1}^{j+1} \beta_k^j (1 - \sigma)^{k-1} \, g^k(z, \sigma) \tag{3.92}$$

and hence

$$\frac{1}{g(z, \sigma)} \; |(\partial_j g)(z, \sigma)| = \left| \sum_{k=1}^{j+1} \beta_k^j (1 - \sigma)^{k-1} \, g^{k-1}(z, \sigma) \right| . \tag{3.93}$$

The coefficients β_k^j satisfy the recurrence

$$\beta_k^{j+1} \;=\; -k\, \beta_k^j + k\, \beta_{k-1}^j, \qquad 1 \leq k \leq j+1 , \tag{3.94}$$

$$\beta_{j+2}^{j+1} \;=\; (j+1) \beta_{j+1}^j , \tag{3.95}$$

with initial condition $\beta_1^0 = 1$. (We also specify $\beta_0^j = 0$ for all j.)

It immediately follows from (3.94), (3.95) that

$$\mathcal{S}^j = \sum_{k=1}^{j+1} |\beta_k^j| \tag{3.96}$$

satisfies

$$\mathcal{S}^{j+1} \leq 2(j+1)\, \mathcal{S}^j , \tag{3.97}$$

and since $\mathcal{S}^0 = 1$,

$$\mathcal{S}^j \le 2^j\, j!\ . \tag{3.98}$$

Thus from (3.98) and (3.89)

$$\begin{aligned}\sup_{z\in\widehat{\mathcal{D}}}\ \sup_{\sigma\in\Lambda}\left|\sum_{k=1}^{j+1}\beta_k^j\,(1-\sigma)^{k-1}\,g^{k-1}(z,\sigma)\right| & \\ &\le\ 2^j\,j!\sup_{k=1,\ldots,j+1}\ \sup_{z\in\widehat{\mathcal{D}}}\ \sup_{\sigma\in\Lambda}\frac{(1-\sigma)^{k-1}}{((1-\sigma)+\sigma e^z)^{k-1}}\\ &\le\ 2^j\,j!\ . \end{aligned} \tag{3.99}$$

The desired result directly follows from (3.93) and (3.99). ■

The growth in the derivatives of g can be related to the growth in the sensitivity derivatives of u.

Convergence Analysis

We next introduce the samples $S_N^{\mathrm{nh}} = G^{\ln}_{[\mu_1^{\min},\mu_1^{\max};N]}$ (see Section 1.4.2),

$$S_N^{\mathrm{nh}} = \left\{(\mu_1)_N^n \equiv e^{\{\ln\mu_1^{\min}+(n-1)\delta_N\}},\ 1\le n\le N\right\},\qquad 2\le N\le N_{\max}\ , \tag{3.100}$$

where

$$\delta_N = \frac{\ln\mu_{\mathrm{r}}}{N-1}\ ; \tag{3.101}$$

note that

$$\hat{\mu}_N^n = \tau((\mu_1)_N^n) = \ln\mu_1^{\min} + \frac{(n-1)}{N-1}\left(\ln\mu_1^{\max}-\ln\mu_1^{\min}\right),\quad 1\le n\le N,\ 2\le N\le N_{\max}\ , \tag{3.102}$$

and hence our sample points are *equidistributed* in $\hat{\mu}$ — $\hat{\mu}_N^n - \hat{\mu}_N^{n-1} = \delta_N$, $2\le n\le N$. (The sample points bear a subscript N as the samples are not nested; however, where no (additional) confusion shall be created, we shall suppress the superscript.) We then define the associated RB Lagrange spaces

$$X_N^{\mathrm{nh}} \equiv W_N^{\mathrm{nh}} \equiv \mathrm{span}\left\{u(\boldsymbol{\mu}_N^n),\ 1\le n\le N\right\},\qquad 2\le N\le N_{\max}\ . \tag{3.103}$$

Although the W_N^{nh} are not hierarchical, obviously $W_2^{\mathrm{nh}} \subset W_3^{\mathrm{nh}} \subset W_5^{\mathrm{nh}} \subset W_9^{\mathrm{nh}} \subset \cdots$; in any event, we do not propose these spaces as practical approximations. The corresponding RB field and output approximations, (3.28), (3.30), shall be denoted $u_{W_N^{\mathrm{nh}}}$ and $s_{W_N^{\mathrm{nh}}}$, respectively.

Given our relation (3.88), we note that

$$u((\mu_1)_N^n) = \hat{u}\left(\tau\left((\mu_1)_N^n\right)\right) = \sum_{i=1}^{\mathcal{N}_{\mathrm{t}}} f_i^{\mathcal{N}_{\mathrm{t}}}\, \Upsilon_i^{\mathcal{N}_{\mathrm{t}}}\, g\left(\hat{\mu}_N^n, \lambda_i^{\mathcal{N}_{\mathrm{t}}}\right) \ . \tag{3.104}$$

We next introduce, for any given $\widehat{C}_n \colon \widehat{\mathcal{D}} \to \mathbb{R}$, $1 \le n \le N$, the function $\tilde{g}_N \colon \widehat{\mathcal{D}} \times \Lambda \to \mathbb{R}$ given by

$$\tilde{g}_N(\hat{\mu}, \sigma) = \sum_{n=1}^{N} \widehat{C}_n(\hat{\mu})\, g(\hat{\mu}_N^n, \sigma) \ . \tag{3.105}$$

It then follows from (3.88) and (3.105) that the function $\hat{w}_N \colon \widehat{\mathcal{D}} \to \mathbb{R}$ given by

$$\hat{w}_N(\hat{\mu}) = \sum_{i=1}^{\mathcal{N}_{\mathrm{t}}} f_i^{\mathcal{N}_{\mathrm{t}}}\, \Upsilon_i^{\mathcal{N}_{\mathrm{t}}}\, \tilde{g}_N(\hat{\mu}, \lambda_i^{\mathcal{N}_{\mathrm{t}}}) \tag{3.106}$$

is a member of our RB space W_N^{nh}.

We shall subsequently choose polynomial interpolants for out "best-fit" functions $\tilde{g}_N$; we thus introduce the basic results here. Given a function $h \in C^0(\mathcal{D})$, and positive integers i and M such that $i + M \le N + 1$, we defined $\mathcal{I}_{N,M}^i h$ as the unique $(M-1)^{\mathrm{th}}$-order polynomial satisfying $(\mathcal{I}_{N,M}^i h)(\hat{\mu}_N^{i+m-1}) = h(\hat{\mu}_N^{i+m-1})$, $1 \le m \le M$.

We further introduce the Lagrange basis functions $L_{N,M}^{i;m}$, $1 \le m \le M$: $L_{N,M}^{i;m}$ is the unique $(M-1)^{\mathrm{th}}$-order polynomial satisfying $L_{N,M}^{i;m}(\hat{\mu}_N^{i+m'-1}) = \delta_{m\,m'}$, $1 \le m, m' \le M$. We can then express our interpolant as

$$(\mathcal{I}_{N,M}^i h)(\hat{\mu}) = \sum_{m=1}^{M} L_{N,M}^{i;m}(\hat{\mu})\, h(\hat{\mu}_N^{i+m-1}) \tag{3.107}$$

for any $\hat{\mu} \in \widehat{\mathcal{D}}$. Note that $\mathcal{I}_{N,M}^i$ is, of course, polynomial; however $\mathcal{I}_{N,M}^i \circ \tau$ is *not* polynomial.

We now recall the basic remainder results of Lagrange interpolation (particularized to our case of interest) [42, 124]: for $h \in C^M(\widehat{\mathcal{D}})$,

$$|h(\hat{\mu}) - (\mathcal{I}_{N,M}^i h)(\hat{\mu})| \le \frac{((M-1)\delta_N)^M}{M!} \left(\sup_{y \in \widehat{\mathcal{D}}} h^{(M)}(y) \right) \ , \tag{3.108}$$

for any $\hat{\mu} \in [\hat{\mu}_N^i, \hat{\mu}_N^{i+M-1}]$; here $h^{(m)}$ is the m^{th}-derivative of h. (Note that, from (3.102), $(M-1)\delta_N$ is the *length* of the interval $[\hat{\mu}_N^i, \hat{\mu}_N^{i+M-1}]$.)

We now turn to the main result in

Proposition 3D. *Assume*

$$\ln \mu_{\mathrm{r}} = \ln\left(\frac{\mu_1^{\max}}{\mu_1^{\min}}\right) > \frac{1}{2e} \tag{3.109}$$

(in fact, just a technical convenience) and

$$N \geq N_{\text{crit}} \equiv 1 + [2e \ln \mu_{\mathrm{r}}]_+ \ , \tag{3.110}$$

where for arg $\in \mathbb{R}$ *the function* $[\text{arg}]_+$ *(respectively,* $[\text{arg}]_-$*) shall denote the smallest integer* $\geq$ arg *(respectively, the largest integer* $\leq$ arg*). Then*

$$\frac{|||u^{\mathcal{N}_{\mathrm{t}}}(\mu_1) - u_{W_N^{\mathrm{nh}}}(\mu_1)|||_{\boldsymbol{\mu}}}{|||u^{\mathcal{N}_{\mathrm{t}}}(\mu_1)|||_{\boldsymbol{\mu}}} \leq e^{-\frac{(N-1)}{(N_{\text{crit}}-1)}}, \qquad \forall\, \mu_1 \in \mathcal{D} \ , \tag{3.111}$$

and

$$\frac{s^{\mathcal{N}_{\mathrm{t}}}(\mu_1) - s_{W_N^{\mathrm{nh}}}(\mu_1)}{s^{\mathcal{N}_{\mathrm{t}}}(\mu_1)} \leq e^{-\frac{2(N-1)}{(N_{\text{crit}}-1)}}, \qquad \forall\, \mu_1 \in \mathcal{D} \ , \tag{3.112}$$

for W_N^{nh} *defined in (3.103).*

Proof. We first note that for any $\hat{w}_N(\hat{\mu}) \in W_N^{\mathrm{nh}}$ of the form (3.106) we can write

$$\begin{aligned} \frac{|||\hat{u}(\hat{\mu}) - \hat{w}_N(\hat{\mu})|||_{\tau^{-1}(\hat{\mu})}}{|||\hat{u}(\hat{\mu})|||_{\tau^{-1}\hat{\mu}}} \quad &= \quad \left(\frac{\sum_{i=1}^{\mathcal{N}_{\mathrm{t}}} \left(f_i^{\mathcal{N}_{\mathrm{t}}}\right)^2 \left(g\left(\hat{\mu}, \lambda_i^{\mathcal{N}_{\mathrm{t}}}\right) - \tilde{g}_N\left(\hat{\mu}, \lambda_i^{\mathcal{N}_{\mathrm{t}}}\right)\right)^2 \Big/ g\left(\hat{\mu}, \lambda_i^{\mathcal{N}_{\mathrm{t}}}\right)}{\sum_{i=1}^{\mathcal{N}_{\mathrm{t}}} \left(f_i^{\mathcal{N}_{\mathrm{t}}}\right)^2 \Big/ g\left(\hat{\mu}, \lambda_i^{\mathcal{N}_{\mathrm{t}}}\right)} \right)^{1/2} \\ &\leq \quad \sup_{z \in \widehat{\mathcal{D}}} \sup_{\sigma \in \Lambda} \frac{1}{g(z,\sigma)} \left| g(z,\sigma) - \tilde{g}_N(z,\sigma) \right| \ , \qquad \forall\, \hat{\mu} \in \widehat{\mathcal{D}} \ . \end{aligned} \tag{3.113}$$

We now choose $\tilde{g}_N(\hat{\mu}, \sigma)$ judiciously.

In particular, given any $\hat{\mu} \in \widehat{\mathcal{D}}$, we identify M $(2 \leq M \leq N)$ contiguous sample points $\hat{\mu}_N^{i^*(\hat{\mu})}, \ldots, \hat{\mu}_N^{i^*(\hat{\mu})+M-1}$ and an associated "enclosing" interval $\mathcal{J}_M^{\hat{\mu}} \equiv [\hat{\mu}_N^{i^*(\hat{\mu})}, \hat{\mu}_N^{i^*(\hat{\mu})+M-1}]$ of length $(M-1)\delta_N$ (in $\hat{\mu}$) such that $\hat{\mu} \in \mathcal{J}_M^{\hat{\mu}}$. There are obviously many possible choices for $i^*(\hat{\mu})$: for our "crude" purposes here, any choice suffices; we take $i^*(\hat{\mu})$ to be the smallest i such that, for the given M, $\hat{\mu} \in [\hat{\mu}_N^i, \hat{\mu}_N^{i+M-1}]$. We then take $\tilde{g}_N(\hat{\mu}, \sigma) \equiv \tilde{g}^*_{N,M}(\hat{\mu}, \sigma) \equiv \big(\mathcal{I}_{N,M}^{i^*(\hat{\mu})}\, g(\,\cdot\,, \sigma)\big)(\hat{\mu})$. We must and can confirm that, as necessary, $\tilde{g}^*_{N,M}(\hat{\mu}, \sigma)$ has the form (3.105): we directly identify from (3.107) that (i) $\widehat{C}_n = 0$ for $n < i^*(\hat{\mu})$ and $n > i^*(\hat{\mu})$, and (ii) $\widehat{C}_n = L_{N,M}^{i^*(\hat{\mu}); n - i^*(\hat{\mu})+1}(\hat{\mu})$ for $i^*(\hat{\mu}) \le n \le i^*(\hat{\mu}) + M - 1$.

It directly follows from the standard Lagrange interpolant remainder formula (3.108) and Lemma 3C that, if we define

$$B^*(N, M) \equiv \sup_{z \in \widehat{\mathcal{D}}} \sup_{\sigma \in \Lambda} \frac{1}{g(z, \sigma)} \left| g(z, \sigma) - \tilde{g}^*_{N,M}(z, \sigma) \right| \tag{3.114}$$

then

$$B^*(N, M) \le (2(M-1)\delta_N)^M \ . \tag{3.115}$$

It remains to select the optimal — or at least a good suboptimal — M that minimizes $(2(M-1)\delta_N)^M$ for $M \in \{2, \ldots, N\}$. For simplicity, we shall first relax this problem and look for $\overline{M} \in [2, N] \subset \mathbb{R}$ such that $(2(\overline{M}-1)\delta_N)^{\overline{M}}$ is suitably small.

It is readily demonstrated that the function $(2r)^{r/\delta_N}$ — motivated by the identification "$r \approx (M-1)\delta_N$" — attains the global minimum of $e^{-\left(\frac{1}{2e\delta_N}\right)} = e^{-\left(\frac{N-1}{2e \ln \mu_r}\right)} \le e^{-\left(\frac{N-1}{N_{\rm crit}-1}\right)}$ over all $r \in [0, \infty[$ for $r_{\rm opt} = 1/2e$. We thus choose

$$\overline{M}_{\rm opt} = 1 + \frac{r_{\rm opt}}{\delta_N} \ ; \tag{3.116}$$

note that $\overline{M}_{\rm opt} \ge 2$ (respectively, $\overline{M}_{\rm opt} \le N$) thanks to our condition on N, (3.110) (respectively, our condition on $\mu_{\rm r}$, (3.109)). Thus, since $2(\overline{M}_{\rm opt}-1)\delta_N = 2r_{\rm opt}$,

$$(2(\overline{M}_{\rm opt}-1)\delta_N)^{\overline{M}_{\rm opt}-1} = (2r_{\rm opt})^{\frac{r_{\rm opt}}{\delta_N}} \le e^{-\left(\frac{N-1}{N_{\rm crit}-1}\right)} \ . \tag{3.117}$$

It remains only to address the integer nature of M.

In particular, we shall now choose $M_{\rm opt} = [\overline{M}_{\rm opt}]_-$, the largest integer $\le \overline{M}_{\rm opt}$. Clearly, $2 \le M_{\rm opt} \le N$, as required. Furthermore, (i) since $M_{\rm opt} \le \overline{M}_{\rm opt}$, $2(M_{\rm opt}-1)\delta_N < 2(\overline{M}_{\rm opt}-1)\delta_N$ $(= \frac{1}{e} < 1)$, and (ii) from the definition of $[\ \ ]_-$, $M_{\rm opt} > \overline{M}_{\rm opt} - 1$. Thus,

$$(2(M_{\rm opt}-1)\delta_N)^{M_{\rm opt}} \le (2(\overline{M}_{\rm opt}-1)\delta_N)^{\overline{M}_{\rm opt}-1} \le e^{-\left(\frac{N-1}{N_{\rm crit}-1}\right)} \ . \tag{3.118}$$

We thus conclude from (3.113), (3.114), (3.115), and (3.118) that, under the hypotheses of the Proposition 3D, for any $\mu_1 \in \mathcal{D}$, there exists a function $w_N(\mu_1)$ $(= \hat{w}_N(\tau(\mu_1))$ such that

$$\frac{|||u^{\mathcal{N}_\mathrm{t}}(\mu_1) - w_N(\mu_1)|||_{\boldsymbol{\mu}}}{|||u^{\mathcal{N}_\mathrm{t}}(\mu_1)|||_{\boldsymbol{\mu}}} \leq e^{-\left(\frac{N-1}{N_\mathrm{crit}-1}\right)} \ . \tag{3.119}$$

The energy result (3.111) then directly follows from (3.119) and Proposition 3A, (3.24); the output result (3.112) directly follows from $s^{\mathcal{N}_\mathrm{t}}(\mu_1) = |||u^{\mathcal{N}_\mathrm{t}}(\mu_1)|||^2_{\boldsymbol{\mu}}$, (3.119), and Proposition 3A, (3.26). ■

As anticipated, given the smooth nature of the parametric dependence and hence the underlying parametric manifold $\mathcal{M}$, we achieve *exponential convergence*. (Note we do not exploit any smoothness in space.) We also observe that our constants related to the convergence rate — in particular N_crit — are *independent* of $\mathcal{N}_\mathrm{t}$.

We can also demonstrate

Corollary 3E. *For any $N \in \mathbb{N}$, $N \geq 2$,*

$$\frac{s^{\mathcal{N}_\mathrm{t}}(\mu_1) - s_{W_N^\mathrm{nh}}(\mu_1)}{s^{\mathcal{N}_\mathrm{t}}(\mu_1)} \leq (B^*(N,N))^2, \qquad \forall\, \mu_1 \in \mathcal{D} \ , \tag{3.120}$$

for the RB space W_N^nh given in (3.103) and $B^*(N, M = N)$ defined in (3.114). (Note for $M = N$, $\tilde{g}^*_{N,M}(\,\cdot\,,\sigma) \equiv (\mathcal{I}^1_{N,N}\, g(\,\cdot\,,\sigma)$ is simply the $(N-1)^{\mathrm{th}}$-order Lagrangian interpolant of $g(\,\cdot\,,\sigma)$ through the $\hat{\mu}^i_N$, $1 \leq i \leq N$.)

Proof. The result directly follows from (3.113) and (3.114) of Proposition 3D, $s^{\mathcal{N}_\mathrm{t}}(\mu_1) = |||u^{\mathcal{N}_\mathrm{t}}(\mu_1)|||^2_{\boldsymbol{\mu}}$, and (3.26) of Proposition 3A. ■

This Corollary is of interest for two reasons. First, we expect the bound (3.120) to be considerably sharper than (3.112), and hence to better demonstrate the *reason* for rapid RB convergence — smoothness in parameter. (Of course, (3.120) is really only a pseudo-*a priori* result; we will simply evaluate $B^*(N,N)$ numerically.) Second, the result is valid for all N (≥ 2), and in particular for the small N of practical relevance. Indeed, the bound should not be good for larger N.

We make several comments motivated by the proof of Proposition 3D. First, we note the rather small — in fact, logarithmic — effect of the extent of the parameter domain, μ_{r}, on the exponential convergence rate. Although the proof does suggest that the best approximation might include (with *significant weight*) only some set of snapshots in the vicinity of any particular $\mu_1 \in \mathcal{D}$, the active region (related to r_{opt} in the proof) is $O(1)$ in $\ln \mu_1$; furthermore, the optimal Galerkin approximation will no doubt appeal to all N snapshots. In short, the RB approximation is global and high order, and can efficiently treat global parameter domains.

Second, the logarithmic transformation suggested by the proof — and in fact responsible for the weak dependence of the convergence rate on μ_{r} — is more generally relevant: a good preprocessing transformation prior to generation of (say, train) samples. The logarithmic distribution can be motivated intuitively: in our problem, μ_1 and $1/\mu_1$ enter in a symmetric fashion — we can consider $a^1(u(\mu_1), v) + (1/\mu_1)a^2(u(\mu_1), v)$ rather than $\mu_1 a^1(u(\mu_1), v) + a^2(u(\mu_1), v)$. As we shall see, the $\Theta^q(\boldsymbol{\mu})$ are often of the form $\mu_{\boldsymbol{\cdot}}$ or $1/\mu_{\boldsymbol{\cdot}}$ (even in the presence of geometric parameters), and hence the log argument is rather broadly applicable. (Note our results of Proposition 3D are not sensitive to small perturbations in the logarithmic samples [90, 91].) Our observations here are generally true not only for parametrically coercive problems but also for more general coercive problems.

Given the smoothness in parameter and our construction of Proposition 3D, we might ask why we can not simply directly interpolate our field variable u (and even our output) in μ_1. We consider this question in greater detail in the next section in the context of higher parameter dimensions. But we emphasize already here, for $P = 1$, the power of the Galerkin recipe. In particular, our candidate best fit in the proof of Proposition 3D and Corollary 3E is, as already noted, *not* polynomial in μ_1 but rather polynomial in $\ln(\mu_1)$ — not necessarily an obvious choice *a priori*. Notwithstanding, it follows from Proposition 3A, (3.24), that the Galerkin procedure provably will do *better* than this candidate "best fit" and in fact perforce choose the best (in the energy norm) linear combination of snapshots. We anticipate that (at

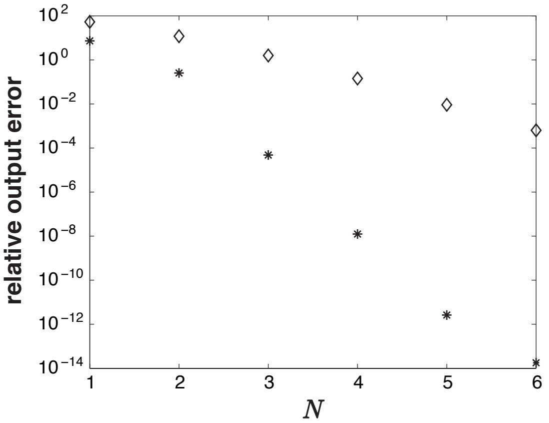

Figure 3.2: The relative error in the output over Ξ^0_{train} for $s_{W_N^{\text{nh}}}$: the actual error ($*$) and the pseudo-*a priori* bound of Corollary 3E ($\Diamond$).

least in the energy norm) the "constructed" *sub-optimality* of any proposed interpolant will, in general, be trumped by the "automatic" Galerkin *optimality* of Proposition 3A.

Numerical Results

We now present some numerical results for our model problem to better understand the implications of Proposition 3D and Corollary 3E and the sampling discussions of Section 3.4. We consider here the case $\mu_r = 100$ corresponding to $\mu_1^{\min} = 0.1$ and $\mu_1^{\max} = 10.0$ — a very extensive variation in the parameter. Most of our results will be based on a train sample $\Xi^0_{\text{train}} = G^{\ln}_{[\mu_1^{\min},\mu_1^{\max};10{,}000]}$ (see Section 1.4.2).

We plot in Figure 3.2 (i) the maximum over Ξ^0_{train} of $|s(\mu_1) - s_{W_N^{\text{nh}}}(\mu_1)|/s(\mu_1)$, the relative error in the output for the (non-hierarchical) RB approximation space W_N^{nh} of (3.103) associated with the "optimal" log sample S_N^{nh} of (3.100), and (ii) the pseudo-*a priori* bound for $|s(\mu_1) - s_{W_N^{\text{nh}}}(\mu_1)|/s(\mu_1)$ of Corollary 3E, corresponding to (numerical computation of) (3.114) for the particular case $M = N$. Note for (i) we consider a $\mathbb{P}_1$ finite element approximation of dimension $\mathcal{N}_{\text{t}} = 1024$.

We observe that the RB output approximation converges exponentially and rapidly. We also note that — as must be the case from Proposition 3A, (3.26) — the Galerkin approximation is indeed better than the interpolant as measured in the energy norm: the actual output error is considerably smaller than $(B^*(N,N))^2$. Nevertheless, $(B^*(N,N))^2$ constitutes a reasonable bound, and we thus conclude that the basic premise that informs the construction of Proposition 3D and Corollary 3E is valid: the RB method provides a framework for *high order* approximation in a *smooth* parametric variable.

However, the *a priori* bound for $|s(\mu_1) - s_{W_N^{\rm nh}}(\mu_1)|/s(\mu_1)$ of Proposition 3D, (3.112), is unfortunately not "practically" relevant. In particular, $N_{\rm crit} = 26$ (for $\mu_1^{\rm max}/\mu_1^{\rm min} = \mu_r = 100$) is much too pessimistic: the threshold resolution $N_{\rm crit}$ of (3.110) is thus much too stringent — "off the plot" in Figure 3.2; and the convergence exponent $-2/(N_{\rm crit} - 1)$ of (3.112) is much too small — an order of magnitude too conservative. The culprit is clearly our overly crude bound for $B^*(N, M_{\rm opt})$.

We next compare in Figure 3.3(a) the error in the output as a function of N for two different RB approximations: (i) the maximum over $\Xi^0_{\rm train}$ of $|s(\mu_1) - s_{W_N^{\rm nh}}(\mu_1)|/s(\mu_1)$, and (ii) $(\bar{\varepsilon}_N^{\rm out,*})^2$ of (3.72) for Greedy$^{\rm out}$($N_0 = 1$, $\mu_1^{1\,\rm out,*} = 1$, $\Xi^0_{\rm train}$, $\varepsilon_{\rm tol,min}$, $\omega_N(\mu_1) = \sqrt{s_N(\mu_1)}$), (which from (3.26) is a bound for) the maximum over $\Xi^0_{\rm train}$ of $|s(\mu_1) - s_{W_N^{\rm out,*}}(\mu_1)|/s(\mu_1)$. (We obtain very similar results if we calculate the maximum errors over an independent test sample $\Xi_{\rm test} \in \mathcal{D}$ rather than over $\Xi^0_{\rm train}$ — since $n_{\rm train} \gg N$.)

We observe that the hierarchical greedy RB approximation — the true (practical) RB approximation — behaves roughly as well as the non-hierarchical approximation associated with the presumably optimal logarithmic samples: the greedy selection procedure correctly identifies nested samples for which the Galerkin procedure can provide very rapid convergence. We conclude that, fortunately, we do not need to exploit special information or parameter tranformations to achieve (in practice) rapid RB convergence.

However, the ln sample is not just an artifact of the proof of Proposition 3D. We present in

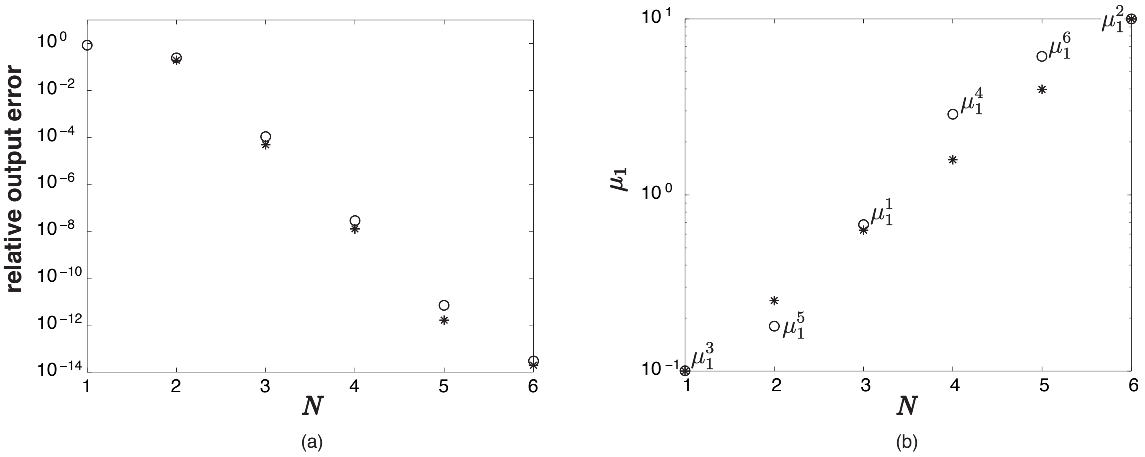

Figure 3.3: (a) The relative error in the output over Ξ^0_{train} as a function of N for $s_{W_N^{\text{nh}}}$ ($*$) and $s_{W_N^{\text{out},*}}$ (○). (b) The logarithmic sample S_N^{nh} ($*$) and the greedy sample $S_N^{\text{out},*}$ (○) for $N = 6$; the greedy points are re-ordered to facilitate comparison.

Figure 3.3(b) the logarithmic sample S_N^{nh} and the greedy sample $S_N^{\text{out},*}$ for $N = 6$. Clearly, the two samples are somewhat similar. There is a way in which we can exploit (and have exploited) the ln distribution in a non-binding fashion: through importance sampling as reflected in Ξ^0_{train}. In general, the greedy result will be insensitive to Ξ_{train} if Ξ_{train} is sufficiently "dense" in $\mathcal{D}$; however, the ln choice (rather than lin) for Ξ_{train} can be important in higher parameter dimensions (see Section 3.5.3) in *reducing* n_{train} — and hence Offline expense — at constant RB "quality."

The truth approximation in Figure 3.3(a) is a $\mathbb{P}_1$ finite element approximation of dimension $\mathcal{N}_{\text{t}} = 1024$ over a uniform triangulation. (Note for this problem the dimension of $\mathcal{M}$ is in fact the number of degrees of freedom on the interface $\overline{\Omega}^1 \cap \overline{\Omega}^2$, or roughly $\sqrt{\mathcal{N}_{\text{t}}}$.) We already know that our bound of Proposition 3D is independent of $\mathcal{N}_{\text{t}}$, and hence we expect (and observe) very little change in $|s^{\mathcal{N}_{\text{t}}}(\mu_1) - s_{W_N^{\text{nh}}}(\mu_1)|/s^{\mathcal{N}_{\text{t}}}(\mu_1)$ as we vary $\mathcal{N}_{\text{t}}$. Furthermore, the spaces generated by our greedy process are also quite insensitive to $\mathcal{N}_{\text{t}}$: we present in Figure 3.4 $(\bar{\varepsilon}_N^{\text{out},*})^2$ (for Greedy$^{\text{out}}(1, \mu_1^{1\,\text{out},*} = 1,\, \Xi^0_{\text{train}},\, \varepsilon_{\text{tol,min}},\, \omega_N(\mu_1) = \sqrt{s_N(\mu_1)}))$ as a function of N but now for the different truth approximations $\mathcal{N}_{\text{t}} = 64$, $\mathcal{N}_{\text{t}} = 256$, $\mathcal{N}_{\text{t}} = 1024$, and $\mathcal{N}_{\text{t}} = 4096$. We observe that the greedy convergence curve is little effected by $\mathcal{N}_{\text{t}}$ for $\mathcal{N}_{\text{t}}$ sufficiently large: the RB

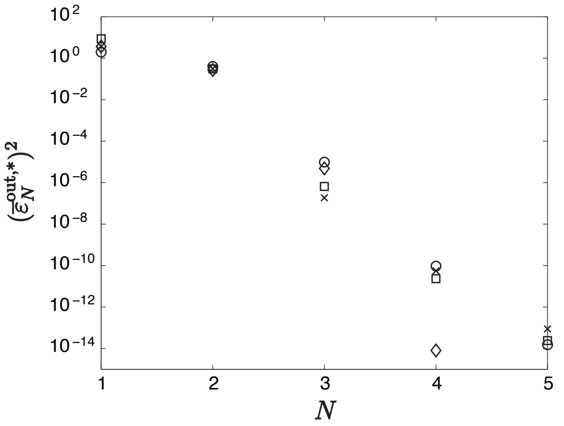

Figure 3.4: Ouput error measure $(\bar{\varepsilon}_N^{\text{out},*})^2$ for $\text{Greedy}^{\text{out}}(1, \mu_1^{1\,\text{out},*} = 1, \Xi^0_{\text{train}}, \varepsilon_{\text{tol,min}}, \omega_N(\mu_1) = \sqrt{s_N(\mu_1)})$ as a function of N for different $\mathcal{N}_\text{t}$: $\mathcal{N}_\text{t} = 64$ ($\Diamond$); $\mathcal{N}_\text{t} = 256$ (○); $\mathcal{N}_\text{t} = 1024$ ($\Box$); $\mathcal{N}_\text{t} = 4096$ ($\times$).

approach provides a stable approximation to $u^{\mathcal{N}_\text{t}}$ as $\mathcal{N}_\text{t} \to \infty$. (Note that for $\mathcal{N}_\text{t}$ too small we can, for larger N, exhaust span$\mathcal{M}$ — somewhat apparent in Figure 3.4 for $\mathcal{N}_\text{t} = 64$; we shall further explore this phenomenon, for a richer multi-parameter problem, in Section 3.5.3.)

Finally, we take the opportunity for this rather simple ($P = 1$) model problem to compare the greedy and POD approaches. (For larger P, the computational cost of the POD approach is typically prohibitive.) In order to be able to provide a more meaningful comparison, we now consider the non-output greedy algorithm of Section 3.4.3 such that both the greedy optimization and the POD are defined with respect to an X-norm objective. We now consider a smaller training sample $\Xi^1_{\text{train}} = G^{\ln}_{[\mu_1^{\min}, \mu_1^{\max}; 500]}$. For our truth approximation, we take $\mathcal{N}_\text{t} = 1024$.

We can envision computing eight quantities

$$\begin{aligned}
&(i) && \max_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \|u(\boldsymbol{\mu}) - u_{W^*_N}(\boldsymbol{\mu})\|_X \ , \\
&(ii) && \sqrt{\frac{1}{n_{\rm train}} \sum_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \|u(\boldsymbol{\mu}) - u_{W^*_N}(\boldsymbol{\mu})\|^2_X} \ , \\
&(iii) && \max_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \inf_{w_N\in W^*_N} \|u(\boldsymbol{\mu}) - w_N\|_X \ , \\
&(iv) && \sqrt{\frac{1}{n_{\rm train}} \sum_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \inf_{w_N\in W^*_N} \|u(\boldsymbol{\mu}) - w_N\|^2_X} \ , \\
&(v) && \max_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \|u(\boldsymbol{\mu}) - u_{X^{\rm POD}_N}(\boldsymbol{\mu})\|_X \ , \\
&(vi) && \sqrt{\frac{1}{n_{\rm train}} \sum_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \|u(\boldsymbol{\mu}) - u_{X^{\rm POD}_N}(\boldsymbol{\mu})\|^2_X} \ , \\
&(vii) && \max_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \inf_{w_N\in X^{\rm POD}_N} \|u(\boldsymbol{\mu}) - w_N\|_X \ , \\
&(viii) && \sqrt{\frac{1}{n_{\rm train}} \sum_{\boldsymbol{\mu}\in\Xi^1_{\rm train}} \inf_{w_N\in X^{\rm POD}_N} \|u(\boldsymbol{\mu}) - w_N\|^2_X} \ .
\end{aligned} \tag{3.121}$$

We make several observations.

First, we note that $(i) \geq (ii)$, $(iii) \geq (iv)$, $(v) \geq (vi)$, and $(vii) \geq (viii)$: the $L^\infty(\Xi^1_{\rm train})$ is stronger than the $L^2(\Xi^1_{\rm train})$-norm. Second, we note that $(i) \geq (iii)$, $(ii) \geq (iv)$, $(v) \geq (vii)$, and $(vi) \geq (viii)$: the error in the RB Galerkin approximation of $u(\boldsymbol{\mu})$ measured in the X-norm will always be greater than the error in the X-projection of $u(\boldsymbol{\mu})$ measured in the X-norm. Third, we note that $(iv) \geq (viii)$: from (3.52), the POD is optimal in the $L^2(\Xi^1_{\rm train})$-"projection" metric. We have verified that all of our numerical results do in fact honor these sets of relations.

As already introduced, the "native" quantities associated with the greedy and POD sampling strategies are (3.121) $(i) = \overline{\varepsilon}^*_N$ of (3.66) and (3.121) $(viii) = \overline{\overline{\varepsilon}}^{\rm POD}_N$ of (3.51), respectively. These quantities are not computationally directly comparable: $\overline{\overline{\varepsilon}}^{\rm POD}_N$ is defined relative to

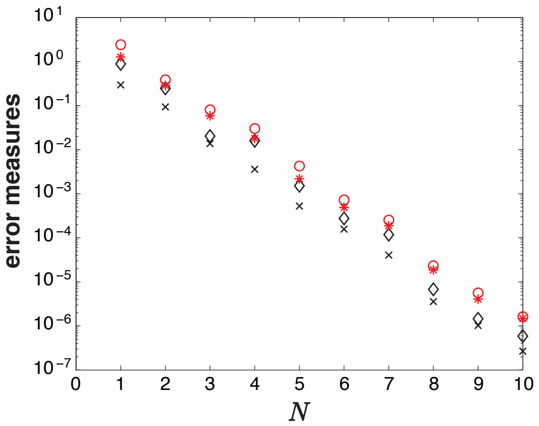

Figure 3.5: Error measures for the greedy and POD RB approximations. We present results for the $L^\infty(\Xi^1_{\text{train}})$-norm: (3.121) (i) greedy ($*$) and (3.121) (v) POD (○). We also present results for the $L^2(\Xi^1_{\text{train}})$-norm: (3.121) (ii) greedy ($\Diamond$) and (3.121) (vi) POD ($\times$).

a weaker norm over Ξ^1_{train} and with respect to the projection — and is therefore "favored." Hence we instead compare, in Figure 3.5, (i) with (v) and (ii) with (vi). As might be expected, the greedy and POD each perform slightly better in the norms over Ξ^1_{train} which inform the respective objective function/optimization problems. The advantage of the greedy approach is perhaps the stronger norm but, much more importantly, the computational efficiency: we can readily consider very large n_{train} with relatively little increase in Offline cost; this will be particularly advantageous in higher parameter dimensions.

3.5.3 Higher Parameter Dimensions

For $P = 1$ the theory is thus in some sense rather complete: clearly, we can extend the proof of Proposition 3D to more general a^1 (and, in fact, to more general $(\cdot\,,\cdot)_X$) [90, 91]. However, for $P > 1$, the state of affairs is much less satisfactory: not only can we not construct a meaningful or practically relevant *a priori* theory, but in fact we can not even readily understand the rapid convergence observed numerically. Part of the issue is the more general problem of interpolation in higher dimensional spaces; and part of the issue is the particular problems associated with analysis of partial differential equations.

In particular, it might not be overly difficult to extend the theory for $P = 1$ to tensor-product samples and spaces for $P > 1$ — for example, in $P = 2$, $S_{N=N_1N_2} = G^{\mathrm{ln}}_{[\mu_1^{\mathrm{min}},\mu_1^{\mathrm{max}};N_1]} \times G^{\mathrm{ln}}_{[\mu_2^{\mathrm{min}},\mu_2^{\mathrm{max}};N_2]}$. (Unfortunately) these spaces are clearly prohibitively/exponentially expensive both Offline *and* Online for larger P — the convergence rate with N degrades very rapidly with P. (Fortunately) the greedy sampling process generates *non*-tensor-product samples that provide very rapid convergence — the convergence rate with N appears to depend only weakly on P. However, the latter is based on observation and not theory: we can not quantify the conditions under which rapid convergence must obtain; rapid convergence remains largely a mystery. (Perhaps one might envision an analysis similar to Proposition 3D defined over curves in $\mathcal{D}$ — essentially an arc-length representation of our parametrization. However, the construction of sufficiently smooth curves is not at all self-evident.)

We do provide here some of the empirical evidence which suggests that, at least for some problems, RB approximation is viable for modest P — several parameters — and in some cases even "large" P — $O(10)$. We consider the ThermalBlock problem of Section 2.2.1 with $B_1 = B_2 = 3$ and hence $P = 8$; the particular situation is depicted in Figure 2.1. As before, we consider the parameter range $\mu_r = 100$. We choose for our truth solution a $\mathbb{P}_1$ finite element approximation of dimension $\mathcal{N}_{\mathrm{t}}$ over a uniform triangulation. For our training sample we shall consider both $\Xi^{\mathrm{ln}}_{\mathrm{train}} = G^{\mathrm{ln}}_{[\mathrm{MC};n_{\mathrm{train}}]}$ and $\Xi^{\mathrm{lin}}_{\mathrm{train}} = G^{\mathrm{lin}}_{[\mathrm{MC};n_{\mathrm{train}}]}$.

We plot in Figure 3.6(a) $(\varepsilon_N^{\mathrm{out},*})^2$ for $\mathrm{Greedy}^{\mathrm{out}}(1, \boldsymbol{\mu}^{1\,\mathrm{out},*} = (1,\ldots,1), \Xi^{\mathrm{ln}}_{\mathrm{train}}, \varepsilon_{\mathrm{tol,min}}, \omega_N(\boldsymbol{\mu}) = \sqrt{s_N(\boldsymbol{\mu})})$ as a function of N for $n_{\mathrm{train}} = 500$, 1000, 5000, 10000, and in Figure 3.6(b) $(\varepsilon_N^{\mathrm{out},*})^2$ for $\mathrm{Greedy}^{\mathrm{out}}(1, \boldsymbol{\mu}^{1\,\mathrm{out},*} = (1,\ldots,1), \Xi^{\mathrm{lin}}_{\mathrm{train}}, \varepsilon_{\mathrm{tol,min}}, \omega_N(\boldsymbol{\mu}) = \sqrt{s_N(\boldsymbol{\mu})})$ as a function of N for $n_{\mathrm{train}} = 500$, 1000, 5000, 10000. (In both cases, $\mathcal{N}_{\mathrm{t}} = 661$.) We observe that in both the ln and lin cases the results are sensitive to n_{train} for smaller n_{train}; however, the ln sample approaches a roughly n_{train}-independent asymptote — a better and more reliable RB approximation — more quickly than the lin sample. (Note that we now calculate and present $\varepsilon_N^{\mathrm{out},*}$ rather than $\bar{\varepsilon}_N^{\mathrm{out},*}$ — as in actual practice.)

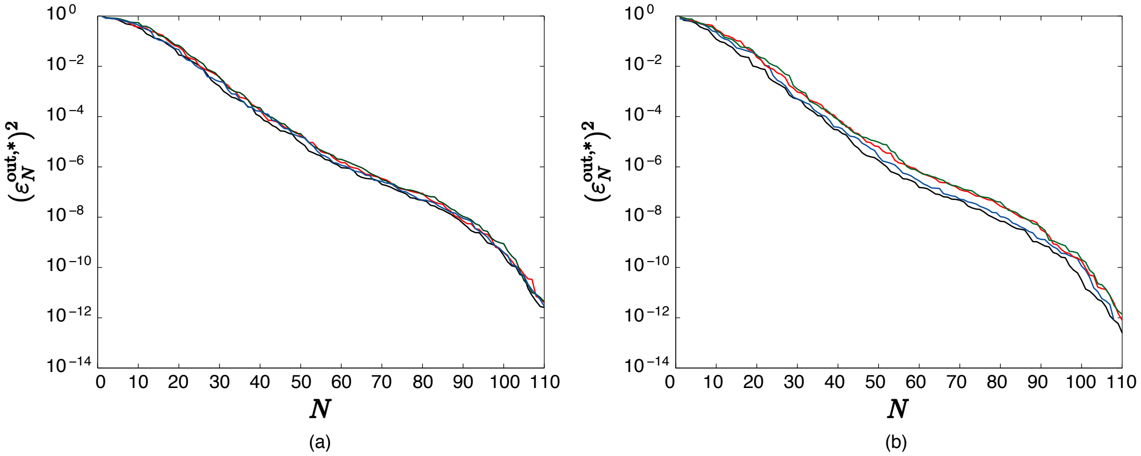

Figure 3.6: Output measure $(\varepsilon_N^{\text{out},*})^2$ for Greedy$^{\text{out}}(1, \boldsymbol{\mu}^{1\,\text{out},*} = (1,\ldots,1), \Xi_{\text{train}}, \varepsilon_{\text{tol,min}}, \omega_N(\boldsymbol{\mu}) = \sqrt{s_N(\boldsymbol{\mu})})$ as a function of N for (a) $\Xi_{\text{train}} = G^{\text{ln}}_{[\text{MC};n_{\text{train}}]}$, and (b) $\Xi_{\text{train}} = G^{\text{lin}}_{[\text{MC};n_{\text{train}}]}$, for different n_{train}: $n_{\text{train}} = 500$ (—); $n_{\text{train}} = 1000$ (—); $n_{\text{train}} = 5000$ (—); $n_{\text{train}} = 10000$ (—).

The good news is that even for P larger the greedy sampling procedure can provide what appears to be rapidly convergent approximations: we achieve a relative output accuracy of 0.01% over a fine sample in $\mathcal{D}$ with only $N \cong 40$. The bad news is that in $P = 8$ dimensions even $n_{\text{train}} = 10{,}000$ is in fact very small — "on average" about 3 points per parameter dimension — and hence we can not be certain that there are not (many!) points in $\mathcal{D}$ for which the error remains quite large. The latter highlights the necessity of our *a posteriori* error estimators of the next section, which will permit us to at least verify Online that any particular prediction is accurate . . . or not. (It is not true either in the greedy or the POD contexts that the error over Ξ_{train} is the error over $\mathcal{D}$ — though this is often incorrectly presumed.)

We present in Figure 3.7 the ln training sample results of Figure 3.6(a) for the particular case of $n_{\text{train}} = 500$ but now for $\mathcal{N}_{\text{t}} = 178$, $\mathcal{N}_{\text{t}} = 453$, $\mathcal{N}_{\text{t}} = 661$, $\mathcal{N}_{\text{t}} = 1737$, $\mathcal{N}_{\text{t}} = 2545$, and $\mathcal{N}_{\text{t}} = 6808$. We again recall that the dimension of $\mathcal{M}$ is in fact not $\mathcal{N}_{\text{t}}$ but rather the number of degrees of freedom on the block interfaces; we can detect this effect for the smaller $\mathcal{N}_{\text{t}}$ and larger N — we "exhaust" $\mathcal{M}$. We observe very little effect of (*sufficiently large*) $\mathcal{N}_{\text{t}}$ on the RB convergence results; we confirm that the RB approach provides a stable approximation to $u^{\mathcal{N}_{\text{t}}}$ as $\mathcal{N}_{\text{t}} \to \infty$.

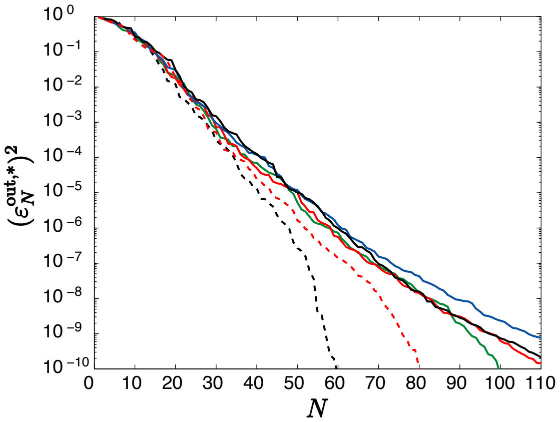

Figure 3.7: Output measure $(\varepsilon_N^{\text{out},*})^2$ for $\text{Greedy}^{\text{out}}(1, \boldsymbol{\mu}^{1\,\text{out},*} = (1, \ldots, 1), \Xi_{\text{train}}^{\ln}, \varepsilon_{\text{tol,min}}, \omega_N(\boldsymbol{\mu}) = \sqrt{s_N(\boldsymbol{\mu})})$ for different $\mathcal{N}_{\text{t}}$: $\mathcal{N}_{\text{t}} = 137$ (- - -); $\mathcal{N}_{\text{t}} = 453$ (- - -); $\mathcal{N}_{\text{t}} = 661$ (—); $\mathcal{N}_{\text{t}} = 1737$ (—); $\mathcal{N}_{\text{t}} = 2545$ (—); $\mathcal{N}_{\text{t}} = 6808$ (—).

It is clear from a comparison of the ThermalBlock results of Figure 3.6(a) and Figure 3.3(a) that the dimension (at fixed accuracy) of the RB approximation space for $B_1 B_2 = 9$ is certainly larger than the dimension of the RB approximation space for $B_1 B_2 = 2$. (Note however, that in Figure 3.3(a) we report the actual error whereas in Figure 3.6(a) we report the (much less expensive) error bound; hence Figure 3.6(a) is a bit pessimistic.) Nevertheless, the growth in requisite RB dimension is in fact quite modest given the much more significant growth in the number of parameters. The even more demanding case of $B_1 = B_2 = 5$ is considered in [137]: again, the effect of increased P is noticeable but rather modest.

In general, parametrically coercive problems and in particular *compliant* parametrically coercive problems exhibit many special properties. For example, for the ThermalBlock problem — for which furthermore the Θ_a^q, $1 \leq q \leq Q_a$, are *linear* in $\boldsymbol{\mu}$ — the output is monotonically decreasing in each parameter. However, the *field variable* associated with the ThermalBlock problem can in fact exhibit rather rich behavior [137], and hence the rapid RB convergence we observe would not appear to be trivial.

We emphasize that it is the *field variable* which must be (smooth and) well approximated by the RB approach; the scalar output is then "derivative." It is passage through the field variable which — through Galerkin orthogonality, error estimation, and the greedy procedure — arguably endows the RB approach with the very rapid and verifiable convergence properties for larger P. We can certainly also entertain a much simpler “connect-the-dots” approach: direct approximation of $s\colon \mathcal{D} \to \mathbb{R}$. However, we contend that connect-the-dots — unlike the RB approach — is simply not viable in higher parameter dimensions: tensor product grids are patently infeasible and scattered data approximations [64, 153] are typically rather poorly convergent. We further contend that connect-the-dots — unlike the RB approach — does not admit any rigorous and efficient *a posteriori* error estimation procedure: in particular for smaller N, the output does not carry sufficient residual or stability information. Thus as regards both efficiency and reliability the RB approach is preferred. (Another justification is simply that we like the RB approach better — and presumably so does any reader that has persevered this far in the book.)

In fact, even for $P = 1$, connect-the-dots is not obviously better than RB. We consider (i) $(N-1)^{\text{th}}$-order polynomial interpolation of $s(\mu_1)$ on the points $G^{\text{Cheb,lin}}_{[\mu_1^{\min},\mu_1^{\max};N]}$ (see Section 1.4.2), (ii) $(N-1)^{\text{th}}$-order polynomial interpolation of $s(\mu_1)$ on the points $G^{\text{Cheb,ln}}_{[\mu_1^{\min},\mu_1^{\max};N]}$, and (iii) $(N-1)^{\text{th}}$-order interpolation of $\hat{s}(\hat{\mu}) = s(e^{\hat{\mu}})$ on the points $G^{\text{Cheb,lin}}_{[\ln\mu_1^{\min},\ln\mu_1^{\max};N]}$. Note that (ii) and (iii) are not equivalent, even though in μ_1 the (abscissa, ordinate) interpolation pairs are identical: (ii) is polynomial in μ_1, while (iii) is polynomial in $\ln\mu_1$ (in fact, the output associated with our best-fit surrogate of Corollary 3E).

In Figure 3.8 we plot the maximum over a sample $\Xi_{\text{test}} = G^{\ln}_{[\text{MC},1000]}$ of the relative output error (the output error normalized by $s(\mu_1)$) for these three interpolants as well as for the greedy RB approximation reported in Figure 3.3(a). In fact, the RB approximation is roughly as accurate as the (two better of the three) interpolants. (Note that Proposition 3A does not in any way preclude an output interpolant that is occasionally or even always better than the RB prediction: (3.26) should not be interpreted as optimality of the output.) Since the cost of the polynomial interpolation is only N^2 (compared to N^3 for the RB), connect-the-dots is arguably

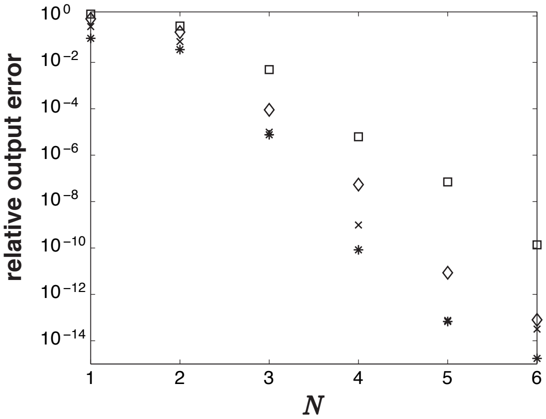

Figure 3.8: Relative output error over Ξ_{test} for the "connect-the-dots" interpolants (*i*) (□), (*ii*) (×), (*iii*) (∗), as well as the greedy RB approximation (◊) of Figure 3.3a.

more efficient for this simple $P = 1$ problem. However, this conclusion is specious: first, for $P > 1$ the situation is much different; and second, even for $P = 1$, without a rigorous and inexpensive *a posteriori* error estimator (in the Online stage), we can not choose N rationally and in particular minimally to ensure the desired accuracy. We now turn to the subject of RB *a posteriori* error estimation.

Chapter 4

A Posteriori Error Estimation

4.1 Motivation and Requirements

Effective *a posteriori* error bounds for the quantity of interest — our output — are crucial both for the efficiency and the reliability of RB approximations. As regards *efficiency* (related to the concept of "adaptivity" within the FE context), error bounds play a role in both the Offline and Online stages. In the greedy algorithms, the application of error bounds (as surrogates for the actual error) permits significantly larger training samples at greatly reduced Offline computational cost. These more extensive training samples in turn engender RB approximations which provide higher accuracy at greatly reduced Online computational cost. The error bounds also serve directly in the Online stage — to find the smallest RB dimension N that achieves the requisite accuracy — to further optimize Online performance. In short, *a posteriori* error estimation permits us to (inexpensively) control the error which in turn permits us to minimize the computational effort.

As regards *reliability*, it is clear that our Offline sampling procedures can not be exhaustive: for larger parameter dimensions P there will be large "parts" of the parameter set $\mathcal{D}$ that remain unexplored — the output error uncharacterized; we must admit that we will only encounter most parameter values in $\mathcal{D}$ Online. (In the POD context, $\overline{\overline{\varepsilon}}_N^{\mathrm{POD}}$ is often reported as the "error" over $\mathcal{D}$): $\overline{\overline{\varepsilon}}_N^{\mathrm{POD}}$ is *not* the error over $\mathcal{D}$, but rather the error over Ξ_{train}. Similarly, in the greedy context, $\overline{\varepsilon}_N^*$ is *not* the error (bound) over $\mathcal{D}$, but rather the error (bound) over Ξ_{train}.) Our *a posteriori* estimation procedures ensure that we can rigorously and efficiently bound the output error in the Online (deployed/application) stage — for *any* given "new" $\boldsymbol{\mu} \in \mathcal{D}$. We can thus be sure that constraints are satisfied, feasibility (and safety/failure) conditions are verified, and prognoses are valid: *real-time or design decisions are endowed with the full assurances of the "truth" solution*. In short, *a posteriori* error bounds permit us to confidently — with certainty — exploit the rapid predictive power of the RB approximation.

We should emphasize that *a posteriori* output error bounds are particularly important for RB approximations. First, RB approximations are *ad hoc*: each problem is different as regards discretization. Second, RB approximations are typically pre-asymptotic: we will choose N quite small — before any "tail" in the convergence rate. And third, the RB basis functions can not be directly related to any spatial or temporal scales: physical intuition is of little value. And fourth and finally, the RB approach is typically applied in the real-time context: there is no time for Offline verification; errors are immediately manifested and often in deleterious ways. There is, thus, even greater need for *a posteriori* error estimation in the RB context than in the much more studied FE context [4, 5, 12, 11, 13, 22].

Our motivations for error estimation in turn place requirements on our error bounds. First, the error bounds must be *rigorous* — valid for all N and for all parameter values in the parameter domain $\mathcal{D}$: non-rigorous error "indicators" may suffice for adaptivity, but not for reliability. Second, the bounds must be reasonably *sharp*: an overly conservative error bound can yield ineffcient approximations (N too large) or suboptimal engineering results (unnecessary safety margins); design should be dictated by the output and not the output error. And third, the bounds must be very *efficient*: the Online operation count and storage to compute the RB error bounds — the marginal or asymptotic cost — must be independent of $\mathcal{N}_{\mathrm{t}}$ (and hopefully commensurate with the cost associated with the RB output prediction). We do re-emphasize here that our RB error bounds are defined relative to the underlying "truth" FE approximation; however, we also recall that the RB Online cost is independent of $\mathcal{N}_{\mathrm{t}}$, and hence the truth approximation can and should be chosen conservatively.

4.2 Coercivity Lower Bound

4.2.1 Preliminaries

The material in this particular subsection will have broader application within the book (in particular, in Part IV on parabolic problems), and we thus consider a slightly larger family of bilinear forms. In particular, we shall now permit a *non-symmetric* parametrically coercive bilinear form b: $X \times X \times \mathcal{D} \to \mathbb{R}$. (We choose the notation b rather than a since we will sometimes require stability lower bounds for forms other than our PDE form a; however, in most cases, $b = a$.)

We shall, however, continue to assume that b is coercive and continuous,

$$0 < \left(\alpha^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu}) \equiv\right) \alpha(\boldsymbol{\mu}) = \inf_{w \in X} \frac{b_{\mathrm{S}}(w,w;\boldsymbol{\mu})}{\|w\|_X^2}, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ , \tag{4.1}$$

and

$$\sup_{w\in X} \sup_{v \in X} \frac{|b(w,v;\boldsymbol{\mu})|}{\|w\|_X \, \|v\|_X} = \gamma(\boldsymbol{\mu}) \left(\equiv \gamma^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu})\right) < \infty, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ , \tag{4.2}$$

respectively. Recall that $b_{\mathrm{S}}(w,v;\boldsymbol{\mu}) = \frac{1}{2}(b(w,v;\boldsymbol{\mu}) + b(v,w;\boldsymbol{\mu}))$ is the symmetric part of b.

Given that b need not be symmetric, we must slightly generalize our choice of X inner product and norm. In particular, given a $\overline{\boldsymbol{\mu}} \in \mathcal{D}$, we now define

$$(w,v)_X \equiv b_{\mathrm{S}}(w,v;\overline{\boldsymbol{\mu}}), \qquad \forall\, w,v \in X \ , \tag{4.3}$$

and hence

$$\|w\|_X = b_{\mathrm{S}}^{1/2}(w,w,;\overline{\boldsymbol{\mu}}), \qquad \forall\, w \in X \ ; \tag{4.4}$$

note since b is coercive, $b_{\mathrm{S}}(w,v;\overline{\boldsymbol{\mu}})$ is indeed a well-defined inner product. Clearly (4.4) reduces to our earlier definition (2.12) in the case (of interest in Part I) in which b $(= a)$ is symmetric.

As before, we continue to assume that b is *parametrically* coercive; however, since now $b \neq b_{\mathrm{S}}$, we must recall our more careful definition of parametric coercivity. We first define $c(w, v; \boldsymbol{\mu}) \equiv b_{\mathrm{S}}(w, v; \boldsymbol{\mu})$, $\forall\, w, v, \in X$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$. Since b is affine, c perforce also admits an affine (symmetric) decomposition,

$$c(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \Theta_c^q(\boldsymbol{\mu})\, c^q(w, v), \qquad \forall\, w, v \in X,\ \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \tag{4.5}$$

with $c^q(w, v) = c^q(v, w)$, $\forall\, w, v \in X$, $1 \leq q \leq Q_c$. We then say that b is *parametrically coercive* if

$$\Theta_c^q(\boldsymbol{\mu}) > 0, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D},\ 1 \leq q \leq Q_c\ , \tag{4.6}$$

and

$$c^q(w, w) \geq 0, \qquad \forall\, w \in X,\ 1 \leqq q \leq Q_c\ . \tag{4.7}$$

Note that the parametric coercivity condition is defined in terms of $c \equiv b_{\mathrm{S}}$, not b: there can be skew-symmetric components to b that need not honor our "positivity conditions" (4.6),(4.7) — a classical and important example is the (steady or unsteady) convection diffusion equation.

We observe that if b *is* symmetric, then (we may choose) $Q_b = Q_c$, $\Theta_b^q(\boldsymbol{\mu}) = \Theta_c^q$, $1 \leq q \leq Q_b$, and $b^q(w, v) = c^q(w, v)$, $\forall\, w, v \in X$, $1 \leq q \leq Q_b$. It follows that

$$\Theta_b^q(\boldsymbol{\mu}) > 0, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D},\ 1 \leq q \leq Q_b\ , \tag{4.8}$$

and

$$b^q(w, w) \geq 0, \qquad \forall\, w \in X,\ 1 \leq q \leq Q_b\ . \tag{4.9}$$

Note the $b^q(w, v)$, $1 \leq q \leq Q_b$, are symmetric positive semidefinite bilinear forms and thus the Cauchy-Schwarz inequality is applicable: $|b^q(w, v)| \leq (b^q(w, w))^{1/2}\, (b^q(v, v))^{1/2}$, $\forall\, w, v \in X$, $1 \leq q \leq Q_b$.

4.2.2 The "min Θ" Approach

We shall now develop a positive lower bound for the coercivity "constant," $0 < \alpha_{\mathrm{LB}}(\boldsymbol{\mu}) \leq \alpha^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in \mathcal{D}$. This lower bound is *required* — a computational ingredient in our *a posteriori* error bounds. (Indeed, as we shall see, the critical simplification afforded by the hypothesis of parametric coercivity is the existence of an *explicitly* constructed and calculated coercivity lower bound.) For the case of a symmetric — as in the compliant problems of Part I — we shall also develop a finite upper bound for the continuity “constant,” $\gamma^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}) \le \gamma_\mathrm{UB}(\boldsymbol{\mu}) < \infty$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$. This upper bound is more elective — typically only a theoretical clarification in our effectivity discussions.

We now introduce a (readily evaluated) function $\Theta^{\min,\overline{\boldsymbol{\mu}}}_{b_\mathrm{S}}\colon \mathcal{D} \to \mathbb{R}_+$, defined by

$$\Theta^{\min,\overline{\boldsymbol{\mu}}}_{b_\mathrm{S}}(\boldsymbol{\mu}) = \min_{q\in\{1,\ldots,Q_c\}} \frac{\Theta^q_c(\boldsymbol{\mu})}{\Theta^q_c(\overline{\boldsymbol{\mu}})} \ . \tag{4.10}$$

We can then demonstrate

Lemma 4A. *For b parametrically coercive,*

$$0 < \Theta^{\min,\overline{\boldsymbol{\mu}}}_{b_\mathrm{S}}(\boldsymbol{\mu}) \le \alpha^{\mathcal{N}_\mathrm{t}}(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \tag{4.11}$$

for $\Theta^{\min,\overline{\boldsymbol{\mu}}}_{b_\mathrm{S}}$ defined in (4.10).

Proof. We note that (for $c = b_\mathrm{S}$),

$$\begin{aligned} c(w,w;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_c} \Theta^q_c(\boldsymbol{\mu})\, c^q(w,w) \\ &= \sum_{q=1}^{Q_c} \frac{\Theta^q_c(\boldsymbol{\mu})}{\Theta^q_c(\overline{\boldsymbol{\mu}})}\, \Theta^q_c(\overline{\boldsymbol{\mu}})\, c^q(w,w) \\ &\ge \left(\min_{q\in\{1,\ldots,Q_c\}} \frac{\Theta^q_c(\boldsymbol{\mu})}{\Theta^q_c(\overline{\boldsymbol{\mu}})}\right) b_\mathrm{S}(w,w;\overline{\boldsymbol{\mu}}) \\ &= \Theta^{\min,\overline{\boldsymbol{\mu}}}_{b_\mathrm{S}}(\boldsymbol{\mu}) \|w\|^2_X, \qquad \forall\, w \in X,\ \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \end{aligned} \tag{4.12}$$

from our positivity conditions and choice of norm. Hence

$$\alpha(\boldsymbol{\mu}) \equiv \inf_{w\in X} \frac{c(w,w;\boldsymbol{\mu})}{\|w\|^2_X} \ge \Theta^{\min,\overline{\boldsymbol{\mu}}}_{b_\mathrm{S}}(\boldsymbol{\mu}) > 0, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \tag{4.13}$$

as desired. ■

We may thus choose

$$\alpha_{\mathrm{LB}}(\boldsymbol{\mu}) \equiv \Theta_{b_{\mathrm{S}}}^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \tag{4.14}$$

as our "$\min \Theta$" coercivity constant lower bound. We shall need to compute $\alpha_{\mathrm{LB}} = \Theta_{b_{\mathrm{S}}}^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})$ to evaluate the error bounds of the next section. Clearly, the Online complexity is $O(Q_b)$ — typically negligible. (Recall that we generally assume that the Θ_b^q, $1 \leq q \leq Q_b$, are simple algebraic expressions.)

We now further assume that b is symmetric. We may then define

$$\Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) = \max_{q \in \{1,\ldots,Q_b\}} \frac{\Theta_b^q(\boldsymbol{\mu})}{\Theta_b^q(\overline{\boldsymbol{\mu}})} \ , \tag{4.15}$$

and, for future reference

$$\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \equiv \frac{\Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})}{\Theta_{b_{\mathrm{S}}}^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})} \ . \tag{4.16}$$

We can then prove

Lemma 4B. *For b parametrically coercive and symmetric,*

$$\gamma^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu}) \leq \Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) < \infty, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ , \tag{4.17}$$

for $\Theta_b^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})$ defined in (4.15).

Proof. We note that

$$\begin{aligned} b(w,v;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu})\, a^q(w,v) \\ &= \sum_{q=1}^{Q_b} \frac{\Theta_b^q(\boldsymbol{\mu})}{\Theta_b^q(\overline{\boldsymbol{\mu}})}\, \Theta_b^q(\overline{\boldsymbol{\mu}})\, b^q(w,v) \\ &\leq \left(\max_{q \in \{1,\ldots,Q_b\}} \frac{\Theta_b^q(\boldsymbol{\mu})}{\Theta_b^q(\overline{\boldsymbol{\mu}})} \right) \sum_{q=1}^{Q_b} \Theta_b^q(\overline{\boldsymbol{\mu}})\, |b^q(w,v)| \\ &\leq \Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \sum_{q=1}^{Q_b} \left(\Theta_b^q(\overline{\boldsymbol{\mu}})\, b^q(w,w)\right)^{1/2} \left(\Theta_b^q(\overline{\boldsymbol{\mu}})\, b^q(v,v)\right)^{1/2} \\ &\leq \Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})\, \|w\|_X\, \|v\|_X \qquad \forall\, w,v \in X \ , \end{aligned} \tag{4.18}$$

from our positivity conditions, application (twice) of the Cauchy-Schwarz inequality, and our choice of norm. Hence

$$\gamma(\boldsymbol{\mu}) = \sup_{w \in X} \sup_{v \in X} \frac{b(w,v;\boldsymbol{\mu})}{\|w\|_X \, \|v\|_X} \leq \Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) < \infty, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ , \tag{4.19}$$

as desired. ■

We may thus choose

$$\gamma_{\mathrm{UB}}(\mu) \equiv \Theta_{b_{\mathrm{S}}}^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \ . \tag{4.20}$$

(as our "max Θ" continuity upper bound).

We emphasize that Lemma 4A and Lemma 4B are restricted to parametrically coercive and parametrically coercive symmetric forms, respectively. Furthermore, the results presented here are only directly applicable to the particular choice of inner product $(w,v)_X \equiv b_{\mathrm{S}}(w,v;\overline{\boldsymbol{\mu}})$, though this restriction is readily relaxed.

4.3 *A Posteriori* Error Estimators

Note that in this section we re-place ourselves in the compliant (and hence *a* symmetric) framework of Part I.

4.3.1 Prerequisites

The central equation in *a posteriori* theory is the error residual relationship. In particular, it follows from the problem statements for $u^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu})$, (2.51), and $u_N(\boldsymbol{\mu})$, (3.22), that the error $(e^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu}) \equiv)\ e(\boldsymbol{\mu}) \equiv u^{\mathcal{N}_{\mathrm{t}}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}) \in X\ (\equiv X^{\mathcal{N}_{\mathrm{t}}})$ satisfies

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = r_N(v;\boldsymbol{\mu}), \qquad \forall\, v \in X \ . \tag{4.21}$$

Here $r(v;\boldsymbol{\mu}) \in X'$ (the dual space to X) is the residual,

$$r(v;\boldsymbol{\mu}) \equiv f(v;\boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \qquad \forall\, v \in X \ . \tag{4.22}$$

(Indeed, (4.21) directly follows from (4.22), $f(v;\boldsymbol{\mu}) = a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu})$, $\forall\, v \in X$, bilinearity of a, and the definition of $e(\boldsymbol{\mu})$.)

It shall prove convenient to introduce the Riesz representation of $r(v;\boldsymbol{\mu})$, $\hat{e}(\boldsymbol{\mu}) \in X$: from Section 1.2.1 of Chapter 1, $\hat{e}(\boldsymbol{\mu}) \in X$ satisfies

$$(\hat{e}(\boldsymbol{\mu}), v)_X = r(v;\boldsymbol{\mu}), \qquad \forall\, v \in X \ . \tag{4.23}$$

We can thus also write the error residual equation as

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = (\hat{e}(\boldsymbol{\mu}), v)_X, \qquad \forall\, v \in X \ . \tag{4.24}$$

(We note that for our choice of inner product (2.28), $\hat{e}(\overline{\boldsymbol{\mu}}) = e(\overline{\boldsymbol{\mu}})$.)

It also follows from (1.8) (see Section 1.2.1) that

$$\|r(\,\cdot\,;\boldsymbol{\mu})\|_{X'} \equiv \sup_{v\in X} \frac{r(v;\boldsymbol{\mu})}{\|v\|_X} = \|\hat{e}(\boldsymbol{\mu})\|_X; \tag{4.25}$$

the evaluation of the dual norm of the residual through the Riesz representation is central to the Offline-Online procedures to be developed in Section 4.4 below.

It may appear that, since $u(\boldsymbol{\mu})$ and $e(\boldsymbol{\mu})$ satisfy very similar equations — the same operator with different right-hand sides — it would be just as easy to find $u(\boldsymbol{\mu})$ as $e(\boldsymbol{\mu})$. The critical (though trivial) point is that we can be much more "relaxed" in our treatment of the error: a bound for the field $u(\boldsymbol{\mu})$ or output $s(\boldsymbol{\mu})$ good to 100% is patently useless; however, a bound for the *error* $e(\boldsymbol{\mu})$ or $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$ good to 100% (or even 500%) is quite useful.

4.3.2 Energy and Output Error Bounds

We define error estimators for the energy norm, output, and "relative" output as

$$\Delta_N^{\text{en}}(\boldsymbol{\mu}) \ \equiv \ \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{\alpha_{\text{LB}}^{1/2}(\boldsymbol{\mu})}\ , \tag{4.26a}$$

$$\Delta_N^{s}(\boldsymbol{\mu}) \ \equiv \ \frac{\|\hat{e}(\boldsymbol{\mu})\|_X^2}{\alpha_{\text{LB}}(\boldsymbol{\mu})}, \qquad \big(= (\Delta_N^{\text{en}}(\boldsymbol{\mu}))^2\big)\ , \tag{4.26b}$$

and

$$\Delta_N^{s,\mathrm{rel}}(\boldsymbol{\mu}) \;\equiv\; \frac{\|\hat{e}(\boldsymbol{\mu})\|_X^2}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})\, s_N(\boldsymbol{\mu})}, \qquad \left(= \frac{\Delta_N^s(\boldsymbol{\mu})}{s_N(\boldsymbol{\mu})}\right) , \tag{4.26c}$$

respectively. Here $\|\hat{e}(\boldsymbol{\mu})\|_X$ is the dual norm of the residual, as defined in (4.25), and $\alpha_{\mathrm{LB}}(\boldsymbol{\mu}) \equiv \Theta_{a_{\mathrm{S}}}^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})$ of (4.10) (for $b = a$). (Note in Chapter 3 we denote $\Delta_N^{\mathrm{en}}(\boldsymbol{\mu})$ by the more explicit label $\Delta_{X_N}^{\mathrm{en}}(\boldsymbol{\mu})$ to emphasize (in the greedy algorithm) the particular RB space.)

We next introduce the effectivities associated with these error estimators:

$$\eta_N^{\mathrm{en}}(\boldsymbol{\mu}) \;\equiv\; \frac{\Delta_N^{\mathrm{en}}(\boldsymbol{\mu})}{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}} , \tag{4.27a}$$

$$\eta_N^{s}(\boldsymbol{\mu}) \;\equiv\; \frac{\Delta_N^{s}(\boldsymbol{\mu})}{s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})} , \tag{4.27b}$$

and

$$\eta_N^{s,\mathrm{rel}}(\boldsymbol{\mu}) \;\equiv\; \frac{\Delta_N^{s,\mathrm{rel}}(\boldsymbol{\mu})}{\left(s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})\right)/s(\boldsymbol{\mu})} . \tag{4.27c}$$

Clearly, the effectivities are a measure of the quality of the proposed estimator: for rigor, we shall insist upon effectivities ≥ 1; for sharpness, we desire effectivities as close to unity as possible.

We can then prove (recall we remain here in the parametrically coercive compliant and hence *symmetric* framework)

Proposition 4C. *For any $N = 1, \ldots, N_{\max}$, the effectivities (4.27a) and (4.27b) satisfy*

$$1 \;\leq\; \eta_N^{\mathrm{en}}(\boldsymbol{\mu}) \;\leq\; \sqrt{\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})}, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} , \tag{4.28a}$$

$$1 \leq\; \eta_N^{s}(\boldsymbol{\mu}) \;\leq\; \theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} , \tag{4.28b}$$

respectively. Furthermore, for $\Delta_N^{s,\mathrm{rel}}(\boldsymbol{\mu}) \leq 1$, the effectivity (4.27c) satisfies

$$1 \;\leq\; \eta_N^{s,\mathrm{rel}}(\boldsymbol{\mu}) \;\leq\; 2\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \; ; \tag{4.28c}$$

in fact, the left inquality in (4.28c) is valid for all $\boldsymbol{\mu} \in \mathcal{D}$ *and for* all $N = 1, \ldots, N_{\max}$.

Proof. It follows directly from (4.24) for $v = e(\boldsymbol{\mu})$ and the Cauchy-Schwarz inequality that

$$|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}^2 \leq \|\hat{e}(\boldsymbol{\mu})\|_X \, \|e(\boldsymbol{\mu})\|_X \ . \tag{4.29}$$

But from coercivity and Lemma 4A $\alpha_{\rm LB}^{1/2}(\boldsymbol{\mu}) \, \|e(\boldsymbol{\mu})\|_X \leq a^{1/2}(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) \equiv |||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$, and hence from (4.29), (4.26a), and (4.27a) $|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N^{\rm en}(\boldsymbol{\mu})$ or $\eta_N^{\rm en}(\boldsymbol{\mu}) \geq 1$. We now again consider (4.24) — but now for $v = \hat{e}(\boldsymbol{\mu})$ — and the Cauchy-Schwarz inequality to obtain

$$\|\hat{e}(\boldsymbol{\mu})\|_X^2 \leq |||\hat{e}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \, |||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \ . \tag{4.30}$$

But from continuity and Lemma 4B $|||\hat{e}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \gamma_{\rm UB}^{1/2}(\boldsymbol{\mu}) \, \|\hat{e}(\boldsymbol{\mu})\|_X$, and hence $\Delta_N^{\rm en}(\boldsymbol{\mu}) \equiv \alpha_{\rm LB}^{-1/2}(\boldsymbol{\mu}) \, \|\hat{e}(\boldsymbol{\mu})\|_X \leq \alpha_{\rm LB}^{-1/2}(\boldsymbol{\mu}) \, \gamma_{\rm UB}^{1/2}(\boldsymbol{\mu}) \, |||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$, or $\eta_N^{\rm en}(\boldsymbol{\mu}) \leq \sqrt{\gamma_{\rm UB}(\boldsymbol{\mu})/\alpha_{\rm LB}(\boldsymbol{\mu})}$. Thus, recalling (4.16) and(4.20), (4.28a) is proven.

Next we know from Proposition 3A that $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) = |||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}^2$, and hence since $\Delta_N^s(\boldsymbol{\mu}) = (\Delta_N^{\rm en}(\boldsymbol{\mu}))^2$

$$\eta_N^s(\boldsymbol{\mu}) \equiv \frac{\Delta_N^s(\boldsymbol{\mu})}{s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})} = \frac{(\Delta_N^{\rm en}(\boldsymbol{\mu}))^2}{|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}^2} = (\eta_N^{\rm en}(\boldsymbol{\mu}))^2 \ ; \tag{4.31}$$

(4.28b) directly follows from (4.28a) and (4.31).

Finally, since $\Delta_N^{s,\rm rel}(\boldsymbol{\mu}) = \Delta_N^s(\boldsymbol{\mu})/s_N(\boldsymbol{\mu})$,

$$\eta_N^{s,\rm rel}(\boldsymbol{\mu}) = (s(\boldsymbol{\mu})/s_N(\boldsymbol{\mu})) \ \eta_N^s(\boldsymbol{\mu}) \ . \tag{4.32}$$

But we know from Proposition 3A that $s_N(\boldsymbol{\mu}) \leq s(\boldsymbol{\mu})$, which with (4.28b) proves the left inequality in (4.28c) for all $\boldsymbol{\mu} \in \mathcal{D}$ and for all $N = 1, \ldots, N_{\max}$. If we now further expand $s(\boldsymbol{\mu})/s_N(\boldsymbol{\mu}) = 1 + ((s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}))/s_N(\boldsymbol{\mu})) \leq 1 + \Delta_N^{s,\rm rel}(\boldsymbol{\mu})$ (since $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) \leq \Delta_N^s(\boldsymbol{\mu})$ from (4.28b), and $\Delta_N^{s,\rm rel}(\boldsymbol{\mu}) = \Delta_N^s(\boldsymbol{\mu})/s_N(\boldsymbol{\mu})$) we recover from (4.32) and (4.28b) the right inequality in (4.28c) under our (verifiable) hypothesis $\Delta_N^{s,\rm rel}(\boldsymbol{\mu}) \leq 1$. ■

Note that $\alpha_{\rm LB}(\boldsymbol{\mu})$ and $\gamma_{\rm UB}(\boldsymbol{\mu})$ in Proposition 4C refer to $\Theta_{a{\rm S}}^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})$ and $\Theta_a^{\max,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})$, respectively.

We conclude from Proposition 4C that the estimators $\Delta_N^{\rm en}(\boldsymbol{\mu})$, $\Delta_N^s(\boldsymbol{\mu})$, and $\Delta_N^{s,\rm rel}(\boldsymbol{\mu})$ are in fact *rigorous upper bounds* for the RB error in the energy norm, the RB output error, and the RB relative output error, respectively. Furthermore, the effectivity of the energy-norm and output error estimators is bounded from above independent of N. We shall return to a more quantitative discussion of the estimator effectivities — and associated implications — in subsequent sections.

4.3.3 $\boldsymbol{X}$-Norm Error Bounds

Although our bounds on the output are arguably the most relevant, it will also prove useful (e.g., in visualization contexts) to provide a certificate of fidelity for the full field variable $u(\boldsymbol{\mu})$ in a norm which is independent of $\boldsymbol{\mu}$. Towards that end, we introduce the error estimators for the X-norm and relative X-norm,

$$\Delta_N(\boldsymbol{\mu}) \equiv \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})} , \tag{4.33a}$$

$$\Delta_N^{\mathrm{rel}}(\boldsymbol{\mu}) \equiv 2\frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{\alpha_{\mathrm{UB}}(\boldsymbol{\mu})\,\|u_N(\boldsymbol{\mu})\|_X} , \tag{4.33b}$$

respectively, and associated effectivities,

$$\eta_N(\boldsymbol{\mu}) \equiv \frac{\Delta_N(\boldsymbol{\mu})}{\|e(\boldsymbol{\mu})\|_X} , \tag{4.34a}$$

and

$$\eta_N^{\mathrm{rel}}(\boldsymbol{\mu}) \equiv \frac{\Delta_N^{\mathrm{rel}}(\boldsymbol{\mu})}{(\|e(\boldsymbol{\mu})\|_X/\|u(\boldsymbol{\mu})\|_X)} . \tag{4.34b}$$

Again, our goal is effectivities ≥ 1 but very close to 1.

We can then prove

Proposition 4D. *For any $N = 1, \ldots, N_{\max}$, the effectivity (4.34a) satisfies*

$$1 \leq \eta_N(\boldsymbol{\mu}) \leq \theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} . \tag{4.35}$$

Furthermore, for $\Delta_N^{\mathrm{rel}}(\boldsymbol{\mu}) \leq 1$ the effectivity (4.34b) satisfies

$$1 \leq \eta_N^{\mathrm{rel}}(\boldsymbol{\mu}) \leq 3\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) ; \tag{4.36}$$

note in this case even the left inequality is conditional.

Proof. The left inequality of (4.35) follows directly from (4.28a) of Proposition 4C, $|||e(\boldsymbol{\mu})|||^2_{\boldsymbol{\mu}} \geq \alpha_{\mathrm{LB}}(\boldsymbol{\mu})\, \|e(\boldsymbol{\mu})\|^2_X$, and the definition of $\Delta_N(\boldsymbol{\mu})$, (4.33a); the right inequality of (4.35) follows directly from (4.28a) of Proposition 4C, $|||e(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \gamma^{1/2}_{\mathrm{UB}}(\boldsymbol{\mu})\, \|e(\boldsymbol{\mu})\|_X$, the definition of $\Delta_N(\boldsymbol{\mu})$ in (4.33a) and (4.16).

To demonstrate (4.36), we first note that

$$\eta^{\mathrm{rel}}_N(\boldsymbol{\mu}) = 2\frac{\|u(\boldsymbol{\mu})\|_X}{\|u_N(\boldsymbol{\mu})\|_X}\, \eta_N(\boldsymbol{\mu}) = 2\left(1 + \frac{\|u(\boldsymbol{\mu})\|_X - \|u_N(\boldsymbol{\mu})\|_X}{\|u_N(\boldsymbol{\mu})\|_X}\right)\eta_N(\boldsymbol{\mu}) \ . \tag{4.37}$$

We then observe that, since $\big|\|u(\boldsymbol{\mu})\|_X - \|u_N(\boldsymbol{\mu})\|_X\big|/\|u_N(\boldsymbol{\mu})\| \leq \|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X/\|u_N(\boldsymbol{\mu})\| \leq \frac{1}{2}\Delta^{\mathrm{rel}}_N(\boldsymbol{\mu})$ (from (4.34a), (4.35), (4.33a), and (4.33b)) $\leq \frac{1}{2}$ (from our verifiable hypothesis),

$$\frac{1}{2} \leq 1 + \frac{\|u(\boldsymbol{\mu})\|_X - \|u_N(\boldsymbol{\mu})\|_X}{\|u_N(\boldsymbol{\mu})\|_X} \leq \frac{3}{2} \ . \tag{4.38}$$

The result (4.36) then directly follows from (4.35), (4.37), and (4.38). ■

(Note we can improve the effectivity upper bound of (4.36) (by different choices of prefactors) but at the expense of a more restrictive hypothesis on $\Delta^{\mathrm{rel}}_N(\boldsymbol{\mu})$.) Not surprisingly (given the *a priori* result of Proposition 3A), we lose a factor of $\sqrt{\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})}$ in the X-norm result (4.35), relative to the energy norm result, (4.28a).

4.3.4 Measures and Implications of Sharpness

To begin, we introduce a test sample $\Xi_{\mathrm{test}} \subset \mathcal{D}$ of size n_{test}. Then for ($^\bullet$ =) "en", "s," "s, rel," or "rel," we define the maximum effectivity and the average effectivity as

$$\eta^\bullet_{N,\max} \equiv \max_{\mu \in \Xi_{\mathrm{test}}} \eta^\bullet_N(\boldsymbol{\mu}) \ , \tag{4.39}$$

and

$$\eta^\bullet_{N,\mathrm{ave}} \equiv \frac{1}{n_{\mathrm{test}}} \sum_{\boldsymbol{\mu} \in \Xi_{\mathrm{test}}} \eta^\bullet_N(\boldsymbol{\mu}) \ , \tag{4.40}$$

respectively. Clearly, $\eta^\bullet_{N,\max}$ measures worst-case behavior, and $\eta^\bullet_{N,\mathrm{ave}}$ measures "expected" behavior. (We may also consider median behavior if we wish to further reduce the effect of outliers.)

It follows from Proposition 4C that (say, for the absolute output error)

$$\eta^s_{N,\max} \le \max_{\boldsymbol{\mu}\in\Xi_{\text{test}}} \theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \le \max_{\boldsymbol{\mu}\in\mathcal{D}} \theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \equiv \eta^s_{\max,\text{UB}} \ . \tag{4.41}$$

This upper bound is independent of N, and hence the error bounds are well-defined as N increases. Furthermore, and perhaps even more importantly, our upper bound is independent of $\mathcal{N}_{\text{t}}$, the dimension of the truth approximation: our error bounds are stable as $\mathcal{N}_{\text{t}} \to \infty$; this reflects the proper choice of norm consistent with the exact (continuous) formulation, $H^1(\Omega)$. However, $\eta^s_{\max,\text{UB}}$ can be quite large, as we shall see in our examples. (In some cases — *P small* — we can improve the effectivity with "multi-point" inner products; we discuss this extension in Section 4.5.)

We make several comments. First, in many cases, the upper bound is quite pessimistic due to the various inequalities and associated "worst-case" alignment assumptions in the proofs of Proposition 4C; we provide some numerical evidence shortly. Second, within the (often) exponentially convergent RB context, effectivities of $O(10)$ or even $O(100)$ — anathema in the FE context — are not too unacceptable: the increased N' required to satisfy $\varepsilon^s_{\text{tol}} = \Delta^s_N(\boldsymbol{\mu})$ will be only *modestly larger* than the N'' required to satisfy $\varepsilon^s_{\text{tol}} = s(\boldsymbol{\mu}) - s_{N''}(\boldsymbol{\mu})$. For example, if we assume that $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) = C_1 e^{-C_2 N}$, $N' - N'' \le (\ln \eta^s_{\max,\text{UB}})/C_2$: the effect is *additive* and *logarithmic* in the effectivity.

We also note that the max effectivity is not always the most relevant measure of performance. In particular, in the greedy algorithm, it is certainly important that the *large* errors are accurately predicted; however, the *smaller* errors play no role in the selection process. More generally, it is typically the larger errors that will be of greatest concern in applications. We thus introduce an alternative estimator performance measure, the "*ratio of maxima*": the ratio of the maximum predicted error to the maximum actual error. (In contrast, $\eta^\bullet_{N,\max}$ is the *maximum of the ratio* of the predicted error to the actual error.)

We give here the "ratio of maxima" definition for $\bullet = s$ (though the other norms admit analogous measures). To wit, we define

$$\rho^s_{\mathrm{err},N} = \frac{\displaystyle\max_{\boldsymbol{\mu}\in\Xi_{\mathrm{test}}} \Delta^s_N(\boldsymbol{\mu})}{\displaystyle\max_{\boldsymbol{\mu}\in\Xi_{\mathrm{test}}} (s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}))} \ . \tag{4.42}$$

As already motivated, this measure is particularly significant within the Greedy$^{\mathrm{out}}$ algorithm of Section 3.4.4: if we take $\Xi_{\mathrm{test}} \equiv \Xi_{\mathrm{train}}$, then we directly obtain (for absolute error — $\omega_N(\boldsymbol{\mu}) = 1$)

$$\frac{\varepsilon^{\mathrm{out},*}_N}{\overline{\varepsilon}^{\mathrm{out},*}_N} = \sqrt{\rho^s_{\mathrm{err},N}} \ . \tag{4.43}$$

Hence, as anticipated, if $\rho^s_{\mathrm{err},N}$ is reasonably close to unity — and even if $\eta^s_{N,\max}$ is very large — our error bound is a good surrogate for the true error in the greedy selection process. (Of course, $\rho^s_{\mathrm{err},N} \sim O(1)$ does not necessarily imply that Δ^s_N and $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$ attain these maxima at the same point in $\mathcal{D}$.)

4.3.5 Numerical Results: ThermalBlock

We consider here the ThermalBlock problem, Ex1 of Section 2.2.1.

$P = 1$ Parameter

We first investigate the case $B_1 = 2$, $B_2 = 1$, and hence $P = 1$, analyzed from an *a priori* perspective in Section 3.5.2. As before, $1/\mu_1^{\min} = \mu_1^{\max} = \sqrt{\mu_{\mathrm{r}}} = 10$ and hence $\mu_1^{\max}/\mu_1^{\min} = \mu_{\mathrm{r}} = 100$. For our inner product, we choose $\overline{\mu}_1 = 1$: it is precisely for our error estimators that the choice of $\overline{\boldsymbol{\mu}}$ is important; as we shall see (in Section 4.5), $\overline{\boldsymbol{\mu}} = 1$ — the "logarithmic center" of $\mathcal{D}$ — is in fact optimal. For the truth discretization, we take $\mathcal{N}_{\mathrm{t}} = 256$: we confirm, per the theory, that the effectivities are insensitive to $\mathcal{N}_{\mathrm{t}}$ for sufficiently large $\mathcal{N}_{\mathrm{t}}$.

Our RB approximation is generated by a Greedy$^{\mathrm{out}}$ approach: we take $\omega_N(\boldsymbol{\mu}) = \omega(\boldsymbol{\mu}) = 1$ (in this section for convenience we focus on *absolute* output errors since the effectivity results are unconditional and hence more succinctly described); and we choose $\Xi_{\mathrm{train}} = G^{\ln}_{[\mathrm{MC};10^4]}$ —

$n_{\text{train}} = 10^4$ is certainly adequate for a single parameter. Hence our RB approximation is given by $\text{Greedy}^{\text{out}}(N_0 = 1,\, \mu_1^{1\,\text{out},*} = 1,\, \Xi_{\text{train}},\, \varepsilon_{\text{tol,min}},\, \omega_N(\mu_1) = 1)$.

For this case, it is simple to derive (recall $\overline{\mu}_1 = 1$) that

$$\theta^{\overline{\boldsymbol{\mu}}}(\mu_1) = \text{Max}\left(\frac{1}{\mu_1}, \mu_1\right) \quad ; \tag{4.44}$$

clearly,

$$\theta^{\overline{\boldsymbol{\mu}}}(\mu_1) \leq \sqrt{\mu_{\text{r}}}, \qquad \forall\, \mu_1 \in \mathcal{D} \ . \tag{4.45}$$

It can then be shown that (for $n_{\text{test}} \to \infty$) for $N = 1, \ldots, N_{\max}$

$$\eta^s_{N,\text{ave}}(\mu_1) \leq \eta^s_{\text{ave,UB}} \equiv \frac{2(\sqrt{\mu_{\text{r}}} - 1)}{\ln \mu_{\text{r}}}, \qquad \forall\, \mu_1 \in \mathcal{D} \ , \tag{4.46}$$

and that furthermore (directly from (4.41) and (4.45))

$$\eta^s_{N,\max}(\mu_1) \leq \eta^s_{\max,\text{UB}} \equiv \sqrt{\mu_{\text{r}}}, \qquad \forall\, \mu_1 \in \mathcal{D} \ . \tag{4.47}$$

As expected, the bounds are independent of N (and $\mathcal{N}_{\text{t}}$), and $\eta^s_{\text{ave,UB}}$ is less than $\eta^s_{\max,\text{UB}}$. Note also the *relatively weak dependence* of both $\eta^s_{\text{ave,UB}}$ and $\eta^s_{\max,\text{UB}}$ on the extent of the parameter domain, μ_{r}.

We present in Table 4.1 $\Delta^s_{N,\max} = \max_{\boldsymbol{\mu} \in \Xi_{\text{test}}} \Delta^s_N(\boldsymbol{\mu})$, $\eta^s_{N,\text{ave}}$, $\eta^s_{N,\max}$, and $\rho^s_{\text{err},N}$ as a function of N; for these results we choose $\Xi_{\text{test}} = G^{\ln}_{[\text{MC};10^4]}$. We observe that $\eta^s_{N,\text{ave}} \leq \eta^s_{\text{ave,UB}} = 7.81$, and that furthermore our theoretical upper bound is not overly pessimistic. Similarly, we obtain $\eta^s_{N,\max} \leq \eta^s_{\max,\text{UB}} = 10$, and again note that our theoretical bound is (unfortunately) quite sharp. Finally, we note that $\rho^s_{\text{err},N} \leq \eta^s_{N,\max}$, as must be the case from the respective definitions (4.41), (4.42) of these two metrics; $\rho^s_{\text{err},N}$ is in fact quite close to unity for all N.

$\boldsymbol{P = 8}$ Parameters

We now investigate the case $B_1 = 3$, $B_2 = 3$, and hence $P = 8$, analyzed from an (empirical) *a priori* perspective in Section 3.5.3. As before, $1/\mu_i^{\min} = \mu_i^{\max} = \sqrt{\mu_r}$ $(= 10)$, $1 \leq i \leq P$, and hence $\mu_i^{\max}/\mu_i^{\min} = \mu_r$ $(= 100)$, $1 \leq i \leq P$; we take $\overline{\mu}_i = 1$, $1 \leq i \leq P$, the "optimality"

N	$\Delta^s_{N,\max}$	$\eta^s_{N,\text{ave}}$	$\eta^s_{N,\max}$	$\rho^s_{\text{err},N}$
1	7.2084E+00	2.3417	3.3305	3.2508
2	4.5371E−01	2.4858	3.6850	1.5630
3	6.9652E−04	6.2195	9.8551	3.4143
4	1.3744E−07	3.3219	7.2632	2.1666
5	3.1140E−11	6.0789	7.0453	2.1823

Table 4.1: Ex1, ThermalBlock for $B_1 = 2$, $B_2 = 1$, $(P = 1)$: output error bounds and effectivities.

of which shall be analyzed in Section 4.5. For our truth approximation we take $\mathcal{N}_\text{t} = 661$; we again confirm that the effectivities are insensitive to $\mathcal{N}_\text{t}$.

Our RB approximation is generated by a Greedy$^\text{out}$ approach for $\omega_N(\boldsymbol{\mu}) = \omega(\boldsymbol{\mu}) = 1$ (absolute output error) and $\Xi_\text{train} = G^\text{ln}_{[\text{MC};10^6]}$. (Note $n_\text{train} = 10^6$ is really too small to completely characterize the error over $\mathcal{D}$; however we can and will of course always confirm the accuracy (for any given $\boldsymbol{\mu}$) Online via $\Delta^s_N(\boldsymbol{\mu})$.) Hence our RB approximation is given by Greedy$^\text{out}$($N_0 = 1$, $\boldsymbol{\mu}^{1\,\text{out},*} = (1,\ldots,1)$, Ξ_train, $\varepsilon_\text{tol,min}$, $\omega_N(\boldsymbol{\mu}) = 1$).

For this case, we can derive (recall $\overline{\boldsymbol{\mu}} = (1,\ldots,1)$) that

$$\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) = \text{Max}\left[1, \tfrac{1}{\mu_1}, \tfrac{1}{\mu_2}, \ldots, \tfrac{1}{\mu_P}\right] \times \text{Max}\left[1, \mu_1, \ldots, \mu_P\right] \ ; \tag{4.48}$$

hence,

$$\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \le \mu_\text{r}, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ . \tag{4.49}$$

It is no longer simple to derive an upper bound for $\eta^s_{N,\text{ave}}$. However, it is clear that for $N = 1, \ldots, N_\text{max}$,

$$\eta^s_\text{max}(\boldsymbol{\mu}) \le \eta^s_\text{max,UB} \equiv \mu_\text{r}, \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D} \ . \tag{4.50}$$

Note the stronger dependence on μ_r in the $P > 1$ case.

We present in Table 4.2 $\Delta^s_{N,\max} = \max_{\boldsymbol{\mu}\in\Xi_\text{test}} \Delta^s_N(\boldsymbol{\mu})$, $\eta^s_{N,\text{ave}}$, $\eta^s_{N,\max}$, and $\rho^s_{\text{err},N}$ as a function of N; for these results we choose for our test sample $\Xi_\text{test} = G^\text{ln}_{[\text{MC};10^6]}$ — again too small, but adequate for our purposes here. We observe that $\eta^s_{N,\max} \le \mu_\text{r} = 100$; unfortunately, the theoretical bound is reasonably tight, and hence $\eta^s_{N,\max}$ is quite large. However, $\rho^s_{\text{err},N}$, the

N	$\Delta^s_{N,\max}$	$\eta^s_{N,\mathrm{ave}}$	$\eta^s_{N,\max}$	$\rho^s_{\mathrm{err},N}$
5	8.2199E+00	5.6395	28.5220	7.5180
10	2.2036E+00	6.7067	31.2850	4.8877
15	8.2560E−01	7.4207	32.9266	11.5448
20	2.0020E−01	7.5587	37.3024	11.8552
25	7.1300E−02	7.9920	36.6976	18.7523
30	1.5100E−02	12.1138	62.2537	18.3489
35	5.2000E−03	16.4900	84.2649	32.1640
40	1.2000E−03	14.4598	73.1151	25.9760
45	3.0000E−04	10.0536	56.6545	14.9053
50	1.0000E−04	10.2566	57.5113	22.4168
55	3.0000E−05	9.3783	60.7000	13.8695
60	1.0000E−05	8.0103	43.3108	14.7932
65	6.0000E−06	7.5970	53.7690	15.2386
70	2.0000E−06	8.4598	36.5435	10.4904
75	6.0000E−07	7.6310	31.7752	13.1075
80	8.0000E−08	7.3846	37.4073	12.6413
85	1.0000E−08	7.5917	38.8586	12.8422
90	5.0000E−09	8.6520	57.9131	9.7080
95	1.0000E−09	8.8307	62.0965	15.4785

Table 4.2: Ex1, ThermalBlock for $B_1 = 3$, $B_2 = 3$, ($P = 8$): output error bounds and effectivities.

arguably more relevant metric, is considerably smaller than $\eta^s_{N,\max}$: as expected (and as can be confirmed by a scatter plot of $\eta^s_N(\boldsymbol{\mu})$ vs. $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \Xi_{\mathrm{test}}$), the largest effectivities are associated with the smaller errors. Furthermore, the effect of overestimation is reasonably small given the rapid convergence of the RB approximation. (For example, the N' required to achieve $\Delta^s_{N',\max} = 0.01$ is $N' = 33$; the N'' required to achieve $\max_{\boldsymbol{\mu}\in\Xi_{\mathrm{test}}} s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$ $(= \Delta^s_{N'',\max}/\rho^s_{\mathrm{err},N}) = 0.01$ is only modestly smaller, $N'' = 23$.)

In Section 4.5 we shall consider an improvement upon the results presented here. At the same time, we can better understand the "best" choice for $\overline{\boldsymbol{\mu}}$ to minimize the effectivity and hence sharpen the bounds. However, we must first address the issue of Offline-Online effort.

4.4 Offline-Online Computational Procedures

The error bounds of the previous section are of no utility without an accompanying Offline-Online computational approach.

4.4.1 Ingredients

Dual Norm of Residual

The computationally crucial component of all the error bounds of the previous section is $\|\hat{e}(\boldsymbol{\mu})\|_X$, the dual norm of the residual. To develop an Offline-Online procedure, we first expand the residual (4.22) as

$$r(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu})\, f^q(v) - \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \Theta_a^q(\boldsymbol{\mu})\, u_{N\,n}(\boldsymbol{\mu})\, a^q(\zeta_n, v), \qquad \forall\, v \in X \ ; \tag{4.51}$$

(4.51) follows directly from our affine assumption (4.5) and our RB representation (3.28). It is clear $r(v;\boldsymbol{\mu})$ can be expressed as a sum of

$$Q_N \equiv Q_f + Q_a N \tag{4.52}$$

products of parameter-dependent functions and parameter-independent linear functionals.

To render this identification more apparent, we define $\underline{\mathcal{E}}_N$: $\mathcal{D} \rightarrow \mathbb{R}^{Q_N}$ as

$$\begin{aligned} \underline{\mathcal{E}}_N(\boldsymbol{\mu}) = \Big(\ & \Theta_f^1(\boldsymbol{\mu}), \ldots, \Theta_f^{Q_f}(\boldsymbol{\mu}) \ , \\ & \Theta_a^1(\boldsymbol{\mu})\, u_{N\,1}(\boldsymbol{\mu}), \ldots, \Theta_a^{Q_a}(\boldsymbol{\mu})\, u_{N\,1}(\boldsymbol{\mu}) \ , \\ & \Theta_a^1(\boldsymbol{\mu})\, u_{N\,2}(\boldsymbol{\mu}), \ldots, \Theta_a^{Q_a}(\boldsymbol{\mu})\, u_{N\,2}(\boldsymbol{\mu}) \ , \\ & \qquad \vdots \\ & \Theta_a^1(\boldsymbol{\mu})\, u_{N\,N}(\boldsymbol{\mu}), \ldots, \Theta_a^{Q_a}(\boldsymbol{\mu})\, u_{N\,N}(\boldsymbol{\mu}) \Big)^{\mathrm{T}} \end{aligned} \tag{4.53}$$

and $\underline{h}_N \in (X')^{Q_N}$ as

$$\begin{aligned}\underline{h}_N(v) = \Big(\; & f^1(v), \dots, f^{Q_f}(v) \;, \\ & -a^1(\zeta_1, v), \dots, -a^{Q_a}(\zeta_1, v) \;, \\ & -a^1(\zeta_2, v), \dots, -a^{Q_a}(\zeta_2, v) \;, \\ & \qquad\qquad \vdots \\ & -a^1(\zeta_N, v), \dots, -a^{Q_a}(\zeta_N, v) \Big)^{\mathrm{T}} \;.\end{aligned} \tag{4.54}$$

We may then write, from (4.51), (4.53), and (4.54),

$$r(v; \boldsymbol{\mu}) = \sum_{n=1}^{Q_N} \mathcal{E}_{N\,n}(\boldsymbol{\mu})\, h_{N\,n}(v), \qquad \forall\, v \in X \;, \tag{4.55}$$

where $\underline{\mathcal{E}}_N = (\mathcal{E}_{N\,1}, \dots, \mathcal{E}_{N\,Q_N})^{\mathrm{T}}$ and $\underline{h}_N = (h_{N\,1}, \dots, h_{N\,Q_N})^{\mathrm{T}}$.

It follows directly from (4.55) and (4.23) that $\hat{e}(\boldsymbol{\mu}) \in X$ satisfies

$$(\hat{e}(\boldsymbol{\mu}), v)_X = \sum_{n=1}^{Q_N} \mathcal{E}_{N\,n}(\boldsymbol{\mu})\, h_{N\,n}(v), \qquad \forall\, v \in X \;, \tag{4.56}$$

and hence that

$$\hat{e}(\boldsymbol{\mu}) = \sum_{n=1}^{Q_N} \mathcal{E}_{N\,n}(\boldsymbol{\mu})\, \hat{g}_{N\,n} \;, \tag{4.57}$$

where

$$(\hat{g}_{N\,n}, v)_X = h_{N\,n}(v), \qquad \forall\, v \in X,\ 1 \le n \le Q_N \;. \tag{4.58}$$

The $\hat{g}_{N\,n}$, $1 \le n \le Q_N$, hence satisfy *parameter-independent* scalar (or vector) Poisson-like (or elasticity-like) problems.

We can now readily construct $\|\hat{e}(\boldsymbol{\mu})\|_X^2$ from (4.56) as

$$\|\hat{e}(\boldsymbol{\mu})\|_X^2 = \sum_{n=1}^{Q_N} \sum_{m=1}^{Q_N} \mathcal{E}_{N\,n}(\boldsymbol{\mu})\, \mathcal{E}_{N\,m}(\boldsymbol{\mu})\ (\hat{g}_{N\,n}, \hat{g}_{N\,m})_X \;\;. \tag{4.59}$$

We thus introduce $\underline{\mathbb{G}}_N \in \mathbb{R}^{Q_N \times Q_N}$ as

$$\mathbb{G}_{N\,n\,m} = (\hat{g}_{N\,n}, \hat{g}_{N\,m})\,, \qquad 1 \le n, m \le Q_N \;, \tag{4.60}$$

in terms of which we can express the dual norm of the residual as

$$\|\hat{e}(\boldsymbol{\mu})\|_X = \left(\sum_{n=1}^{Q_N} \sum_{m=1}^{Q_N} \mathcal{E}_{N\,n}(\boldsymbol{\mu})\, \mathcal{E}_{N\,m}(\boldsymbol{\mu})\, \mathbb{G}_{N\,n\,m} \right)^{\frac{1}{2}} \tag{4.61}$$

(or, in matrix form, as $\|\hat{e}(\boldsymbol{\mu})\|_X = (\underline{\mathcal{E}}_N^{\mathrm{T}}(\boldsymbol{\mu})\, \underline{\mathbb{G}}_N\, \underline{\mathcal{E}}_N(\boldsymbol{\mu}))^{1/2}$).

Before proceeding, we provide a more explicit representation of $\underline{\mathbb{G}}_N$. To begin we note that, for any $v = \sum_{i=1}^{\mathcal{N}_{\mathrm{t}}} v_i\, \varphi_i \in X$,

$$\underline{h}_N(v) = \underline{\mathbb{H}}_N^{\mathrm{T}}\, \underline{v} \ , \tag{4.62}$$

where $\underline{\mathbb{H}}_N \in \mathbb{R}^{\mathcal{N}_{\mathrm{t}} \times N}$ is given by

$$\mathbb{H}_{N\,i\,n} = h_{N\,n}(\varphi_i), \qquad 1 \le i \le \mathcal{N}_{\mathrm{t}},\ 1 \le n \le N \ . \tag{4.63}$$

(Recall that $\{\varphi_i\}_{1 \le i \le \mathcal{N}_{\mathrm{t}}}$ is the basis set for our FE truth approximation space.) It is then readily derived from (4.58), (4.63), and (2.45) that

$$\underline{\mathbb{G}}_N = \underline{\mathbb{H}}_N^{\mathrm{T}}\, \underline{\mathbb{X}}^{-1}\, \underline{\mathbb{H}}_N, \qquad 1 \le n, m \le Q_N \ ; \tag{4.64}$$

recall $\underline{\mathbb{X}} = \underline{\mathbb{X}}^{\mathcal{N}_{\mathrm{t}}}$ is our truth “inner product” matrix.

The Offline-Online decomposition is now clear. In the Offline stage we form the parameter-independent quantity $\underline{\mathbb{G}}_N$ via (4.64). Then, in the Online stage, given any “new” value of $\boldsymbol{\mu}$ — and $\Theta_f^q(\boldsymbol{\mu})$, $1 \le q \le Q_f$, $\Theta_a^q(\boldsymbol{\mu})$, $1 \le q \le Q_a$, $u_{N\,n}(\boldsymbol{\mu})$, $1 \le n \le N$, and hence $\underline{\mathcal{E}}_N(\boldsymbol{\mu})$ — we simply perform the sum (4.61): the Online operation count is $Q_N^2 = (Q_f + Q_a N)^2$ and clearly independent of $\mathcal{N}_{\mathrm{t}}$. We provide more details and analysis of the Offline-Online procedure below.

Stability Factors and Normalizations

In addition to $\|\hat{e}(\boldsymbol{\mu})\|_X$, our error bounds require computation of $\alpha_{\mathrm{LB}}(\boldsymbol{\mu})$ of Section 4.2; typically no Offline effort, and only $O(Q_a)$ operations Online — and hence negligible. (Note, however, that for problems that are not parametrically coercive, the stability constant lower bound is far from trivial computationally: we address this in Part II (for general coercive operators) and in Part III (for non-coercive operators).)

We must also compute, for the relative measures, the normalizations $s_N(\boldsymbol{\mu})$ and $\|u_N(\boldsymbol{\mu})\|_X$. Since $s_N(\boldsymbol{\mu})$ is already provided by the Online RB procedure (of Section 3.3), only $\|u_N(\boldsymbol{\mu})\|_X$ is "new." To compute $\|u_N(\boldsymbol{\mu})\|_X$ we need only note that

$$\begin{aligned} \|u_N(\boldsymbol{\mu})\|_X &= a^{\frac{1}{2}}(u_N(\boldsymbol{\mu}), u_N(\boldsymbol{\mu}); \overline{\boldsymbol{\mu}}) \\ &= \left(\underline{u}_N^{\mathrm{T}}(\boldsymbol{\mu})\, \underline{A}_N(\overline{\boldsymbol{\mu}})\, \underline{u}_N(\boldsymbol{\mu})\right)^{\frac{1}{2}} \\ &= \left(\underline{u}_N^{\mathrm{T}}(\boldsymbol{\mu})\, \underline{\mathbb{X}}_N\, \underline{u}_N(\boldsymbol{\mu})\right)^{\frac{1}{2}} \ , \end{aligned} \tag{4.65}$$

where $\underline{\mathbb{X}}_N \in \mathbb{R}^{N \times N}$ is the RB X-inner product matrix

$$\underline{\mathbb{X}}_N \equiv \underline{\mathbb{Z}}_N\, \underline{\mathbb{X}}^{\mathcal{N}_{\mathrm{t}}}\, \underline{\mathbb{Z}}_N \ . \tag{4.66}$$

The Offline-Online decomposition is apparent. In the Offline stage we form the parameter-independent RB inner-product matrix, $\underline{\mathbb{X}}_N \in \mathbb{R}^{N \times N}$. Then, in the Online stage, given any "new" value $\boldsymbol{\mu}$, we need only perform the inner product (4.65): the Online operation count is N^2, and clearly independent of $\mathcal{N}_{\mathrm{t}}$.

4.4.2 Operation Count and Storage

We can now succinctly describe the Offline and Online stage and provide associated operation counts and storage. In the Offline stage, we must first form the $\underline{\mathbb{H}}_{N_{\max}} \in \mathbb{R}^{\mathcal{N}_{\mathrm{t}} \times Q_{N_{\max}}}$ — (essentially) $Q_a N_{\max}$ $\underline{A}^{\mathcal{N}_{\mathrm{t}}}$-matvecs and $Q_{N_{\max}} \mathcal{N}_{\mathrm{t}}$ "temporary" (more precisely, Offline-only) storage. We next find $\underline{\mathbb{X}}^{-1}\, \underline{\mathbb{H}}_{N_{\max}}$ — $\underline{\mathbb{X}}^{\mathcal{N}_{\mathrm{t}}}$-solve($Q_{N_{\max}}$) operations and $Q_{N_{\max}} \mathcal{N}_{\mathrm{t}}$ "temporary" (more precisely, Offline-only) storage. Finally, we form $\underline{\mathbb{G}}_{N_{\max}} = \underline{\mathbb{H}}^{\mathrm{T}}_{N_{\max}}\, (\underline{\mathbb{X}}^{-1}\, \underline{\mathbb{H}}_{N_{\max}})$ — $Q^2_{N_{\max}}$ $X^{\mathcal{N}}$-inprods.

It is clear that, for the Offline stage to be efficient (as possible), we must exploit two properties: first, because of the large number ($Q^2_{N_{\max}}$) of matrix vector products, we must take advantage of the sparsity in the truth FE stiffness matrix (or other artifices permitting rapid "action"); second, because of the large number ($Q_{N_{\max}}$) of "$\underline{\mathbb{X}}$" solves, we must take advantage of the parameter-independent nature of $\underline{\mathbb{X}}$ — for example, in the direct context, we effect in a pre-processing step a minimum-fill-in re-ordering [124] followed by a Cholesky decomposition.

The link between the Offline and Online stages is the "permanent" storage of quantities computed in the Offline stage and then invoked in the Online stage. The item to be stored, in essense the Online storage, is $\underline{\mathbb{G}}_{N_{\max}} \in \mathbb{R}^{Q_{N_{\max}} \times Q_{N_{\max}}}$ — $(Q_f + Q_a N_{\max})^2$ words. We again emphasize the importance of our *hierarchical* RB approximation: for any given N (given our judicious ordering (4.53), (4.54)), $\underline{\mathbb{G}}_N$ is the $(Q_N \times Q_N)$ principal submatrix of $\underline{\mathbb{G}}_{N_{\max}}$; we thus need only compute and store $\underline{\mathbb{G}}_{N_{\max}}$ in the Offline stage, and then extract the requisite $\underline{\mathbb{G}}_N$ in the Online stage.

In the Online stage, given any "new" $\boldsymbol{\mu}$ (and the RB solution $\underline{u}_N(\boldsymbol{\mu})$) we need only evaluate the inner product (4.61) — $Q_N^2 = (Q_f + Q_a N)^2$ operations. The crucial point, as always, is that the Online operation count and storage — not only for $\underline{u}_N(\boldsymbol{\mu})$ and $s_N(\boldsymbol{\mu})$ *but now also for the error bound* $\Delta_N^s(\boldsymbol{\mu})$ (from $\|\hat{e}(\boldsymbol{\mu})\|_X$) — is *independent of* $\mathcal{N}_{\mathrm{t}}$: we can thus provide real-time *and* reliable prediction. As a corollary of our $\mathcal{N}_{\mathrm{t}}$-independent *marginal* cost we note that the *average* cost to evaluate $\Delta_N^s(\boldsymbol{\mu})$ over a sample Ξ_{train} of size n_{train} is independent of $\mathcal{N}_{\mathrm{t}}$ as $n_{\mathrm{train}} \to \infty$: this provides the search efficiency required by the greedy algorithm.

If we compare the Online cost to evaluate $s_N(\boldsymbol{\mu})$ — essentially $N^3 + Q_a N^2$ — to the Online cost to evaluate $\Delta_N^s(\boldsymbol{\mu})$ — essentially $Q_a^2 N^2$ — we conclude that for N small $\Delta_N^s(\boldsymbol{\mu})$ will dominate (due to the Q_a^2 scaling) whereas for N large $s_N(\boldsymbol{\mu})$ will dominate (due to the N^3 scaling). In actual practice, for reasonably high accuracy, the costs of $s_N(\boldsymbol{\mu})$ and $\Delta_N(\boldsymbol{\mu})$ are typically commensurate — at least for Q_a not too large.

Finally, we close with a brief note on round-off effects. We note that each term in the sum (4.61) is in fact $O(1)$, and thus — given that $\|\hat{e}(\boldsymbol{\mu})\|_X$ is small (for larger N) — there is significant cancellation. We conclude that for $\|\hat{e}(\boldsymbol{\mu})\|_X^2 \sim$ machine precision the dual norm of the residual will no longer be reliable. Fortunately, for this complicant case, $\Delta_N^s(\boldsymbol{\mu}) \sim \|\hat{e}(\boldsymbol{\mu})\|_X^2$, and hence the round-off errors generated by the superposition/summation will only be observed for academically (i.e., ridiculously) small errors. However, this will be somewhat less the case for the non-compliant problems treated in Part II and Part III.

4.4.3 "Modalities"

There are two fashions in which we exploit our error bounds.

The first fashion is in the Offline stage, in the Greedy algorithms, as discussed in Sections 3.4.3 and 3.4.4. In this case we take advantage of the $\mathcal{N}_{\mathrm{t}}$-independent average cost $\boldsymbol{\mu} \to \Delta_N^s(\boldsymbol{\mu})$ in the limit of many queries. We have already provided the greedy operation count in Section 3.4.3, (3.67). From Section 4.4.2 we now understand the contributions from the error bound: the second line of (3.67) — due to formation of $\mathbb{H}$ and $\mathbb{G}$; and the $O(Q_a^2 N_{\max}^3)$ (in fact, $O(Q_N^2 N_{\max}^3)$) term — due to evaluation of Δ_N^s over Ξ_{train} for $1 \leq N \leq N_{\max}$.

The second fashion is in the Online stage. In particular, given any desired output accuracy $\varepsilon_{\mathrm{des}} \geq \varepsilon_{\mathrm{tol,min}}$ and any *particular* new value of $\boldsymbol{\mu}$, we would like to obtain $s_N(\boldsymbol{\mu})$ such that $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) \leq \varepsilon_{\mathrm{des}}$. (We consider the absolute error; a similar procedure applies to relative error.) Ideally, we would search for the smaller N, $N^*(\boldsymbol{\mu}, \varepsilon_{\mathrm{des}})$, such that $\Delta_N^s(\boldsymbol{\mu}) \leq \varepsilon_{\mathrm{des}}$. In practice, to ensure that the search costs do not predominate, we settle for the following supoptimal "coarse" result. We first set N^0 to be the smallest N such that $\varepsilon_N^* \leq \varepsilon_{\mathrm{des}}$ for all $N' \geq N$ (a simple search of $N_{\max}$ entries), and compute $\Delta_{N^0}^s(\boldsymbol{\mu})$. If $\Delta_{N^0}^s(\boldsymbol{\mu}) \leq \varepsilon_{\mathrm{des}}$, we choose $N = N^0$, evaluate $s_{N^0}(\boldsymbol{\mu})$ and terminate; if $\Delta_{N^0}^s(\boldsymbol{\mu}) > \varepsilon_{\mathrm{des}}$, we set $N' = N^0 + \Delta N$ and repeat the "check and increment" process. (Typically, ΔN is chosen propositional to $N_{\max} - N_0$.) It is possible that $s(\boldsymbol{\mu}) - s_{N_{\max}}(\boldsymbol{\mu}) \geq \varepsilon_{\mathrm{des}} \geq \varepsilon_{\mathrm{tol,min}}$ (since Ξ_{train} is not exhaustive); at that point we can either re-assess our requirements, or return to the Offline stage).

4.5 Extension: Multiple Inner Products

We briefly illustrate an effectivity improvement relevant to the small-P case. (The computational and storage cost for higher P is prohibitive.) The improvement is effected through the introduction of multiple inner products [150]: the focus is on the output effectivity; for the output bound, the inner product is a means to an end — and hence can be optimized. We shall also better understand the role of $\overline{\boldsymbol{\mu}}$.

The idea is very simple (we consider here only the absolute output error). We first introduce a sample $\mathcal{V}^{\overline{K}} = \{\overline{\boldsymbol{\mu}}^1 \in \mathcal{D}, \ldots, \overline{\boldsymbol{\mu}}^{\overline{K}} \in \mathcal{D}\}$ of $\overline{K}$ points in $\mathcal{D}$; to each point in $\mathcal{V}^{\overline{K}}$ we then associate an inner product,

$$(w, v)_{X,k} = a(w, v; \overline{\boldsymbol{\mu}}^k), \qquad \forall\, w, v \in X,\ 1 \le k \le \overline{K}\ . \tag{4.67}$$

Our earlier formulation of course corresponds to $\overline{K} = 1$, $\overline{\boldsymbol{\mu}} = \overline{\boldsymbol{\mu}}^1$.

It is then possible, following the development of the preceding sections, to create $\overline{K}$ (possible) error bounds for $s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$,

$$\Delta^s_{N,k}(\boldsymbol{\mu}) \equiv \frac{\|\hat{e}_k(\boldsymbol{\mu})\|^2_{X,k}}{\alpha_{\mathrm{LB},k}(\boldsymbol{\mu})} \qquad 1 \le k \le \overline{K}\ , \tag{4.68}$$

where $\hat{e}_k(\boldsymbol{\mu}) \in X$, $1 \le k \le \overline{K}$, satisfies

$$(\hat{e}_k(\boldsymbol{\mu}), v)_{X,k} = r(v; \boldsymbol{\mu}), \qquad \forall\, v \in X, \tag{4.69}$$

for $r(v; \boldsymbol{\mu})$ defined in (4.22), and

$$\alpha_{\mathrm{LB},k}(\boldsymbol{\mu}) \equiv \Theta^{\min, \overline{\boldsymbol{\mu}}^k}_{a_{\mathrm{S}}}(\boldsymbol{\mu})\ , \tag{4.70}$$

for $\Theta^{\min, \overline{\boldsymbol{\mu}}}_{a_{\mathrm{S}}}$ defined in (4.10).

We then introduce the effectivities

$$\eta^s_{N,k}(\boldsymbol{\mu}) \equiv \frac{\Delta^s_{N,k}(\boldsymbol{\mu})}{s(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})}, \qquad 1 \le k \le \overline{K}\ . \tag{4.71}$$

It directly follows from Proposition 4C that, for $1 \le N \le N_{\max}$,

$$1 \le \eta^s_{N,k}(\boldsymbol{\mu}) \le \theta^{\overline{\boldsymbol{\mu}}^k}(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}\ , \tag{4.72}$$

for $\theta^{\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu})$ defined in (4.15).

We now define our new error bound as

$$\Delta^s_N(\boldsymbol{\mu}) = \Delta^s_{N,k^*(\boldsymbol{\mu})}\ , \tag{4.73}$$

where $k^*\colon\ \mathcal{D} \to \{1,\ldots,\overline{K}\}$ is some indicator function that, given a $\boldsymbol{\mu}$, finds the best candidate error bound amongst the $\overline{K}$ possibilities. There are several possibilities for the indicator strategy k^*.

A first option (O1) is some "explicit rule": some simple partition of $\mathcal{D}$ into $\overline{K}$ subdomains $\mathcal{D}_k$ such that $k^*(\boldsymbol{\mu} \in \mathcal{D}_k) = k$; for example,

$$k^*(\boldsymbol{\mu}) = \arg \min_{k \in \{1,\ldots,\overline{K}\}} |\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}^k|\ , \tag{4.74}$$

where $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^P$. A second option (O2), arguably the best, is to choose

$$k^*(\boldsymbol{\mu}) = \arg \min_{k \in \{1,\ldots,\overline{K}\}} \theta^{\overline{\boldsymbol{\mu}}^k}(\boldsymbol{\mu})\ ; \tag{4.75}$$

in essence, we select the error bound which minimizes the effectivity upper bound (4.28b). A third option (O3), best in sharpness but more Online-expensive, is to choose

$$k^*(\boldsymbol{\mu}) = \arg \min_{k \in \{1,\ldots,\overline{K}\}} \Delta^s_{N,k}(\boldsymbol{\mu})\ ; \tag{4.76}$$

in essence, we first compute all $\overline{K}$ error bounds and then select the error bound which is smallest (and hence sharpest).

The downside to this approach is, of course, cost: the Offline operation count *and* the Online storage scale linearly with $\overline{K}$. The method is thus only really viable for rather modest $\overline{K}$ and hence rather modest P. The good news is that, for modest $\overline{K}$, for O1 and O2 the Online operation count is insensitive to $\overline{K}$: the evaluation of $\|\hat{e}_{k^*(\boldsymbol{\mu})}\|_{X,k^*(\boldsymbol{\mu})}$ dominates the simple optimization (4.74) or (4.75). (For O3 the Online operation count scales linearly with $\overline{K}$, and in this sense O3 is less desirable than O2; O2 arguably offers the best balance between sharpness and cost.)

As a simple example we consider the ThermalBlock problem, Ex1, for $B_1 = 2$, $B_2 = 1$, and hence $P = 1$. We consider $1/\mu_1^{\min} = \sqrt{\mu_{\rm r}}$, $\mu_1^{\max} = \sqrt{\mu_{\rm r}}$ such that $\mu_1^{\max}/\mu_1^{\min} = \mu_{\rm r}$. We recall that, for a single inner product "at" $\overline{\mu}_1 = 1$, $\eta^s_{N,\max} = \mu_{\rm r}^{1/2}$. We now consider a grid $G^{\ln}_{[\mu_1^{\min},\mu_1^{\max};\overline{K}+1]} = [z_1, \ldots, z_{\overline{K}+1}]$; we then set

$$\ln \overline{\mu}_1^k = \frac{1}{2}\left(\ln z_k + \ln z_{k+1}\right), \qquad 1 \leq k \leq \overline{K} \ . \tag{4.77}$$

It is a simple matter to derive

$$\theta^{\overline{\mu}^k}(\mu_1) = \mathrm{Max}\left(\frac{\overline{\mu}_1^k}{\mu_1}, \frac{\mu_1}{\overline{\mu}_1^k}\right) \ , \tag{4.78}$$

and to further conclude (consider $\mu_1 = z_{k+1}$, say)

$$\min_{k \in \{1,\ldots,\overline{K}\}} \max_{\mu_1 \in \mathcal{D}} \theta^{\overline{\mu}^k}(\mu_1) = (\sqrt{\mu_{\rm r}})^{\frac{1}{\overline{K}}} \ . \tag{4.79}$$

It thus follows that, with O2, we obtain

$$\eta^s_{N,\max} \leq (\sqrt{\mu_{\rm r}})^{\frac{1}{\overline{K}}} \ \left(= e^{\frac{1}{2\overline{K}} \ln \mu_{\rm r}}\right) \ . \tag{4.80}$$

Note this reduces to our earlier results for $P = 1$ from Section 4.3.5:

$$\eta^s_{N,\max} \leq \sqrt{\mu_r} \ \text{ for } \ G^{\ln}_{[\mu_1^{\min},\mu_1^{\max};2]} = [\ln \mu_1^{\min}, \ln \mu_1^{\min}] \ ,$$

(and hence $\overline{\mu}_1^1 = 1$ for $\mu_1^{\min} = 1/\sqrt{\mu_r}$, $\mu_1^{\max} = \sqrt{\mu_r}$).

We thus observe that we can (say) control $\eta^s_{N,\max} \leq 10$ if we choose $\overline{K} = [\ln \mu_{\rm r}/2 \ln 10]_+$ (recall $[\quad]_+$ rounds up to the nearest integer). Hence even for $\mu_{\rm r} = 10^6$ — $\mu_1^{\min} = 10^{-3}$ to $\mu_1^{\max} = 10^3$ — we require only $\overline{K} = 6$ inner products: the crucial point is that $\overline{K}$ (at fixed effectivity) scales *logarithmically with* $\mu_{\rm r}$. This result is quite general in fact for both parametrically coercive and coercive problems given the typical parametric coefficient dependence encountered. Unfortunately, to obtain a similar result for $P > 1$ we require roughly

$[\ln \mu_{\rm r}/\ln 10]_+^P$ inner products, and hence — as advertised earlier — the approach is limited to rather small P, or at least to problems with significant ranges in only a few parameter directions.

We make one final point on optimality for the $P = 1$ case. It is clear from the relation (4.78), which is valid for any set of point $\mathcal{V}^{\overline{K}} \equiv \{\overline{\boldsymbol{\mu}}^1, \ldots, \overline{\boldsymbol{\mu}}^{\overline{K}}\}$, that the set of points (4.77) in fact minimizes $\max_{\mu_1 \in \mathcal{D}} \min_{k \in \{1,\ldots,\overline{K}\}} \theta^{\overline{\boldsymbol{\mu}}^k}(\mu_1)$. Thus, for $\overline{K} = 1$, we see that $\overline{\boldsymbol{\mu}}^1 = 1$ (for $\mu_1^{\max} = 1/\mu_1^{\min} = \sqrt{\mu_r}$) is indeed optimal “in the O2 sense.”

Chapter 5

Software Guide

5.1 Introduction

5.1.1 Overview

We introduce in this chapter the MIT-copyright software that we provide. (For description of another RB-related software system, see [121].) This software is intended to serve several functions: first, to facilitate the implementation of the methods described in Chapters 3 and 4 — primarily for "users" of the methodology; and second, to facilitate understanding and extension of the methods described in Chapters 3 and 4 — primarily for "developers" of the methodology. The former need not proceed further than the required MATLAB® scripts, command-line functions calls, and associated INs and OUTs. The latter can choose to delve into — and adapt and modify to their ends — our MATLAB® `.m` codes: in all cases, we provide source code. (Note to distinguish between our problem inputs $\boldsymbol{\mu}$ and outputs $s(\boldsymbol{\mu})$ and the necessary MATLAB® `.m` code arguments and returned quantities we shall refer to the MATLAB variables as "INs" and "OUTs", respectively.)

The software we discuss in this chapter is limited to the abstraction of Part I (i.e., Chapter 2). We shall provide software in subsequent Parts of the book to treat the increasingly general abstractions that we consider. The software for all Parts is very similar as regards both general structure and (user-supplied) INs and OUTs; hence, once the reader has understood the software for (the particularly simple abstraction of) Part I, the reader will have largely mastered — at least from a pragmatic perspective — the software for all the Parts.

The requirements upon the user are fivefold: (*i*) any reasonable laptop or desktop — obviously performance will depend on CPU speed and memory/cache size; (*ii*) MATLAB® Version 6.5 or greater installed (no special toolkits are *needed* for the software of Part I, although the PDE Toolbox® can certainly be *useful*); (*iii*) a thorough understanding of the abstraction of Chapter 2 — for reduction/expression of the user's problem of interest to the requisite form; (*iv*) a good high-level understanding of the numerical methods of Chapters 3 and 4; and (*v*) access to and sufficient familiarity with FE software (and underlying theory) — for development/specification of the "truth" approximation for the user's problem of interest in the necessary "affine" representation (Section 2.1.2). As regards (*iii*), our software will verify compliance and parametric coercivity; however, it is in the interest of the user to confirm these properties before investing further effort. As regards (*iv*), it is sufficient (e.g., for "users") to understand the significance of the numerical quantities (e.g., s_N, N, $\varepsilon_N^{\mathrm{out},*}$, $\Delta_N^s(\boldsymbol{\mu})$): detailed knowledge of the theory and algorithms is not required; we provide pointers back to the necessary definitions as reminders.

As regards (*v*), the INs required by our RB software can be generated by any FE code the user may prefer; often, access to the FE source code may be necessary in order to generate the required affine decomposition, (2.41)–(2.43). We note that FE packages oriented towards, or based upon, domain decomposition for definition of problem geometry and coefficients are particularly well suited to the reduced basis approach. One example of such a package is the MATLAB PDE Toolbox® (`http://www.mathworks.com`): in this case, it is possible to create the finite element inputs to the RB software without modification of, or *even access to*, the source code — the available export features suffice. Another example is COMSOL Multiphysics$^{\mathrm{TM}}$ (`http://www.comsol.com`), which provides scripts for exporting data and structures in MATLAB®.

In Section 5.1.2 we provide instructions for download and installation of the software. In Section 5.2 we briefly recall our ThermalBlock example for the particular case of $B_1 = 3$ and $B_2 = 1$: this problem will serve in Chapter 5 as the "Example" by which we describe the software process. In Section 5.2 we also illustrate the simple "interface" between the FE MATLAB PDE Toolbox® and our own RB software. In Section 5.3 we summarize the three essential Steps for (personal fullfillment and) the development and exercise of RB approximations and *a posteriori* error estimators for a new problem: Step1, Problem Definition; Step2, Offline Stage; and Step3, Online Stage. In Section 5.4 we provide templates for the data entry process for the development of a new problem: in Section 5.4.1 we consider creation of a new problem from "scratch"; in Section 5.4.2 we consider "offline" adaptivity through Greedy restarts. Finally, in Section 5.5 we provide a small reference manual: the contents of the data structures, and the INs and OUTs associated with each of the essential software routines.

User actions are highlighted in blue text for easy identification.

5.1.2 Software Installation

The user should first create a directory named `rbMIT_System` with a subdirectory named `rbMIT_Library` and a subdirectory named `rbMIT_Aux`; see Figure 5.1 for a schematic. (In what follows, we shall refer to directory and folder interchangeably; we shall denote the directory/folder hierarchy in the usual fashion — for example `\rbMIT_System\rbMIT_Library`.)

The user should then download from our website location

`http://augustine.mit.edu/methodology/software/rbMIT_System`

the directories `rbMIT_Library_PartI_V1` and `rbMIT_Aux_PartI_V1` to a temporary location on the user's computer; the user should then move the *contents* of the directory — *not* the directory — `rbMIT_Library_PartI_V1` (respectively, `rbMIT_Aux_PartI_V1`) to `\rbMIT_System \rbMIT_Library` (respectively, `\rbMIT_System\rbMIT_Aux`).

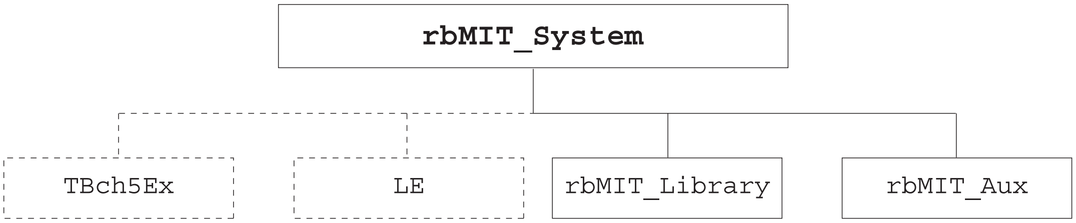

Figure 5.1: Directory structure.

A similar procedure will be followed for (*i*) any updates with corrections/enhancements, `rbMIT_Library_PartI_Vy` and `rbMIT_Aux_PartI_Vy`, and (*ii*) all software additions for the capabilities of the later Parts, `rbMIT_Library_Partx_Vy` and `rbMIT_Aux_Partx_Vy`: in all cases, the user should overwrite any older files with the new files in order for `rbMIT_Library` and `rbMIT_Aux` to remain "current" and "supported"; hence, all user-customized software should be stored in other directories. We will make all attempts to ensure backward-compatibility of all user-specified data and functions.

Software License, Terms and Conditions

By virtue of downloading and installing the Software, the user accepts the following terms and conditions:

Software License

`rbMIT_System` Software Copyright MIT 2006–07

Henceforth, Software shall refer to the `rbMIT_System` software package: the contents of the `rbMIT_Library`, `rbMIT_Aux` folders, and rbMIT_WorkedProblems.

This License governs use of all accompanying Software, and your download and/or use of the Software constitutes acceptance of this License.

You may use this Software for any non-commercial purpose, subject to the restrictions in this License. Some purposes which can be non-commercial are teaching and academic research.

You may not use or distribute this Software or any derivative works in any form for commercial purposes. Examples of commercial purposes would be licensing, leasing, or selling the Software, or distributing the Software for use with commercial products.

You may modify this Software and distribute the modified Software for non-commercial purposes; however, you may not grant rights to the Software or derivative works that are broader than those provided by this License. For example, you may not distribute modifications of the Software under terms that would permit commercial use.

In return, we require that you agree:

- not to remove any copyright or other notices from the Software;
- that if you distribute the Software in source or object form, you will include a verbatim copy of this License;
- that if you distribute derivative works of the Software in source code form you do so only under a license that includes all of the provisions of this License, and if you distribute derivative works of the Software solely in object form you do so only under a license that complies with this License;
- that if you have modified the Software or created derivative works, and distribute such modifications or derivative works, you will cause the modified files to carry prominent notices so that recipients know that they are not receiving the original Software;
- that the Software comes "as is", with no warranties: this means no expressed, implied, or statutory warranty, including without limitation warranties of merchantability or fitness for a particular purpose or any warranty of title or non-infringement; you must pass this disclaimer on whenever you distribute the Software or derivative works;
- that neither MIT nor the authors will be liable for any damages related to the Software or this License, including direct, indirect, special, consequential, or incidental damages;

you must pass this limitation of liability on whenever you distribute the Software or derivative works;

- that your rights under the License end automatically if you breach the License in any way;

- that MIT reserves all rights not expressly granted to you in this License.

(Note: The wording of this license agreement is derived from a Microsoft Shared Source license agreement.)

5.2 An Example

5.2.1 Statement

We consider the ThermalBlock problem (Ex1) introduced in Section 2.2.1 for the particular case $B_1 = 3$, $B_2 = 1$. The case $B_1 = 3$, $B_2 = 1$ is ideal as the example (henceforth "Example") for Chapter 5 by which to illustrate the data entry requirements and processes: sufficiently simple to easily present all the necessary INs and OUTs; yet sufficiently non-degenerate to exercise most all of the important capabilities. (As regards the latter, we choose $B_1 = 3$ (and hence $P = 2$ parameters) rather than $B_1 = 2$ of Section 3.5.2 (and hence $P = 1$ parameters) in order to illustrate the *multi*-parameter case.)

Figure 5.2 depicts the geometry. We recall that for $B_1 = 3$, $B_2 = 1$, our $P + 1 = B_1 B_2 = 3$ subdomain blocks are given by $\Omega^1 =]0, \frac{1}{3}[\times]0, 1[$ (with conductivity μ_1), $\Omega^2 =]\frac{1}{3}, \frac{2}{3}[\times]0, 1[$ (with conductivity μ_2), and $\Omega^3 =]\frac{2}{3}, 1[\times]0, 1[$ (with conductivity unity) such that $\overline{\Omega} = [0, 1]^2 = \overline{\Omega}^1 \cup \overline{\Omega}^2 \cup \overline{\Omega}^3$. The exact space is given by $X^{\mathrm{e}} \equiv \{v \in H^1(\Omega) \,|\, v|_{\Gamma^D} = 0\}$ where $\Gamma^D \equiv \Gamma_{\mathrm{top}}$. We further recall that $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D} = \mathcal{D}_{\mathrm{box}} \equiv [\mu_1^{\min}, \mu_1^{\max}] \times [\mu_2^{\min}, \mu_2^{\max}] \subset \mathbb{R}^{P=2}$; we shall choose $\mu_1^{\min} = \mu_2^{\min} = \mu^{\min} = 0.1$ and $\mu_1^{\max} = \mu_2^{\max} = \mu^{\max} = 10$.

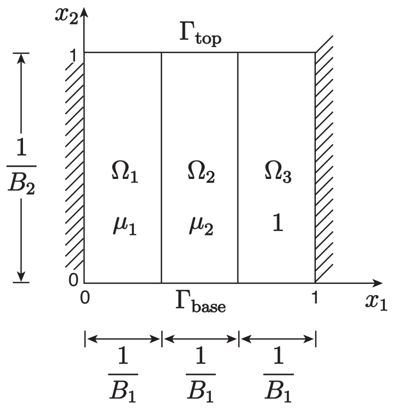

Figure 5.2: Thermal Block for $B_1 = 3$, $B_2 = 1$.

Our parametric bilinear and linear forms a and f are then given by

$$a(w, v; \boldsymbol{\mu}) \equiv \mu_1 \int_{\Omega^1} \nabla w \cdot \nabla v + \mu_2 \int_{\Omega^2} \nabla w \cdot \nabla v + \int_{\Omega^3} \nabla w \cdot \nabla v, \qquad \forall\, w, v \in X^{\mathrm{e}} \text{ , (5.1)}$$

$$f(v) = \int_{\Gamma_{\text{base}}} v, \qquad \forall\, v \in X^{\mathrm{e}} \text{ ,} \tag{5.2}$$

where Γ_{base} is the bottom boundary of Ω. We thus identify the affine representation (2.5), (2.6) for $Q_a = 3$ with $\Theta_a^1 = \mu_1$, $a^1(w, v) = \int_{\Omega^1} \nabla w \cdot \nabla v$, $\Theta_a^2 = \mu_2$, $a^2(w, v) = \int_{\Omega^2} \nabla w \cdot \nabla v$, $\Theta_a^3 = 1$, $a^3(w, v) = \int_{\Omega^3} \nabla w \cdot \nabla v$, and $Q_f = 1$ with $\Theta_f^1 = 1$, $f^1(v) = \int_{\Gamma_{\text{base}}} v$. For our inner product (2.17) we shall choose $\overline{\boldsymbol{\mu}} = (1, 1)$, and hence

$$(w, v)_{X^{\mathrm{e}}} \equiv \int_{\Omega} \nabla w \cdot \nabla v, \qquad \forall\, w, v \in X^{\mathrm{e}} \text{ ;} \tag{5.3}$$

recall from Chapter 4 that our choice of inner product will not affect the accuracy of our RB output prediction but will affect the effectivity of our RB output error bound.

We shall take for our truth approximation the linear FE approximation over the triangulation of Figure 5.3: $X^{\mathcal{N}_{\text{t}}}$ is of dimension $\mathcal{N}_{\text{t}} = 689$. (In fact, for this intentionally very simple model problem, $\mathcal{N}_{\text{t}}$ even for high FE accuracy will be quite small. The example is intended to illustrate the software, not motivate or justify the RB approach.) We note for optimal FE convergence that element boundaries should (and do) coincide with discontinuities in transport coefficients; as we shall observe, this alignment is even more imperative in the RB context — to facilitate the affine decomposition of the FE stiffness matrices.

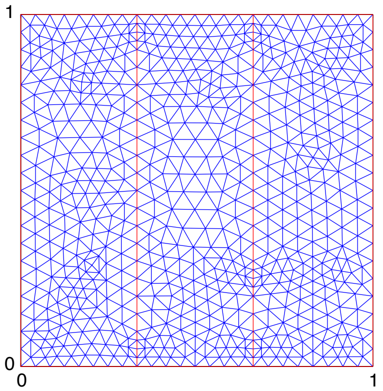

Figure 5.3: The mesh for the `TBCh5Ex` problem: $\dim(X^{\mathcal{N}_\mathrm{t}}) = \mathcal{N}_\mathrm{t} = 689$. The magenta lines denote domain and subdomain boundaries.

We can now define the FE stiffness "submatrices" associated with our affine decomposition (2.16), (2.41), as

$$\mathbb{A}_{i\,j}^{\mathcal{N}_\mathrm{t}\,q} = \int_{\Omega^q} \nabla\varphi_j^{\mathcal{N}_\mathrm{t}} \cdot \nabla\varphi_i^{\mathcal{N}_\mathrm{t}}, \qquad 1 \le i, j \le \mathcal{N}_\mathrm{t},\ 1 \le q \le Q_a\ , \tag{5.4}$$

where the $\{\varphi_i^{\mathcal{N}_\mathrm{t}}\}_{i=1,\ldots,\mathcal{N}_\mathrm{t}}$ are the nodal basis functions associated with $X^{\mathcal{N}}$. It is clear from Figure 5.3 that, for $q = 1, \ldots, Q_a = 3$, $\underline{\mathbb{A}}^{\mathcal{N}_\mathrm{t}\,q}$ contains contributions only from triangles in Ω^q; note by construction each triangle is only associated with a single subdomain. We can also define our FE load/output vector as

$$\mathbb{F}_i^{\mathcal{N}_\mathrm{t}\,1} = \int_{\Gamma_\mathrm{base}} \varphi_i^{\mathcal{N}_\mathrm{t}}, \qquad 1 \le i \le \mathcal{N}_\mathrm{t}\ ; \tag{5.5}$$

recall that $Q_f = 1$.

In actual practice we would typically construct the FE submatrices and vectors by direct stiffness assembly, cycling in turn over the elements of the triangulation restricted to each subdomain. We thus see the important role of domain decomposition in (the FE precursors to) the affine RB framework: in the current example the domain decomposition is naturally associated with heterogeneous coefficients (or "physical properties"); in later Parts, the domain decomposition is naturally induced by the geometric variations.

5.2.2 "Sing Along"

The user may wish to intervalize the instructions in this chapter by "singing along" with the Example. Towards that end, the user should first create a directory `\rbMIT_System\TBch5Ex`. The user should then copy all the contents of `\rbMIT_System\RB_Aux\TBch5Ex_Aux` to `\rbMIT_System\TBch5Ex`.

To sing along, the user should now stay in — and work from — the `\rbMIT_System\TBch5Ex` directory. The user will find that, at each stage of the input process, the necessary scripts and or functions are available in `\rbMIT_System\TBch5Ex`. The user can thus (*i*) read the script/functions to confirm understanding, (*ii*) execute the requisite commands as indicated in Chapter 5, and finally (*iii*) inspect the outputs to verify correct performance.

Before proceeding to Section 5.3, the user must define and load the finite element matrices and vectors `A_FE_1, A_FE_2, A_FE_3`, and `F_FE_1`. There are two options. Either the user can simply load the matrices and vector already provided in our file `TBCh5ExFE`,

```
>> load TBCh5ExFE
```

or the user can proceed to Section 5.2.3 and build these matrices with MATLAB PDE Toolbox®. (Obviously, the latter is an option only if the user has the PDE Toolbox® installed.)

5.2.3 MATLAB PDE Toolbox® Implementation

For the current example, we can enlist the MATLAB PDE ToolBox® to build `A_FE_1` $= \underline{\mathbb{A}}^{\mathcal{N}_t 1}$, A_FE_2 $= \underline{\mathbb{A}}^{\mathcal{N}_t 2}$, A_FE_3 $= \underline{\mathbb{A}}^{\mathcal{N}_t 3}$, and `F_FE_1` $= \underline{\mathbb{F}}^{\mathcal{N}_t 1}$. As MATLAB® variables, `A_FE_1, A_FE_2`, and `A_FE_3` are each sparse $\mathcal{N}_t \times \mathcal{N}_t$ arrays, and `F_FE_1` is a sparse $\mathcal{N}_t \times 1$ array (vector); recall the importance of recognizing and exploiting sparsity in the truth operators.

For packages such as MATLAB PDE Toolbox® that are cognizant of the domain decomposition underlying the property (and more generally, geometry) definition, it is often a simple matter to extract the affine decomposition matrices and vectors required by the RB method.

For example, to obtain the matrix `A_FE_1` we invoke the standard tools to form the "full" stiffness matrix but with Θ_a^1 artificially set to unity and $\Theta_a^q, 2 \leq q \leq Q_a$, artificially set to zero; applying this "trick" in turn to each subdomain we thus create, in Q_a "full" stiffness matrix formations, all the necessary `A_FE_q`, $1 \leq q \leq Q_a$. A similar procedure applies to the load (in this case, just `F_FE_1`). Often, no access to FE source is required.

The particular commands required to effect this procedure are readily available in the GUI of the MATLAB PDE Toolbox® package. First, the geometry, mesh, boundary conditions, and PDE coefficients are specified (our script now in `\rbMIT_System\TBCh5Ex\TBCh5Ex_PDE_Tool.m`); next, the quantities are exported to the MATLAB workspace; finally, the M-functions such as `assempde, assema, assemb` (our script now in `\rbMIT_System\TBCh5Ex\TBCh5Ex_assemble.m`) are invoked to assemble the necessary matrices and vectors. (For a quick guide to the MATLAB PDE Toolbox®, see [1].)

We now explicity indicate the steps for our particular example.

The user should first call

```
>>TBCh5Ex_PDE_Tool
```

at the MATLAB command-line level. This script opens the MATLAB PDE Toolbox® GUI and specifies the geometry, mesh, and PDE structure, coefficients, and boundary conditions. (At this level the user can access all the capabilities provided inside the Toolbox.) Next, from the `Boundary Menu` and `Mesh Menu` select

`Export Decomposed Geometry, Boundary Cond's` and `Export Mesh`,

respectively. For each operation a confirmation – clicking on the button "OK" – is needed. (Note when closing the ToolBox window, the user can indicate "No" to saving any changes.)

Once the user has successfully exported the geometry, boundary conditions, and mesh into the MATLAB® workspace, the user should run the script

```
>>TBCh5Ex_PDE_assemble
```

which creates the file `TBCh5ExFE_abinitio` with `A_FE_1`, `A_FE_2`, `A_FE_3`, and `F_FE_1`.

The user should then load

```
>>TBCh5ExFE_abinitio
```

to bring `A_FE_1`, `A_FE_2`, `A_FE_3`, and `F_FE_1` into the MATLAB® workspace.

5.3 Problem Creation: Summary

The user should first give the new problem a unique name, `*PROBNAME`. We emphasize that `*PROBNAME` is symbolic for — *to be replaced/read everywhere it occurs*— as the actual problem name (e.g., for our Example, `*PROBNAME` ⇒ `TBCh5Ex)`. Note that except for `*PROBNAME` all other (parts of) the MATLAB® `.mat` and `.m` file specifications are universal for all problems.

The user should then create the directory `\rbMIT_System\*PROBNAME`: this directory will contain the necessary data and functions required to define the problem and subsequently construct and evaluate the RB approximations and associated *a posteriori* error estimators. We indicate the directory structure in Figure 5.1 for the case of two problems: `*PROBNAME` ⇒ `TBCh5Ex` (the directory for which has already been created in Section 5.2.2), and say `*PROBNAME` ⇒ `LE` (for Linear Elasticity).

We now describe the development and exercise of RB approximations and *a posteriori* error estimators for a new problem. There are three steps: Step1, Problem Definition; Step2, Offline Stage; and Step3, Online Stage. The first two Steps are performed once (for each new problem); the last Step is of course executed many times.

- In Step1, `\*PROBNAME\*PROBNAME_PROBDEF.mat` is created by the user; a function related to the definition of the parameter domain, `\*PROBNAME\*PROBNAME_InsideOutsideD.m`, and a function related to the parametric dependence of the bilinear and linear forms, `\*PROBNAME\*PROBNAME_Get_Theta_q.m`, must also be specified by the user.

- In Step2, *PROBNAME*PROBNAME_OFFLINE.mat and *PROBNAME*PROBNAME_ONLINE.mat are created by the user and our function \rbMIT_Library\Greedy_parcoer_compliant.m; the latter is an implementation of the Greedy$^{\text{out}}$ algorithm of Section 3.4.4.

  (Note *PROBNAME*PROBNAME_OFFLINE.mat contains data that can serve both (*i*) to refine a RB approximation, as described in Section 5.4.2, and (*ii*) to pursue more advanced "collateral" activities such as visualization.)

- In Step3, the real-time output and error bound are computed by a *PROBNAME-specific instantiation of our "RB Online Evaluator" function \rbMIT_Library\Online_parcoer_compliant.m; the latter is an implementation of the Online procedures of Sections 3.3 and 4.4.

We present in Figure 5.4 an overview of the key software and data ingredients in the problem creation process.

In Section 5.4 we provide problem creation templates for the data entry process: the detailed sequence of definitions and commands that must be executed to develop and execute a new problem. In Section 5.5 we supplement the templates with a small reference manual: we describe (*i*) the contents of the datafiles *PROBNAME_PROBDEF.mat, *PROBNAME_OFFLINE.mat, and *PROBNAME_ONLINE.mat, (*ii*) the requisite INs and OUTs for the (user-supplied) functions *PROBNAME_InsideOutsideD.m and *PROBNAME_Get_Theta_q.m, and (*iii*) the INs and OUTs, and Error Messages/Diagnostics, for the (rbMIT_Library-provided) functions Greedy_parcoer_compliant.m and Online_parcoer_compliant.m.

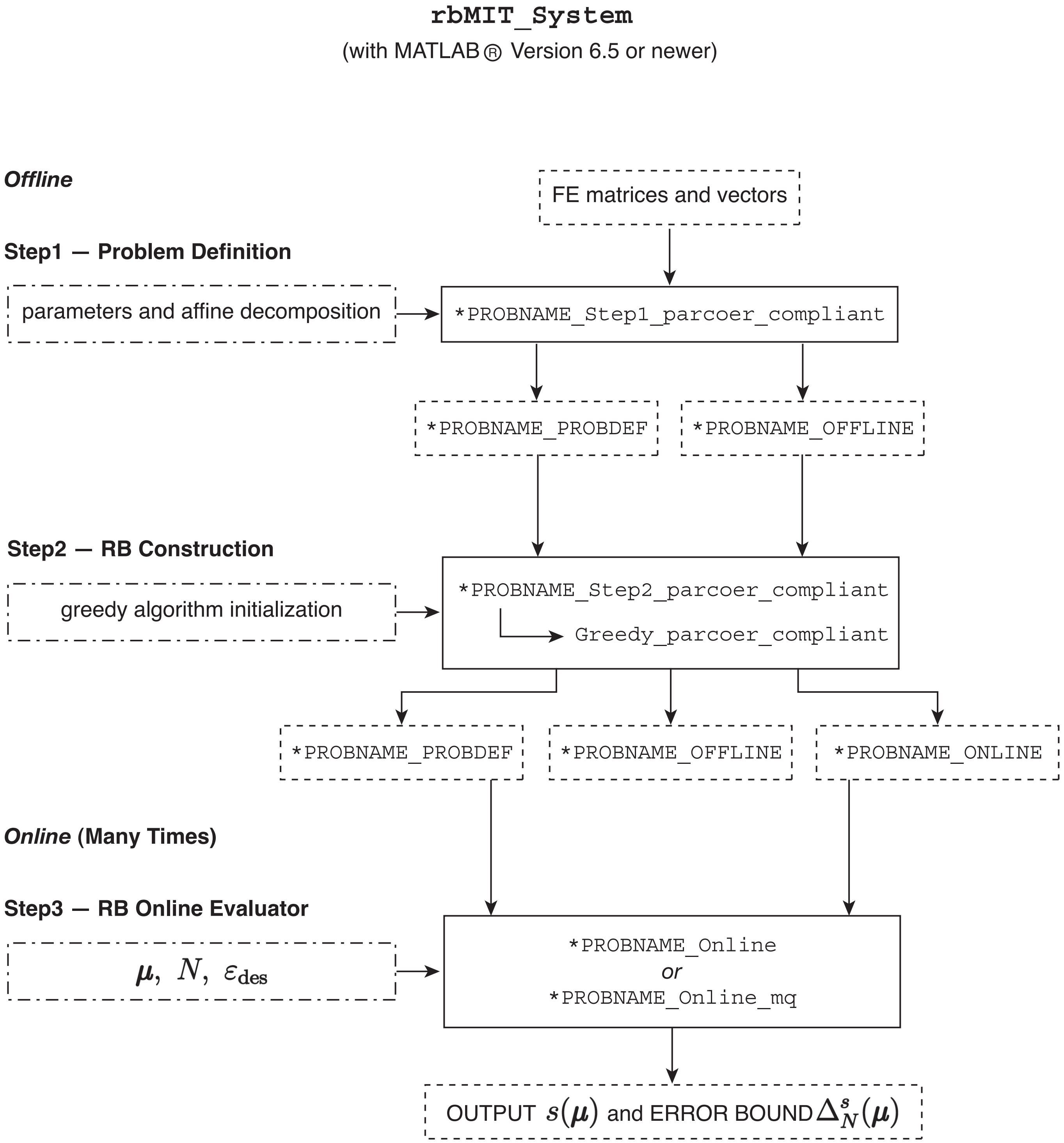
rbMIT_System
(with MATLAB® Version 6.5 or newer)
Offline
Step1 — Problem Definition
FE matrices and vectors
parameters and affine decomposition
*PROBNAME_Step1_parcoer_compliant
*PROBNAME_PROBDEF
*PROBNAME_OFFLINE
Step2 — RB Construction
greedy algorithm initialization
*PROBNAME_Step2_parcoer_compliant
Greedy_parcoer_compliant
*PROBNAME_PROBDEF
*PROBNAME_OFFLINE
*PROBNAME_ONLINE
Online (Many Times)
Step3 — RB Online Evaluator
$\boldsymbol{\mu}, N, \varepsilon_{\text{des}}$
*PROBNAME_Online
or
*PROBNAME_Online_mq
OUTPUT $s(\boldsymbol{\mu})$ and ERROR BOUND $\Delta_N^s(\boldsymbol{\mu})$

Figure 5.4: rbMIT_System Flow Chart.

5.4 Problem Creation: Templates

There are two cases to consider: creation of a new problem from scratch (Section 5.4.1); and modification of an existing problem (Section 5.4.2).

In some cases the user must directly enter command-line instructions; in such cases (as in Section 5.2.3), we precede these user inputs with the MATLAB® prompt `>>`. However, most of the data entry is through templates-cum-MATLAB® scripts; in particular, there is a script for Step1 and a script for Step2. As our templates are actual MATLAB® `.m` scripts (to be executed), we perforce adopt the usual MATLAB® notational conventions: any text not preceded by the comment indications `%` or enclosed by the block comment delimiters `%{` and `%}` will be evaluated. We shall denote by `?` the problem-specific numerical inputs to be provided by the user; for each requested quantity `?` we shall include comment that provides the general definition. In a very few cases, we shall write for part of the user input — not the actual data `?` but rather (typically) a subscript or superscript limit — `VALUE(Arg)`: the user should replace `VALUE(Arg)` with the numerical value of the symbolic variable `Arg` for the user's particular problem `*PROBNAME`. Hence in the scripts the user should search for (predominantly) `?` but also (a few) `VALUE` expressions.

In Step1 (and to a lesser extent Step3), the user must also invoke the editor to create several user-supplied functions in `\*PROBNAME`. We provide templates for these functions.

We caution that the user should proceed in sequence from Step1 to Step2 to Step3: backing up can lead to severe tire damage. Thus if at any Step an error arises we recommend that the user delete `*PROBNAME_PROBDEF.mat`, `*PROBNAME_OFFLINE.mat`, and `*PROBNAME_ONLINE.mat` and recommence with Step1. As our scripts perform most of the typing for the user, data entry is relatively painless; and in the case of an error, correction is particularly simple — only the offending data/line of the script need be rectified.

5.4.1 “Tabula Rasa”: Creation of a New Problem from Scratch

Step1, Problem Definition

Initialize. First, in the directory `rbMIT_System\*PROBNAME`, the user should create a new file `*PROBNAME_Step1_parcoer_compliant.m`. Second, the user should copy the contents of the script `\rbMIT_Aux\Step1_parcoer_compliant.m` to this new file `\rbMIT_System\*PROBNAME\*PROBNAME_Step1_parcoer_compliant.m`. Third, in `\rbMIT_System\*PROBNAME\*PROBNAME_Step1_parcoer_compliant.m` the user should replace all occurrences of the string `USERPROB` with the actual name of the new problem `*PROBNAME` (*not* the string ‘`*PROBNAME`’!): a simple initial global find/replace and save. Finally, the user should then set the directory to `\rbMIT_System\*PROBNAME` for all of Step1.

Note for the Example the user has already created `\rbMIT_System\TBCh5Ex` and furthermore `\rbMIT_System\TBCh5Ex\TBCh5Ex_Step1_parcoer_compliant` should already exist; see Section 5.2.2.

Edit and Execute* Step1 *Script. We include the initialized Step1 script `*PROBNAME_Step1_parcoer_compliant` here for easy reference. (Note the dummy `USERPROB` in the master Step1 script of `\rbMIT_Aux` has already been replaced with the actual name of the user’s new problem, `*PROBNAME`, in the `Step1` script of `\*PROBNAME` listed below.)

`-----BEGIN: *PROBNAME_Step1_parcoer_compliant.m-----`

```
% Script *PROBNAME_Step1_parcoer.m: Copyright MIT 2007.
% Fill PROBDEF structure.
*PROBNAME_PROBDEF.P = ? % scalar $P$, the number of parameters
*PROBNAME_PROBDEF.mu_min = ? % $1\times P$ vector $[\mu^\min_1,\mu^\min_2,\ldots,\mu^\min_P]$ that defines lower limit of ${\cal D}_{box}{$
*PROBNAME_PROBDEF.mu_max = ? % $1\times P$ vector $[\mu^\max_1,\mu^\max_2,\ldots,\mu^\min_P]$ that defines upper limit of ${\cal D}_{box}$
*PROBNAME_PROBDEF.mu_bar = ? % $1\times P$ vector $\overline{\bfmu} \in {\cal D}$ which defines inner product/norm
*PROBNAME_PROBDEF.Q_affine.a = ? % scalar $Q_a$
*PROBNAME_PROBDEF.Q_affine.f = ? % scalar $Q_f$
save *PROBNAME_PROBDEF *PROBNAME_PROBDEF; % save *PROBNAME_PROBDEF structure to file *PROBNAME_PROBDEF.mat

% First enter the FEM matrices.
*PROBNAME_OFFLINE.FEM.matrix.Aq{1} = ?; % ${cal N}_t \times {\cal N}_t$ sparse array $\underline{\mathbb{A}}^{{\cal N}_t\,1}$
*PROBNAME_OFFLINE.FEM.matrix.Aq{2} = ?; % ${cal N}_t \times {\cal N}_t$ sparse array $\underline{\mathbb{A}}^{{\cal N}_t\,2}$
...the user
...should cut and paste
...the requisite number of lines
*PROBNAME_OFFLINE.FEM.matrix.Aq{VALUE(Q_a)} = ?; % ${cal N}_t \times {\cal N}_t$ sparse array $\underline{\mathbb{A}}^{{\cal N}_t\,{Q_a}}$
% Note the FE equations are solved by Cholesky decomposition and forward-/back-substitution;
% the equations and unknowns are first re-ordered by the MATLABR routine symamd to minimize fill-in
% during the Cholesky process.
%
% Next enter the FEM vectors.
*PROBNAME_OFFLINE.FEM.matrix.Fq{1} = ?; % ${cal N}_t \times 1$ sparse array $\underline{\mathbb{F}}^{{\cal N}_t\,1}$
```

```
*PROBNAME_OFFLINE.FEM.matrix.Fq{2} = ?; % ${cal N}_t \times 1$ sparse array $\underline{\mathbb{F}}^{{\cal N}_t\,2}$
...the user
...should cut and paste
...the requisite number of lines
*PROBNAME_OFFLINE.FEM.matrix.Fq{VALUE(Q_f)} = ?; %${cal N}_t \times 1$ sparse array $\underline{\mathbb{F}}^{{\cal N}_t\,{Q_f}}$
save *PROBNAME_OFFLINE *PROBNAME_OFFLINE; % save *PROBNAME_OFFLINE structure (so far) to file *PROBNAME_OFFLINE.mat
```

-----END: *PROBNAME_Step1_parcoer_compliant.m-----

The user should next edit the script \rbMIT_System*PROBNAME*PROBNAME_Step1_parcoer_compliant.m given above: replace the ? (and occasional VALUE(Arg)) with the data associated with the new problem *PROBNAME. The user should then execute the resulting script

```
>>*PROBNAME_Step1_parcoer_compliant
```

from the *PROBNAME (\TBCh5Ex for the Example) directory.

Define User-Supplied Functions. The user should now create the two functions *PROBNAME*PROBNAME_InsideOutsideD.m and *PROBNAME*PROBNAME_Get_Theta_q.m. We include templates USERPROB_InsideOutsideD.m and USERPROB_Get_Theta_q.m in \rbMIT_Aux: the user should copy the contents of these files to *PROBNAME*PROBNAME_InsideOutsideD.m and *PROBNAME*PROBNAME_Get_Theta_q.m, respectively; the user should then replace *all occurrences* of the dummy USERPROB label with *PROBNAME in both files. Finally the user should then modify the "logic" to match the requirements of *PROBNAME. (Recall that for the Example the necessary functions should already exist in rbMIT_System\TBCh5Ex.)

Description of the function*PROBNAME*PROBNAME_InsideOutsideD.m.

The specifications for the function

```
[InsideBoolean] = *PROBNAME_InsideOutsideD(*PROBNAME_PROBDEF, muvectorvalue)
```

are

(i) INs:

(a) *PROBNAME_PROBDEF: A structure — *PROBNAME_PROBDEF; the *PROBNAME_PROBDEF.P scalar and `*PROBNAME_PROBDEF.mu_min` and `*PROBNAME_PROBDEF.mu_max` vectors might prove helpful to the user. (For simplicity, in the function definition, the argument `*PROBNAME_PROBDEF` can of course be replaced with a dummy argument such as `PROBDEF` or `DEF`.)

(b) `muvectorvalue`: An $M \times P$ array — a set of parameter vectors $\boldsymbol{\mu}_m \in \mathbb{R}^P, 1 \leq m \leq M$, where each $\boldsymbol{\mu}_m$ is a $1 \times P$ array.

(ii) OUT:

$$\texttt{InsideBoolean} = \begin{cases} 0 \text{ if } \boldsymbol{\mu}_m \; \notin \mathcal{D} \text{ for } \underline{\textit{any}}\ m \text{ in } \{1, \ldots, M\} \\ 1 \text{ if } \boldsymbol{\mu}_m \; \in \mathcal{D} \text{ for } \underline{\textit{all}}\ m \text{ in } \{1, \ldots, M\} \end{cases} .$$

In essence, `*PROBNAME_InsideOutsideD` is the characteristic function associated with $\mathcal{D}$: it is called by our `\rbMIT_Library` functions as `[InsideBoolean] = *PROBNAME_InsideOutsideD` (`PROBNAME_PROBDEF`, muvectorvalue).

For the template provided (shown below already inizialized to `*PROBNAME`), we consider $\mathcal{D} \equiv \mathcal{D}_{\text{box}} \equiv [\mu_1^{\min}, \mu_1^{\max}] \times \ldots \times [\mu_P^{\min}, \mu_P^{\max}]$:

`----BEGIN \RB_Aux\*PROBNAME_InsideOutsideD.m-----`

```
function [InsideBoolean] = *PROBNAME_InsideOutsideD(PROBDEF, muvectorvalue);
InsideBoolean = 1;   % set inside Boolean to 1
nummupts=size(muvectorvalue,1); % nummupts=M
for p = 1:PROBDEF.P; % to test each parameter component
 for m= 1:nummupts   % to test all parameter vectors
    if(muvectorvalue(m, p) < PROBDEF.mu_min(p))
        InsideBoolean = 0; % Outside D_box from below
        return;
    end;
    if(muvectorvalue(m, p) > PROBDEF.mu_max(p))
        InsideBoolean = 0; % Outside D_box from above
        return;
    end;
 end;
end;
```

`----END \RB_Aux\*PROBNAME_InsideOutsideD.m----`

We emphasize that the above directly applies to *any* problem for which $\mathcal{D} \equiv \mathcal{D}_{\text{box}}$ (including our example, `TBCh5Ex`); however, if $\mathcal{D} \not\equiv \mathcal{D}_{\text{box}}$ the user must substitute the more complex logic associated with the particular parameter domain of interest.

Description of the function*PROBNAME*PROBNAME_Get_Theta_q.m.

The specifications for the function

`[Thetavectorvalue] = *PROBNAME_Get_Theta_q (*PROBNAME_PROBDEF, muvectorvalue)`

are

(*i*) INs:

(*a*) `*PROBNAME_PROBDEF`: A structure — `*PROBNAME_PROBDEF`; the `*PROBNAME_PROBDEF.Q_affine.a` and `*PROBNAME_PROBDEF.Q_affine.f` scalars might prove helpful to the user. (For simplicity, in the function definition, the argument `*PROBNAME_PROBDEF` can of course be replaced with a dummy argument such as `PROBDEF` or `DEF`.)

(*b*) muvectorvalue: An $M \times P$ array — a set of parameter vectors $\boldsymbol{\mu}_m \in \mathbb{R}^P, 1 \leq m \leq M$, where each $\boldsymbol{\mu}_m$ is a $1 \times P$ array.

(*ii*) OUT:

`Thetavectorvalue`: An $M \times (Q_a + Q_f)$ array — a set of coefficient vectors $[\Theta_a^1(\boldsymbol{\mu}_m)\ \Theta_a^2(\boldsymbol{\mu}_m) \cdots \Theta_a^{Q_a}(\boldsymbol{\mu}_m)\ \Theta_f^1(\boldsymbol{\mu}_m)\ \Theta_f^2(\boldsymbol{\mu}_m) \cdots \Theta_f^{Q_f}(\boldsymbol{\mu}_m)]$, $1 \leq m \leq M$, where each coefficient vector is a $1 \times (Q_a + Q_f)$ array.

In essence, `*PROBNAME_Get_Theta_q` encapsulates the parameter dependence of the bilinear and linear form for the problem `*PROBNAME`.

For the template (shown below already initialized to `*PROBNAME`) we provide:

----BEGIN *PROBNAME*PROBNAME_Get_Theta_q.m----

```
function [Thetavectorvalue] = *PROBNAME_Get_Theta_q(DEF, muvectorvalue);

nummupts=size(muvectorvalue,1); % nummupts=M
for m=1:nummupts

% First compute the Theta_a^q, 1 \le q \le Q_a.
Thetavectorvalue(m,1) =  ? ;
% for example, if Theta_a^1=\mu_2*\mu_3, then ? is muvectorvalue(m,2)*muvectorvalue(m,3);
% ...then the user should cut and paste the requisite number of line
Thetavectorvalue(m,VALUE(Q_a)) = ?;

% Now compute the Theta_f^q, 1 \le q \le Q_f.
Thetavectorvalue(m,VALUE(Q_a+1)) = ?;
```

```
% for example, if Theta_f^1=1/\mu_4, ? is 1./muvectorvalue(m,4)
% ...then the user should cut and paste  the requisite number of line
Thetavectorvalue(m,VALUE(Q_a+Q_f)) = ?;
% this corresponds to Theta_f^{Q_f}


end
```

----END *PROBNAME*PROBNAME_Get_Theta_q.m----

We note that the user should remember to use ";" to suppress printing in *PROBNAME_InsideOutsideD and in particular in *PROBNAME_Get_Theta_q, as otherwise during the Greedy process there will be (much) unnecessary data sent to the screen.

Step2, Offline Stage

Initialize. The user should first copy the script \rbMIT_System\rbMIT_Aux\Step2_parcoer_compliant.m to the file \rbMIT_System*PROBNAME*PROBNAME_Step2_parcoer_compliant.m. Then, in \rbMIT_System\PROBNAME\PROBNAME_Step2_parcoer_compliant.m the user should replace all occurrences of USERPROB with the actual name of the new problem, *PROBNAME — a simple initial global find/replace and save. The user should then set the directory to \rbMIT_System*PROBNAME (\rbMIT_System\TBCh5Ex for the Example) for all of Step2.

Note for the Example the user has already created \rbMIT_System\TBCh5Ex and furthermore \rbMIT_System\TBCh5Ex\TBCh5Ex_Step2_parcoer_compliant should already exist; see Section 5.2.2.

***Edit and Execute* Step2 *Script*.** We include the initialized Step2 script *PROBNAME_Step2_parcoer_compliant here for easy reference. (Note the dummy USERPROB in the master Step2 script of \rbMIT_Aux has already been replaced with the actual name of the user's new problem, *PROBNAME, in the Step2 script of *PROBNAME listed below.)

----BEGIN: *PROBNAME_Step2_parcoer_compliant.m----

```
% Script *PROBNAME_Step2_parcoer_compliant.m: Copyright MIT 2007.
% Enter control parameters for Greedy^{out,*} generation of space.
load *PROBNAME_OFFLINE
% First address generation of the sample $\Xi_{train}$.
*PROBNAME_OFFLINE.space.sample.newflg = ?
%  1 $\Rightarrow$ create new sample $\Xi_{train}$; 0 $\Rightarrow$ use existing sample (previously created) $Xi_{train}$
*PROBNAME_OFFLINE.space.sample.densityflg = ?
% 0 $\Rightarrow$ random with uniform density in $\bfmu$; 1 $\Rightarrow$ random with uniform density in $\ln(\bfmu)$ (requires $\mu_{\min} >0$)
```

```
*PROBNAME_OFFLINE.space.sample.size = ? % positive integer $n_{train}$ --- size of $\Xi_{train}$
% Next provide weighting, tolerance,and limits.
*PROBNAME_OFFLINE.space.absrelflg = ? % 0 $\Rightarrow \omega_N = 1$ (absolute output error); 1 $\Rightarrow \omega_N = s_N$ (relative output error)
*PROBNAME_OFFLINE.space.tol = ? % positive real $\varepsilon_{tol,\min}$ --- anticipated smallest desired (absolute or relative) output error
*PROBNAME_OFFLINE.space.Nbarmax = ? % positive integer $\overline{N}_{\max}$ --- upper limit on dimension of (largest) RB space
*PROBNAME_OFFLINE.space.restartflg= 0 % 0 normal running, 1 only if a re-start
% beyond this line no more user input required
save *PROBNAME_OFFLINE *PROBNAME_OFFLINE; % save final *PROBNAME_OFFLINE structure inputs to file *PROBNAME_OFFLINE.mat

% Create ONLINE structure (to which Greedy^{out,*} will supply quantities required for the Online stage).
if (*PROBNAME_OFFLINE.space.restartflg == 0)
*PROBNAME_ONLINE.space = [ ]; % no problem-specific (user) inputs required
else
load *PROBNAME_ONLINE
end
save *PROBNAME_ONLINE *PROBNAME_ONLINE; % save (currently empty) *PROBNAME_ONLINE structure to file *PROBNAME_ONLINE.mat

% Load the problem data required by the Greedy algorithm.
load *PROBNAME_PROBDEF;
load *PROBNAME_OFFLINE;
load *PROBNAME_ONLINE;
% Note the ‘‘load" is a good precaution --- and a necessity if the new problem creation spans several sessions in any case ---
% to ensure complete and correct data in the workspace.

% Call the Greedy routine (which resides in \rbMIT_System\rbMIT_Library).

addpath('../rbMIT_Library')


[*PROBNAME_PROBDEF, *PROBNAME_OFFLINE, *PROBNAME_ONLINE] = Greedy_parcoer_compliant(...
...*PROBNAME_PROBDEF, *PROBNAME_OFFLINE, *PROBNAME_ONLINE, @*PROBNAME_InsideOutsideD,  @*PROBNAME_Get_Theta_q);

% Display the ‘‘user--readable" outputs of the Greedy algorithm. (Note the Greedy code produces many ‘‘non--user--readable" OUTs
% (saved to the OFFLINE and ONLINE structures and  corresponding .mat files) that are needed for (i) ‘‘Inherited" problem
% creation (see Section 5.4.2), and of course (ii) the Online RB output and error bound evaluation s--- Step3 below.
save *PROBNAME_PROBDEF *PROBNAME_PROBDEF;
save *PROBNAME_OFFLINE *PROBNAME_OFFLINE;
save *PROBNAME_ONLINE *PROBNAME_ONLINE;

*PROBNAME_ONLINE.space.Nmax  % scalar --- the value of $N_{\max}$
*PROBNAME_ONLINE.space.eps_out_star %  $1 \times N_{\max}$ vector --- $\varepsilon^{out,\ast}(N), 1 \le N \le N_{\max}$

semilogy(1:size(*PROBNAME_ONLINE.space.eps_out_star,2), *PROBNAME_ONLINE.space.eps_out_star, 'o')
title('Offline Adaptive Sampling')
xlabel('N')
ylabel('\epsilon^{out,*}')
% Note if the Greedy algorithm terminates ‘‘normally" then either *PROBNAME_ONLINE.space.Nmax = Nbarmax or
% *PROBNAME_ONLINE.space.eps_out_star(Nmax) $\le \varepsilon_{tol,\min}$.
% (We consider non-normal termination and associated error messages and remedies in Sections 5.4.2 and 5.5.)
%
```

----END *PROBNAME_Step2_parcoer_compliant.m----

The user should then edit the script rbMIT_System*PROBNAME*PROBNAME_Step2_parcoer_compliant.m — replace the ? with the data associated with the new problem *PROBNAME — and then execute the resulting script

```
>> *PROBNAME_Step2_parcoer_compliant
```

from the \rbMIT_System*PROBNAME directory.

The user should be aware that the script *PROBNAME_Step2_parcoer_compliant fixes also the path to let MATLAB find functions in the directory \rbMIT_System\rbMIT_Library while working in the directory rbMIT_System*PROBNAME. If you have problems or error messages check the path in *PROBNAME_Step2_parcoer_compliant to be sure your operating system is recognizing the correct root. (LINUX machines use “/”, WINDOWS machines generally use

"\" or both.)

Step3, Online Stage

Initialize. The user should set the directory to \rbMIT_System*PROBNAME (\rbMIT_System \TBCh5Ex for the Example). The user should also

```
>>addpath('../rbMIT_Library')
```

in order to access the RB Online Evaluator. (In fact, the more experienced user can run the RB Online Evaluator from any directory: the user need only establish paths not only to \rbMIT_System\RB_Library but also to \rbMIT_System*PROBNAME.)

The user should copy the script \rbMIT_System\rbMIT_Aux\USERPROB_Online.m to the file \rbMIT_System*PROBNAME*PROBNAME_Online.m; then, in \rbMIT_System*PROBNAME*PROBNAME_Online.m the user should replace all occurrences of USERPROB with the actual name of the new problem, *PROBNAME: a simple initial global find/replace and save. Note *PROBNAME_Online is a *PROBNAME-specific "shorthand" that invokes the general RB Online Evaluator.

The user should then also copy the script \rbMIT_System\rbMIT_Aux\USERPROB_Online_mq.m to the file \rbMIT_System*PROBNAME*PROBNAME_Online_mq.m; then, in \rbMIT_System*PROBNAME*PROBNAME_Online_mq.m the user should replace all occurrences of USERPROB with the actual name of the new problem, *PROBNAME. Note *PROBNAME_Online_mq.m is a problem-specific shorthand for the multi-query version of the RB Online Evaluator.

Note for the Example, the functions TBCh5Ex_Online and TBCh5Ex_Online_mq should already exist in the \rbMIT_System\TBCh5Ex directory.

At the beginning of any session, the user should load the necessary data after declaring the variables as global (so that the Online Evaluator can be called more succintly):

```
>>clear *PROBNAME_PROBDEF;

>>clear *PROBNAME_ONLINE;

>>global *PROBNAME_PROBDEF;
```

```
>>global *PROBNAME_ONLINE;

>>load *PROBNAME_PROBDEF;

>>load *PROBNAME_ONLINE;
```

Note once the data from these structures is in the MATLAB workspace (and as long as the data is not subsequently deleted or compromised) there is no need to reload the data before each Online evaluation; indeed, unnecessary reloading will greatly degrade the Online performance.

Evaluate the RB Output and Error Bound. The user (from the `\rbMIT_System\*PROBNAME` directory) should then call the RB Online Evaluator

```
>>[sN, DeltaN] = *PROBNAME_Online (muvectorvalue, N, epsdes)
```

to obtain true output and error bound.

We indicate here INs and OUTs.

(*i*) INs

- (*a*) `muvectorvalue`: A $1 \times P$ real vector — the $\boldsymbol{\mu} \in \mathcal{D}$ of interest.
- (*b*) `N`: A non-negative integer — the dimension N of the RB approximation space.
- (*c*) `epsdes`: A non-negative real scalar — ε_{des}, the desired (maximum acceptable) output error, $s^{\mathcal{N}_t}(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$.

(*ii*) OUTs:

- (*a*) `sN`: A scalar — the RB output prediction, $s_N(\boldsymbol{\mu})$.
- (*b*) `DeltaN`: A scalar — the RB *a posteriori* output error bound, $\Delta^{\cdot}_N(\boldsymbol{\mu})$; the absolute (.   `= s`) or relative (.   `= s,rel`) error is reported depending on the user specification of `*PROBNAME_OFFLINE.space.absrelflg` (also automatically stored in `*PROBNAME_ONLINE.space.absrelflg`).

As described in greater detail in Section 4.4.3, the two inputs `N` and `epsdes` can serve in two different "modes": if `N > 0`, and `epsdes` is set to zero, then `*PROBNAME_Online` returns

$s_N(\boldsymbol{\mu})$ and $\Delta^{\cdot}_N(\boldsymbol{\mu})$. If `N =0`, then `*PROBNAME_Online` returns $s_{N^*}(\boldsymbol{\mu})$ and $\Delta^{\cdot}_{N^*}(\boldsymbol{\mu})$ where N^* is the smallest N' such that $\Delta^{\cdot}_{N'} \leq$ `epsdes` (in actual fact, we settle for an efficiently calculated slightly sub-optimal result, per Section 4.4.3).

Multiple Queries. To extract the output for a range of parameters we provide also a multi-query RB Online Evaluator. In particular the routine below permits rapid presentation of $s_N(\boldsymbol{\mu})$ for variation of "one parameter component at a time." Initialization proceeds as indicated above. The user should then call

```
>>*PROBNAME_Online_mq (mu_index, mu_min, mu_max, muvectorvalue, N, epsdes)
```

to obtain the desired plot.

We indicate the INs and OUTs.

(i) INs:

(a) `mu_index`: Integer — index $i \in \{1, \ldots, P\}$ of the parameter component we wish to vary.

(b) `mu_minplt`: Scalar — minimum plotted value for the parameter to be varied; $\mu^{\min}_{\texttt{mu\_index}} \leq$ `mu_minplt`
$\leq \mu^{\max}_{\texttt{mu\_index}}$.

(c) `mu_maxplt`: Scalar — maximum plotted value for the parameter to be varied; $\mu^{\min}_{\texttt{mu\_index}} \leq \texttt{mu\_maxplt} \leq \mu^{\max}_{\texttt{mu\_index}}$.

(d) `muvectorvalue`: A $1 \times P$ real vector — the value of $\boldsymbol{\mu} \in \mathcal{D}$ of all the fixed parameter component μ_j, $1 \leq j \leq P$, $j \neq \texttt{mu\_index}$; the value of the varying component $\mu_{\texttt{mu\_index}}$ should be set to unity.

(e) `N`: A non-negative integer — the dimension N of the RB approximation space.

(f) `epsdes`: A positive scalar — ε_{des}, the desired (maximum acceptable) output error, $s^{\mathcal{N}_t}(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu})$.

Note `N` and `epsdes` play the same role as in the single query case.

(*ii*) OUTs:

`sN`: We plot the RB output prediction, $s_N(\boldsymbol{\mu})$, as function of $\mu_{\texttt{mu\_index}}$ in the desired range and for the prescribed `N` or the desired `epsdes`. In the plot we also report `DeltaN`, the RB *a posteriori* output error bound, $\Delta_N^s(\boldsymbol{\mu})$.

5.4.2 Greedy Restarts: "Offline Adaptivity"

There are many scenarios in which we might wish to modify a problem: we may wish to change Greedy specifications (of interest for a variety of reasons, as elaborated upon below); we may wish to change the specification of the parameter domain, $\mathcal{D}$ (of interest in incorporating "feedback" from an application such as optimization or parameter estimation); we may wish to change the specification of the truth approximation (of interest in development — rapid testing on a coarse truth approximation followed by production on a fine truth approximation). In almost all scenarios we can re-use most or all of the problem definition; and in many scenarios, we can take good advantage of the existing RB approximation and *a posteriori* error estimation data — a "restart" (of the Greedy algorithm).

We shall focus here on a change to the Greedy specifications. This need can arise for a variety of reasons:

R1 The initial Greedy terminates prematurely (a computer crash or a user `^C` [Ctrl+C]) such that neither the error tolerance $\varepsilon_{\text{tol,min}}$ or the RB dimension limit `Nbarmax` is reached; the user now wishes to recover/continue the calculation. (Note that the Greedy code checkpoints — saves results at intermediate steps in recovery files that are only deleted at the successful completion of a run).

R2 The initial `Nbarmax` is too small to achieve the desired $\varepsilon_{\text{tol,min}}$; the user now wishes to expend — or only now has access to — greater resources to reach the requisite accuracy.

R3 The application is more demanding or sensitive than anticipated; the user decides that a higher accuracy RB output approximation is needed.

R4 The Online RB predictions systematically present an output error larger than $\varepsilon_{\text{tol,min}}$: the user decides to refine the initial train sample Ξ_{train}. (Or the user intentionally first performs a Greedy algorithm over a coarse $\Xi_{\text{train}}^{\text{coarse}}$ in order to reduce Offline computational cost, and now wishes to consider $\Xi_{\text{train}}^{\text{fine}}$ (see Section 3.4.4).)

In all these cases we can re-use both the problem definition and also all the existing RB approximation and *a posteriori* error estimation data: a "restart." (Recall from Section 3.4.4 that our Greedy algorithm is defined relative to any initial sample and associated RB space and basis: it is this restart feature that we exercise here.)

Step1 should *not* be re-executed: this assures compatibility of the original PROBDEF with all subsequent RB approximations. Our focus is thus on Step2. Note that once Step2 has been re-executed, Step3 proceeds exactly as in Section 5.4.1.

Note that the user may wish to save copies of *PROBNAME_OFFLINE and *PROBNAME_ONLINE before embarking on a restart (and similarly for the results of a restart before embarking on a subsequent restart). In this fashion the entire set of refinements remains available and uncorrupted.

We now proceed to indicate the modifications to the Step2 script in each of the cases *R1*–*R4* above.

R1 . To start a recovery procedure the Greedy algorithm must have at least generated samples and completed the first cycle. In the event of a fatal crash, the user must first recover data from temporary files saved as OFFLINE_recovery.mat and ONLINE_recovery.mat.

```
>>load OFFLINE_recovery OFFLINE
>>load ONLINE_recovery ONLINE
>>*PROBNAME_OFFLINE=OFFLINE
>>*PROBNAME_ONLINE=ONLINE
```

```
    >>save *PROBNAME_OFFLINE

    >>save *PROBNAME_ONLINE
```

in order restore the RB structures. The Step2 script should now be modified per the below and then re-executed.

```
      ----begin modifications to *PROBNAME_Step2_parcoer_compliant.m----

*PROBNAME_OFFLINE.space.sample.newflg = 0

   % use the existing sample from earlier and

   % continue the Greedy (hopefully) to conclusion

*PROBNAME_OFFLINE.space.restartflg= 1

  % 1 indicates a Greedy restart

  ----end modifications to *PROBNAME_Step2_parcoer_compliant_compliant.m----
```

R2. The Step2 script should be modified per the below and then re-executed.

```
      ----begin modifications to *PROBNAME_Step2_parcoer_compliant.m----

*PROBNAME_OFFLINE.space.sample.newflg = 0

   % use the existing sample from earlier

...

*PROBNAME_OFFLINE.space.Nbarmax = ?

  % but now increase the upper limit for the number of

  % Greedy cycles to (hopefully) achieve the desired $\varepsilon_{\rm tol,min}$

*PROBNAME_OFFLINE.sample.restartflg= 1

   % 1 indicates a Greedy restart

       ----end modifications to *PROBNAME_Step2_parcoer_compliant.m----
```

R3. The Step2 script should be modified per the below and then re-executed.

```
      ----begin modifications to *PROBNAME_Step2_parcoer_compliant.m----

*PROBNAME_OFFLINE.space.sample.newflg = 0
```

```
   % use the existing sample from earlier
*PROBNAME.OFFLINE.space.tol = ?
   % but with a tighter tolerance/higher accuracy requirement
*PROBNAME_OFFLINE.space.restartflg= 1
  % 1 indicates a Greedy restart
          ----end modifications to *PROBNAME_Step2_parcoer_compliant.m----
```

R4. The Step2 script should be modified per the below and then re-executed.

```
         ----begin modifications to *PROBNAME_Step2_parcoer_compliant.m----

*PROBNAME_OFFLINE.space.sample.newflg = 1
  % use a new sample
*PROBNAME_OFFLINE.space.sample.size = ?
  % with a larger number of points $n_{\text{train}}$ than earlier
*PROBNAME_OFFLINE.sample.restartflg= 1
   % 1 indicates a Greedy restart
          ----end modifications to *PROBNAME_Step2_parcoer_compliant.m----
```

5.5 Problem Creation: Reference Manual

5.5.1 Datafiles

We present in Table 5.1 the contents of the structures (we omit here the `*PROBNAME_` prefix) `PROBDEF`, `OFFLINE`, and `ONLINE` stored in the files `PROBDEF.mat`, `OFFLINE.mat`, and `ONLINE.mat`. For each structure we indicate (by column): the names of the variables (e.g., `PROBDEF.Q_affine_a`); the type of MATLAB® data (e.g., scalar); the mathematical symbol in the text, (e.g., Q_a); the Section(s) and or equation(s) which precisely define the quantity (e.g Chapter 2); and finally the source of the data. For the latter, we abbreviate "u" for user, "G" for the `Greedy_parcoer_compliant` code, and "G*" for an internal function called by Greedy.

Name	Type	Math Symbol-Sec.(eq.)	Source
PROBDEF.Q_affine_a	scalar	Q_a - 2.1.2	u
PROBDEF.Q_affine_f	scalar	Q_f - 2.1.2	u
PROBDEF.P	scalar	P - 1.2.3	u
PROBDEF.mu_min	$1 \times P$	$\boldsymbol{\mu}^{\min}$ - 1.4.2	u
PROBDEF.mu_max	$1 \times P$	$\boldsymbol{\mu}^{\max}$ - 1.4.2	u
PROBDEF.mu_bar	$1 \times P$	$\bar{\boldsymbol{\mu}}$ - 2.1.2	u
PROBDEF.check.Qa	flag $1 \times (Q_a)$	. - 1.2.6	G^*
PROBDEF.check.Aq	flag $1 \times (Q_a)$	. - 1.2.6	G^*
OFFLINE.FEM.matrix.Aq	$Q_a \times \mathcal{N}_\mathrm{t} \times \mathcal{N}_\mathrm{t}$ (sparse)	$\underline{\mathbb{A}}^{\mathcal{N}q}$ - (2.42)	u
OFFLINE.FEM.matrix.Fq	$Q_f \times \mathcal{N}_\mathrm{t}$ (sparse)	$\underline{\mathbb{F}}^{\mathcal{N}q}$ - (2.44)	u
OFFLINE.FEM.matrix.Xnorm	$\mathcal{N}_\mathrm{t} \times \mathcal{N}_\mathrm{t}$ (Cholesky)[1]	$\underline{\mathbb{X}}^{\mathcal{N}}$ - (2.45)	G
OFFLINE.FEM.matrix.HT	$\mathcal{N}_\mathrm{t} \times \mathcal{N}_\mathrm{t}$ (sparse)	. - .	G
OFFLINE.space.restartflg	scalar (flag: $0-1$)	. - .	u
OFFLINE.space.sample.newflg	scalar (flag: $0-1$)	. - .	u
OFFLINE.space.sample.densityflg	scalar (flag: $0-1$)	. - 1.4.2	u
OFFLINE.space.sample.size	scalar	n_train - 3.4	u
OFFLINE.space.absrelflg	scalar (flag: $0-1$)	. - 3.4.4	u
OFFLINE.space.tol	scalar	$\varepsilon_{\mathrm{tol,min}}$ - 3.4.4	u
OFFLINE.space.Nbar	scalar	$\overline{N}$ - 3.4.4	u
OFFLINE.space.Nmax	scalar	$N_{\max}$ - 3.4.4	G
OFFLINE.space.sample.mu_samples	$n_\mathrm{train} \times P$	Ξ_train - 2.1.2	G^*
OFFLINE.space.sample.sample_in_basis	$1 \times n_\mathrm{train}$	. - .	G
OFFLINE.space.Z	$\mathcal{N}_\mathrm{t} \times N_{\max}$	$\mathbb{Z}_N$ - 3.2.1	G^*
OFFLINE.space.matrix.Zqf	$\mathcal{N}_\mathrm{t} \times 1$	-	G^*
OFFLINE.space.matrix.Zqa	$\mathcal{N}_\mathrm{t} \times N$	-	G^*
ONLINE.space.absrelflg	scalar (flag: $0-1$)	. - 3.4.4	G
ONLINE.space.Mus	$N_{\max} \times P$	S_N - 3.4.4	G
ONLINE.space.basisInds	(flag) $1 \times n_\mathrm{train}$	. - .	G
ONLINE.space.eps_out_star	$1 \times N_{\max}$	$\varepsilon_N^{out,*}$ - 3.4.4	G
ONLINE.space.Nmax	scalar	$N_{\max}$ - 3.4.4	G
ONLINE.RB.matrix.Zqfprime_Fq	1×1	-	G^*
ONLINE.RB.matrix.Zqfprime_Aq_Z	$1 \times N_{\max}$	-	G^*
ONLINE.RB.matrix.Aqn	$Q_a \times N_{\max} \times N_{\max}$	$\underline{\mathbb{A}}_N^q$ - (3.40)	G^*
ONLINE.RB.matrix.Zqaprime_Aq_Z	$N_{\max} \times N_{\max}$	-	G^*
ONLINE.RB.matrix.FN	$Q_f \times N_{\max} \times 1$	$\underline{\mathbb{F}}_N^q$ - (3.41)	G^*
ONLINE.RB.matrix.X_norm_rb	$N_{\max} \times N_{\max}$	$\underline{\mathbb{X}}_N$ - (4.66)	G

Table 5.1: Data Files.[1]When Cholesky factorization is performed we store in Xnorm just the upper triangular factor (H); denoting by HT the transpose of H, $\underline{\mathbb{X}}^{\mathcal{N}_\mathrm{t}}$ =HT*H. Note the Greedy infers $\underline{\mathbb{X}}^{\mathcal{N}_\mathrm{t}}$ from the $\underline{\mathbb{A}}^{\mathcal{N}_\mathrm{t}q}$ and $\overline{\boldsymbol{\mu}}$.

5.5.2 Codes

(See Section 5.4.1 for user-defined functions.)

Description of the function `Greedy_parcoer_compliant`**.**

IN's and OUT's. We present in Table 5.2 the IN's and OUT's for `Greedy_parcoer_compliant`: in the first column we list all the elements of `PROBDEF`, `OFFLINE`, and `ONLINE` (already defined in Section 5.5.1), and in the second column we indicate "IN" or "OUT." Note that some of the elements — such as the RB basis functions and the RB matrices — of `OFFLINE` and `ONLINE` can be either IN's or IN's and OUT's: if we create a problem from "scratch," these elements are OUT's; if we modify a problem via restart (see Section 5.4.2), these updated elements are IN's and OUT's. Service internal variables are denoted with "S".

Diagnostics. We present in Table 5.3 the diagnostics available: the first column is the error message; the second column the probable causes of the error; and the third column the possible remedies.

Description of the function `Online_parcoer_compliant`**.**

IN's and OUT's. We present in Table 5.4 the structure (and functions) IN's and OUT's for `Online_rbMIT_parcoer_compliant` (called by `*PROBNAME_Online`): in the first column we list all the elements of `PROBDEF`, `OFFLINE`, and `ONLINE` (already defined in Section 5.5.1), and in the second column we indicate "IN" or "OUT." In Table 5.5 we present the "runtime" (non-structure) IN's and OUT's for `Online_rbMIT_parcoer_compliant`: in the first column we list first the IN's and then the OUT's; in the subsequent columns we indicate the type of data, the mathematical symbol in the text, and the Section(s) and or equation(s) which precisely define the quantity.

Name	IN/OUT
PROBDEF.Q_affine_a	IN
PROBDEF.Q_affine_f	IN
PROBDEF.P	IN
PROBDEF.mu_min	IN
PROBDEF.mu_max	IN
PROBDEF.mu_bar	IN
PROBDEF.check.Qa	S
PROBDEF.check.Aq	S
OFFLINE.FEM.matrix.Aq	IN
OFFLINE.FEM.matrix.Fq	IN
OFFLINE.FEM.matrix.Xnorm	S
OFFLINE.FEM.matrix.HT	S
OFFLINE.FEM.matrixflg	IN
OFFLINE.space.sample.newflg	IN
OFFLINE.space.sample.densityflg	IN
OFFLINE.space.sample.size	IN
OFFLINE.space.absrelflg	IN
OFFLINE.space.tol	IN
OFFLINE.space.Nbar	IN
OFFLINE.space.N	S
OFFLINE.space.Nmax	OUT/IN
OFFLINE.space.sample.mu_samples	S
OFFLINE.space.sample.sample_in_basis	OUT/IN
OFFLINE.space.Z	S
OFFLINE.space.matrix.Zqf	S
OFFLINE.space.matrix.Zqa	S
ONLINE.space.absrelflg	OUT
ONLINE.space.Mus	OUT/IN
ONLINE.space.theta_a	OUT/IN
ONLINE.space.theta_f	OUT/IN
ONLINE.space.basisInds	OUT/IN
ONLINE.space.eps_out_star	OUT/IN
ONLINE.space.Nmax	OUT
ONLINE.RB.matrix.Zqfprime_Fq	OUT
ONLINE.RB.matrix.Zqfprime_Aq_Z	OUT
ONLINE.RB.matrix.Aqn	OUT/IN
ONLINE.RB.matrix.Zqaprime_Aq_Z	OUT
ONLINE.RB.matrix.FN	OUT/IN
ONLINE.RB.matrix.X_norm_rb	OUT/IN
InsideOutsideD	IN
Get_Theta_q	IN

Table 5.2: Greedy Specifications.

Error Message	Cause	Possible Remedies
`all mu_min have to be > 0.0`	requesting a log distribution of sample with `mu_min`< 0	set `mu_min`> 0
`parameters are defined incorrectly`	error during sample generation	check parameter range
`all theta coefficients have to be > 0: parametric coercivity is lost`	$\Theta_a^q < 0$	check Θ_a^q's or go to Part II if Θ_a^q's are correct

`matrices Aq should be symmetric`
`Aq{q} is not symmetric for q=?`
	one of the matrices is not symmetric	check Aq's or go to Part II if Aq's are correct

`matrices Aq should be positive semi-def.: Aq{q} has negative eigenvalues for q=?`	one of the matrices is not positive semi-def.	check Aq's or go to Part II if Aq's are correct
`increase Nbarmax or reduce the tolerance`	`Nbarmax` is too small or `tol` is too strict	re-start the Greedy with a greater `Nbarmax` or change `tol`

Table 5.3: Greedy Diagnostics.

As described in greater detail in Section 4.4.3, the two inputs `N` and `epsdes` can serve in two different "modes": if `N > 0`, and `epsdes` is set to zero, then `*PROBNAME_Online` returns $s_N(\boldsymbol{\mu})$ and $\Delta^{\cdot}_N(\boldsymbol{\mu})$. If `N =0`, then `*PROBNAME_Online` returns $s^{\cdot}_{N^*}(\boldsymbol{\mu})$ and $\Delta^{\cdot}_{N^*}(\boldsymbol{\mu})$ where N^* is the smallest N' such that $\Delta^{\cdot}_{N'} \leq$ `epsdes` (in actual fact, we settle for an efficiently calculated slightly sub-optimal result, per Section 4.4.3). (As indicated in Table 5.5, $. = s$ for `ONLINE.space.absrelflg` $= 0$ and $. = s, \mathrm{rel}$ for `ONLINE.space.absrelflg` $= 1$.)

Diagnostics. We present in Table 5.6 the diagnostics available: the first column is the error message; the second column the probable causes of the error; and the third column the possible remedies.

Name	IN/OUT
PROBDEF.Q_affine_a	IN
PROBDEF.Q_affine_f	IN
PROBDEF.P	IN
PROBDEF.mu_min	IN
PROBDEF.mu_max	IN
ONLINE.space.absrelflg	IN
ONLINE.space.eps_out_star	IN
ONLINE.space.Nmax	IN
ONLINE.RB.matrix.Zqfprime_Fq	IN
ONLINE.RB.matrix.Zqfprime_Aq_Z	IN
ONLINE.RB.matrix.Aqn	IN
ONLINE.RB.matrix.Zqaprime_Aq_Z	IN
ONLINE.RB.matrix.FN	IN
ONLINE.RB.matrix.X_norm_rb	IN
InsideOutsideD	IN
Get_Theta_q	IN

Table 5.4: Online Specifications: Structures.

Name	Type	Math Symbol - Sec.(eq.)
muvectorvalue	$1 \times P$	$\boldsymbol{\mu}$ - 1.2.3
N	scalar	N - 3.2
epsdes	scalar	ε_{des} - 3.4.4
sN	scalar	$s_N(\boldsymbol{\mu})$ - (3.35)
DeltaN	scalar	$\Delta^{\cdot}_N(\boldsymbol{\mu})$ - 4.3

Table 5.5: Online Specifications: Runtime.

Error Message	Cause	Possible Remedies
parameter value is out of range	$\boldsymbol{\mu} \notin \mathcal{D}$	check muvectorvalue
size of reduced basis space should be smaller than Nmax	N input is too large	reduce N so that N<=Nmax
epsdes is too small for given RB approximation	to get the desidered epsdes, Nmax is not sufficient	enrich the basis or reduce epsdes

Table 5.6: Online Diagnostics.

Bibliography

[1] *Partial Differential Equation Toolbox for use with MATLAB® COMSOL AB: User's Guide, Version 1.* The MathWorks, Inc, Natick MA, 2006.

[2] R. A. Adams. *Sobolev Spaces.* Academic Press, 1975.

[3] R. A. Adams. *Calculus: A Complete Course.* Addison Wesley, 2002.

[4] M. Ainsworth and J. T. Oden. *A posteriori* error estimation in finite element analysis. *Comp. Meth. Appl. Mech. Engrg.*, 142:1–88, 1997.

[5] M. Ainsworth and J. T. Oden. A Posteriori *Error Estimation in Finite Element Analysis.* Wiley-Interscience, 2000.

[6] B. O. Almroth, P. Stern, and F. A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16:525–528, 1978.

[7] T. L. Anderson. *Fracture Mechanics: Fundamentals and Application.* CRC, third edition, 2005.

[8] J. A. Atwell and B. B. King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and Computer Modelling*, 33(1-3):1–19, 2001.

[9] I. Babuška. Error-bounds for finite element method. *Numerische Mathematik*, 16:322–333, 1971.

[10] I. Babuška and J. Osborn. Eigenvalue problems. In *Handbook of Numerical Analysis*, volume II, pages 641–787. Elsevier, 1991.

[11] I. Babuška and W. Rheinboldt. *A posteriori* error estimates for the finite element method. *Int. J. Numer. Meth. Eng.*, 12:1597–1615, 1978.

[12] I. Babuška and W. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15:736–754, 1978.

[13] I. Babuška and T. Strouboulis. *The Finite Element Method and its Reliability.* Numerical Mathematics and Scientific Computation. Clarendon Press, Oxford,UK, 2001.

[14] Z. J. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1-2):9–44, 2002.

[15] E. Balmes. Parametric families of reduced finite element models: Theory and applications. *Mechanical Systems and Signal Processing*, 10(4):381–394, 1996.

[16] E. Balsa-Canto, A.A. Alonso, and J.R. Banga. Reduced-order models for nonlinear distributed process systems and their application in dynamic optimization. *Industrial & Engineering Chemistry Research*, 43(13):3353–3363, 2004.

[17] H. T. Banks and K. Kunisch. *Estimation Techniques for Distributed Parameter Systems.* Systems & Control: Foundations & Applications. Birkhäuser, 1989.

[18] 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.*, 339:667–672, 2004.

[19] A. Barrett and G. Reddien. On the reduced basis method. *Z. Angew. Math. Mech.*, 75(7):543–549, 1995.

[20] O. Bashir, K. Willcox, and O. Ghattas. Hessian-based model reduction for large-scale systems with initial condition inputs. *Int. J. for Num. Meth. in Engineering*, 2007. Submitted.

[21] K.-J. Bathe. *Finite Element Procedures.* Prentice Hall, 1996.

[22] R. Becker and R. Rannacher. A feedback approach to error control in finite element method: Basic analysis and examples. *East - West J. Numer. Math.*, 4:237–264, 1996.

[23] P. Benner, V. Mehrmann, and D.C. Sorensen (Eds.). *Dimension Reduction of Large-Scale Systems.* Lecture Notes in Computational Science and Engineering. Springer, Heildeberg, 2003.

[24] A. Bensoussan, J. L. Lions, and G. Papanicolaou. *Asymptotic Analysis of Periodic Structures.* North-Holland, Amsterdam, 1978.

[25] S. Boyaval. Application of reduced basis approximation and *a posteriori* error estimation to homogenization theory. 2007. In progress.

[26] F. Brezzi. On the existence, uniqueness, and approximation of saddle point problems arising from Lagrangian multipliers. *R.A.I.R.O., Anal. Numér.*, 2:129–151, 1974.

[27] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics.* Springer Verlag, 1991.

[28] F. Brezzi, J. Rappaz, and P.A. Raviart. Finite dimensional approximation of nonlinear problems. Part I: Branches of nonsingular solutions. *Numerische Mathematik*, 36:1–25, 1980.

[29] C. Le Bris. *Private Communication.* MIT, 2006.

[30] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici. *A Priori* convergence of multi-dimensional parametrized reduced–basis approximations. 2007. In progress.

[31] T. Bui-Thanh, M. Damodaran, and K. Willcox. Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics (AIAA Paper 2003-4213). In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*, 2003.

[32] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space (AIAA Paper 2007-2049). In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference*, 2007.

[33] J. Burkardt, M. Gunzburger, and H. Lee. Centroidal Voronoi tessellation-based reduced-order modeling of complex systems. *SIAM J. Sci. Comput.*, 2007. In press.

[34] G. Caloz and J. Rappaz. Numerical analysis for nonlinear and bifurcation problems. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis, Vol. V*, Techniques of Scientific Computing (Part 2), pages 487–637. Elsevier Science B.V., 1997.

[35] E. Cancès, C. Le Bris, Y. Maday, and G. Turinici. Towards reduced basis approaches in ab initio electronic structure computations. In *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala)*, volume 17, pages 461–469, 2002.

[36] E. Cancès, C. Le Bris, N. C. Nguyen, Y. Maday, A. T. Patera, and G. S. H. Pau. Feasibility and competitiveness of a reduced basis approach for rapid electronic structure calculations in quantum chemistry. In *Proceedings of the Workshop for High-dimensional Partial Differential Equations in Science and Engineering (Montreal)*, 2006. Submitted.

[37] W. Cazemier. *Proper Orthogonal Decomposition and Low Dimensional Models for Turbolent Flows.* University of Groningen, 1997.

[38] J. Chen and S-M. Kang. Model-order reduction of nonlinear mems devices through arclength-based Karhunen-Loéve decomposition. In *Proceeding of the IEEE international Symposium on Circuits and Systems*, volume 2, pages 457–460, 2001.

[39] Y. Chen and J. White. A quadratic method for nonlinear model order reduction. In *Proceedings of the international Conference on Modeling and Simulation of Microsystems*, pages 477–480, 2000.

[40] E.A. Christensen, M. Brøns, and J.N. Sørensen. Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows. *SIAM J. Scientific Computing*, 21(4):1419–1434, 2000.

[41] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* Classics in Applied Mathematics, 40. SIAM, 2002.

[42] G. Dahlquist and Ȧ. Björck. *Numerical Methods.* Prentice Hall, NJ, 1974.

[43] L. Daniel, C.S. Ong, and J. White. Geometrically parametrized interconnect performance models for interconnect synthesis. In *Proceedings of the 2002 International Symposium on Physical Design, ACM press*, pages 202–207, 2002.

[44] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology.* Springer-Verlag, 1988.

[45] L. Demkowicz. Babuška ↔ Brezzi?? *Technical Report ICES, Institute for Computational Engineering and Sciences*, (06-08), 2006.

[46] Earl H. Dowell and Kenneth C. Hall. Modeling of fluid structure interaction. *Annu. Rev. Fluid. Mech.*, 33:445–490, 2001.

[47] Q. Du, V. Faber, and M. D. Gunzburger. Centroidal Voronoi tesselations: applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.

[48] G. Duvaut and J.L. Lions. *Inequalities in Mechanics and Physics.* Springer-Verlag, Berlin, 1976.

[49] O. Farle, V. Hill, P. Nickel, and R. Dyczij-Edlinger. Multivariate finite element model order reduction for permittivity or permeability estimation. *IEEE Transactions on Megnetics*, 42:623–626, 2006.

[50] 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(1):21–28, 1983.

[51] R. L. Fox and H. Miura. An approximate analysis technique for design calculations. *AIAA Journal*, 9(1):177–179, 1971.

[52] V. Girault and P. Raviart. *Finite Element Approximation of the Navier-Stokes Equations.* Springer-Verlag, 1986.

[53] M. Grepl. *Reduced-Basis Approximations and* A Posteriori *Error Estimation for Parabolic Partial Differential Equations.* PhD thesis, Massachusetts Institute of Technology, May 2005.

[54] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN (Math. Model. Numer. Anal.)*, 2007.

[55] M. A. Grepl, N. C. Nguyen, K. Veroy, A. T. Patera, and G. R. Liu. Certified rapid solution of partial differential equations for real-time parameter estimation and optimization. In *Proceedings of the 2*$^{\text{nd}}$ *Sandia Workshop of PDE-Constrained Optimization: Towards Real-Time PDE-Constrained Optimization*, SIAM Computational Science and Engineering Book Series, 2007. In press.

[56] 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.)*, 39(1):157–181, 2005.

[57] M. D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows.* Academic Press, 1989.

[58] M. D. Gunzburger. *Perspectives in Flow Control and Optimization.* Advances in Design and Control. SIAM, 2003.

[59] M. D. Gunzburger, J. Peterson, and J. N. Shadid. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comp. Meth. Applied Mech.*, 196:1030–1047, 2007.

[60] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized evolution equations. 2006. Submitted.

[61] D. B. P. Huynh and A. T. Patera. Reduced-basis approximation and *a posteriori* error estimation for stress intensity factors. *Int. J. Num. Meth. Eng.*, 2006. Submitted.

[62] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Acad. Sci. Paris, Analyse Numérique*, 2006. Submitted.

[63] F. P. Incropera and D. P. DeWitt. *Fundamentals of Heat and Mass Transfer.* John Wiley & Sons, 1990.

[64] A. Iske. *Multiresolution methods in scattered data modeling.* Springer, 2004.

[65] K. Ito and S. S. Ravindran. A reduced basis method for control problems governed by PDEs. In W. Desch, F. Kappel, and K. Kunisch, editors, *Control and Estimation of Distributed Parameter Systems*, pages 153–168. Birkhäuser, 1998.

[66] 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, 1998.

[67] K. Ito and S. S. Ravindran. Reduced basis method for optimal control of unsteady viscous flows. *International Journal of Computational Fluid Dynamics*, 15(2):97–113, 2001.

[68] K. Ito and J. D. Schroeter. Reduced order feedback synthesis for viscous incompressible flows. *Mathematical And Computer Modelling*, 33(1-3):173–192, 2001.

[69] A. D. Izaak. Kolmogorov widths in finite-dimensional spaces with mixed norms. *Mathematical Notes*, 55(1):43–52, 1994.

[70] M.A. Jabbar and A.B. Azeman. Fast optimization of electromagnetic-problems:the reduced-basis finite element approach. *IEEE Transactions on Magnetics*, 40(4):2161–2163, 2004.

[71] L. V. Kantorovich and G. P. Akilov. *Functional Analysis in Normed Spaces*. The Macmillan Company, 1964.

[72] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae*, 37, 1946.

[73] P. Krysl, S. Lall, and J. E. Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for Numerical Methods in Engineering*, 51:479–504, 2001.

[74] S. N. Kudryavtsev. Widths of classes of finitively smooth functions in Sobolev spaces. *Mathematical Notes*, 77(4):535–539, 2005.

[75] K. Kunish and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Num. Analysis*, 40(2):492–515, 2002.

[76] M. Y. Lin Lee. Estimation of the error in the reduced-basis method solution of differential algebraic equations. *SIAM Journal of Numerical Analysis*, 28:512–528, 1991.

[77] P. A. LeGresley and J. J. Alonso. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. In *Fluids 2000 Conference and Exhibit, Denver, CO*, 2000. Paper 2000-2545.

[78] R. Leis. *Initial Boundary Value Problems in Mathematical Physics*. Wiley, 1986.

[79] J.-L. Lions and E. Magenes. *Non-Homogenous Boundary Value Problems and Applications*. Springer-Verlag, 1972.

[80] M. M. Loeve. *Probablity Theory*. Van Nostrand, 1955.

[81] A. E. Løvgren, Y. Maday, and E. M. Rønquist. A reduced basis element method for complex flow systems. In *ECCOMAS CFD 2006 Proceedings, P. Wesseling, E. Oñate, J. Periaux (Eds.) TU Delft, The Netherlands*. 2006.

[82] A. E. Løvgren, Y. Maday, and E. M. Rønquist. The reduced basis element method for fluid flows. *Journal of Mathematical Fluid Mechanics*, 2006.

[83] A. E. Løvgren, Y. Maday, and E. M. Rønquist. A reduced basis element method for the steady Stokes problem. *Mathematical Modelling and Numerical Analysis (M2AN)*, 40(3):529–552, 2006.

[84] A. E. Løvgren, Y. Maday, and E. M. Rønquist. A reduced basis element method for the steady Stokes problem: Application to hierarchical flow systems. *Modeling, Identification and Control*, 27(2):79–94, 2006.

[85] J. Lumley and P. Blossey. Control of turbulence. *Annu. Rev. Fluid. Mech.*, 30:311–327, 1998.

[86] H.V. Ly and H.T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33, 2001.

[87] 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, 2000.

[88] Y. Maday. Reduced–basis method for the rapid and reliable solution of partial differential equations. In *Proceedings of International Conference of Mathematicians, Madrid*. European Mathematical Society Eds., 2006.

[89] 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, Collége de France Seminar Volume XIV*, pages 533–569. Elsevier Science B.V., 2002.

[90] 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.

[91] 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, 2002.

[92] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[93] M. Meyer and H. G. Matthies. Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods. *Computational Mechanics*, 31(1-2):179–191, 2003.

[94] A. F. Mills. *Heat Transfer*. Prentice-Hall, Inc., 2nd edition, 1999.

[95] D. A. Nagy. Modal representation of geometrically nonlinear behaviour by the finite element method. *Computers and Structures*, 10:683–688, 1979.

[96] A. W. Naylor and G. R. Sell. *Linear Operator Theory in Engineering and Science*, volume 40 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1982.

[97] A. J. Newman. Model reduction via the Karhunen-Loeve expansion part i: an exposition. *Technical Report Institute for System Research University of Maryland*, (96-322), 1996.

[98] N. C. Nguyen, K. Veroy, and A. T. Patera. Certified real-time solution of parametrized partial differential equations. In S. Yip, editor, *Handbook of Materials Modeling*, pages 1523–1558. Springer, 2005.

[99] A. K. Noor. Recent advances in reduction methods for nonlinear problems. *Comput. Struct.*, 13:31–44, 1981.

[100] A. K. Noor. On making large nonlinear problems small. *Comp. Meth. Appl. Mech. Engrg.*, 34:955–985, 1982.

[101] 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.

[102] A. K. Noor and J. M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18(4):455–462, 1980.

[103] A. K. Noor and J. M. Peters. Bifurcation and post-buckling analysis of laminated composite plates via reduced basis techniques. *Comp. Meth. Appl. Mech. Engrg.*, 29:271–295, 1981.

[104] A. K. Noor and J. M. Peters. Tracing post-limit-point paths with reduced basis technique. *Comp. Meth. Appl. Mech. Engrg.*, 28:217–240, 1981.

[105] 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.

[106] A. K. Noor and J. M. Peters. Recent advances in reduction methods for instability analysis of structures. *Comput. Struct.*, 16:67–80, 1983.

[107] 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.

[108] J. T. Oden and L. F. Demkowicz. *Functional Analysis.* CRC Press, Boca Raton, 1996.

[109] I. Oliveira and A. T. Patera. Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations. 2006. In press.

[110] M. Paraschivoiu, J. Peraire, Y. Maday, and A. T. Patera. Fast bounds for outputs of partial differential equations. In J. Borgaard, J. Burns, E. Cliff, and S. Schreck, editors, *Computational methods for optimal design and control*, pages 323–360. Birkhäuser, 1998.

[111] A.T. Patera and E.M. Rønquist. Reduced basis approximations and *a posteriori* error estimation for a Boltzmann model. *Computer Methods in Applied Mecanics and Engineering*, 2006. Submitted.

[112] J. S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM J. Sci. Stat. Comput.*, 10(4):777–786, 1989.

[113] J. R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proceeding of the 37th ACM/IEEE Design Automation Conference*, pages 184–189, 2000.

[114] J. R. Phillips. Projection-based approaches for model reduction of weakly nonlinear systems, time-varying systems. In *IEEE Transactions On Computer-Aided Design of Integrated Circuit and Systems*, volume 22, pages 171–187, 2003.

[115] N.A. Pierce and M. B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Review*, 42(2):247–264, 2000.

[116] A. Pinkus. *n-Widths in Approximation Theory.* Springer, 1985.

[117] O. Pironneau. Calibration of barrier options. In W. Fitzgibbon, R. Hoppe, J. Periaux, O. Pironneau, and Y. Vassilevski, editors, *Advances in Numerical Mathematics.* Moscow, Institute of Numerical Mathematics, Russian Academy of Sciences and Houston, Department of Mathematics, University of Houston, 2006.

[118] T. A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, 1985.

[119] T. A. Porsching and M. Y. Lin Lee. The reduced-basis method for initial value problems. *SIAM Journal of Numerical Analysis*, 24:1277–1287, 1987.

[120] 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 bounds methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

[121] C. Prud'homme, D. Rovas, K. Veroy, and A. 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. Programming.

[122] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis method. 2006. Submitted.

[123] A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis method for optimal control af advection-diffusion processes. In W. Fitzgibbon, R. Hoppe, J. Periaux, O. Pironneau, and Y. Vassilevski, editors, *Advances in Numerical Mathematics*, pages 193–216. Moscow, Institute of Numerical Mathematics, Russian Academy of Sciences and Houston, Department of Mathematics, University of Houston, 2006.

[124] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer, New York, 2000.

[125] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 2nd edition, 1997.

[126] S. S. Ravindran. Reduced-order adaptive controllers for fluid flows using pod. *J. of Scientific Computing*, 15(4):457–478, 2000.

[127] S. S. Ravindran. A reduced order approach to optimal control of fluids flow using proper orthogonal decomposition. *Int. J. of Numerical Methods in Fluids*, 34(5):425–448, 2000.

[128] S. S. Ravindran. Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition. *SIAM J. Sci. Comput.*, 23(6):1924–1942, 2002.

[129] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In *IEEE*

Transactions On Computer-Aided Design of Integrated Circuit and Systems, volume 22, pages 155–170, 2003.

[130] W. C. Rheinboldt. Numerical analysis of continuation methods for nonlinear structural problems. *Computers and Structures*, 13(1-3):103–113, 1981.

[131] 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.

[132] D. Rovas, L. Machiels, and Y. Maday. Reduced basis output bounds methods for parabolic problems. *IMA J. Appl. Math.*, 2005.

[133] G. Rozza. Real-time reduced basis techniques for arterial bypass geometries. In K.J. Bathe, editor, *Computational Fluid and Solid Mechanics*, pages 1283–1287. Elsevier, 2005. Proceedings of the Third M.I.T. Conference on Computational Fluid and Solid Mechanics, June 14-17, 2005.

[134] G. Rozza. Reduced-basis methods for elliptic equations in sub-domains with *a posteriori* error bounds and adaptivity. *Appl. Numer. Math.*, 55(4):403–424, 2005.

[135] G. Rozza. Reduced basis method for Stokes equations in domains with non-affine parametric dependence. *Comp. Vis. Science*, 2006. Available online.

[136] G. Rozza and K. Veroy. On the stability of reduced basis method for Stokes equations in parametrized domains. *Comp. Meth. Appl. Mech. and Eng.*, 196:1244–1260, 2007.

[137] S. Sen. Reduced basis approximation and *a posteriori* error estimation for many-parameter heat conduction problems. 2007. In progress.

[138] S. Sen. *Reduced-Basis Approximation and* A Posteriori *Error Estimation for Many-Parameter Problems.* PhD thesis, Massachusetts Institute of Technology., In progress.

[139] S. Sen, K. Veroy, D. B. P. Huynh, S. Deparis, N. C. Nguyen, and A. T. Patera. “Natural norm” *a posteriori* error estimators for reduced basis approximations. *Journal of Computational Physics*, 217:37–62, 2006.

[140] I. H. Shames. *Introduction to Solid Mechanics.* Prentice Hall, New Jersey, 2nd edition, 1989.

[141] G. Shi and C.-J. R. Shi. Parametric model order reduction for interconnect analysis. In *Proceedings of the 2004 Conference on Asia South Pacific design automation: electronic design and solution fair, IEEE press*, pages 774–779, 2004.

[142] S. Sirisup, D. Xiu, and G. Karniadakis. Equation-free/Galerkin-free POD-assisted computation of incompressible flows. *Journal of Computational Physics*, 207:617–642, 2005.

[143] L. Sirovich. Turbulence and the dynamics of coherent structures, part 1: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.

[144] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[145] T. Tonn and K. Urban. A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies. In *ECCOMAS CFD 2006 Proceedings, P. Wesseling, E. Oñate, J. Periaux (Eds.) TU Delft, The Netherlands.* 2006.

[146] L. Trefethen and D. Bau III. *Numerical Linear Algebra.* SIAM, 1997.

[147] 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. *International Journal for Numerical Methods in Fluids*, 47:773–788, 2005.

[148] 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, 2003.

[149] 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. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, 2003. Paper 2003-3847.

[150] 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. *ESAIM: Control, Optimization and Calculus of Variations*, 8:1007–1028, 2002.

[151] D. S. Weile and E. Michielssen. Analysis of frequency selective surfaces using two-parameter generalized rational Krylov model-order reduction. *IEEE Transactions on Antennas and Propagation*, 49(11):1539–1549, 2001.

[152] D. S. Weile, E. Michielssen, and K. Gallivan. Reduced-order modeling of multiscreen frequency-selective surfaces using Krylov-based rational interpolation. *IEEE Transactions on Antennas and Propagation*, 49(5):801–813, 2001.

[153] H. Wendland. *Scattered Data Approximation.* Cambridge, 2005.

[154] K. Willcox and J. Peraire. Application of model order reduction to compressor aeroelastic models. In *Proceedings of ASME International Gas Turbine and Aeroengine Congress*, pages 2000–GT–0377, Munich, Germany, 2000.

[155] K. Willcox and J. Peraire. Application of reduced-order aerodynamic modeling to the analysis of structural uncertainty in bladed disks. In *Proceedings of ASME International Gas Turbine and Aeroengine Congress*, Amsterdam, The Netherlands, 2002.

[156] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.

[157] K. Willcox, J. Peraire, and J. White. An Arnoldi approach for generation of reduced-order models for turbomachinery. *Computers and Fluids*, 31(3):369–389, 2002.

[158] K. Yosida. *Functional Analysis.* Springer-Verlag, Berlin, 1974.

[159] O. Zienkiewicz and R. Taylor. *Finite Element Method: Volume 1. The Basis.* Butterworth-Heinemann, London, 2000.