Reduced Basis Output Bounds for Linear Elasticity: Application to Microtruss Structures

by

Thomas B. Leurent

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

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....
Department of Mechanical Engineering
May 11, 2001

Certified by .....
Anthony T. Patera
Professor
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Reduced Basis Output Bounds for Linear Elasticity: Application to Microtruss Structures

by

Thomas B. Leurent

Submitted to the Department of Mechanical Engineering
on May 11, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

This thesis presents a two-stage offline/online blackbox reduced basis method for the prediction of outputs of interest of elliptic partial differential equations. The methods are more specifically applied to a the microtruss plate problem modeled with linear elasticity. The microtruss plate studied is a simplified version of a cutting-edge structure for engineering applications; the performance is highly dependent on the microarchitecture and optimal designs have to perform output evaluations for many configurations in the parameter space.

The quasi-instantaneous numerical method presented herein is the basis for intensive trade off analysis at a minimal computational cost. The numerical method is not only efficient but also certain: thanks to rigorous a posteriori error bounds, we may retain only the minimum number of modes necessary to achieve the prescribed accuracy in the output of interest. Two reduced-basis output bounds methods are presented.

The $O(Q^2)$ method has very rigorous mathematical foundations that lay the basis for future output bounds methods. We prove that the $O(Q^2)$ method always bounds the error between the reduced-basis ouput and the "truth" space output. Unfortunately, the effectivity of the $O(Q^2)$ output bounds can be very high when applied to linear elasticity problems.

The $O(Q)$ method is then applied to the microtruss plate problem. We prove for the compliant case that the effectivities are bounded from above; they are indeed found to be very consistent. The computational cost of the $O(Q)$ method is also a distinct advantage for the microtruss plate problem where its geometry requires a separation of the approximation domain into many subdomains.

Thesis Supervisor: Anthony T. Patera
Title: Professor

Acknowledgments

After those fantastic two years spent at MIT during which all my expectations have been surpassed, I wish to thank:

- Tony Patera, my advisor, for taking the time to transmit his mathematical knowledge and for his impeccable management.
- Dimitrios Rovas for his assistance on numerous technical points, including the details of the proofs of Chapters 6 and 7. Most of all I would like to thank Dimitri for his friendship that I found invaluable during my time at MIT.
- Christophe Prud'homme for bringing a very valuable computer science expertise to the group — see the actionable equation of Chapter 8. I would also like to thank Christophe for helping me keep my French alive through numerous numerical methods explanations.
- Karen Veroy for taking an interest in the project and pursuing it further.
- Ivon Oliveira and Yuri Solodukhov for completing a great american–grecian–franco–filipinos–brazilero–russian team.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Engineering design problems are in essence multi-parametric problems with a finite number of outputs of interest. Think of a wing in which the structural design comprises an important number of parameters defining its shape and structural behavior, and with a finite number of outputs of interest, such as the maximum deflection under a shear force, or the buckling limit of its internal trusses. The very large parameter space of the problem makes it difficult to try every possible configuration. The problems that can be entirely formulated analytically can have recourse to rigorous optimization methods, but those problems are few. The majority — those problems which must be solved numerically — must rely on a more extensive intuition of the designer in order to compensate for the high cost of computing the outputs for each configuration. Although engineering intuition is the source of many wonders, it rarely leads close to the *optimal* solution of the problem.

However, the latest mathematical and computational tools available put within reach the optimal solution of many multi-criteria design problems. A very promising numerical method to solve partial differential equations is the reduced-basis method — a projection onto low-order approximation spaces comprising solutions of the problem of interest at selected points in the parameter/design space. The reduced-basis methods have a number of impressive features (i) they can be applied on top of a number of different numerical procedures, namely the residual methods (ii) they enjoy an optimality property that ensures rapid convergence even in high-dimensional param-

eter spaces (iii) most importantly, their variable computational cost per additional output evaluation is extremely low, making them very suitable for large quantities of evaluations.

The reduced-basis method is not yet a widespread numerical design tool for real-world applications. One of the issues at stake is the precision of the results. Although *a priori* error theory proves the optimality of the result in the low-order space, the accuracy of the solutions is highly dependent on the number and choice of modes forming the reduced-basis. This thesis extends the study of *a posteriori* output error bounds for the reduced-basis method. We describe a low cost a posteriori error estimation procedure. Furthermore, we decompose both the reduced-basis and the error estimation into an off-line / on-line formulation. The decomposition can be performed for any problem in which the parameters enter affinely in the differential operator.

We choose to apply the reduced-basis output bounds method to a cutting-edge structural mechanics problem. New processes can manufacture cellular materials with a specified microarchitecture; those materials have been proved to perform very well in weight critical applications and have a huge potential for multi-functional applications that combine structural resistance with heat transfer in turbine blades, gas or liquid storage in fuel cells, etc .... The key to the performance of those cellular materials is their topology and geometry. Optimal studies have been performed analytically with beam and plates analysis and reached conclusive results for some outputs. Other outputs require a better model, linear elasticity, that must be solved numerically. The key to an extensive trade-off analysis is then to use the reduced-basis method in conjunction with our a posteriori error estimations.

This thesis contains a variety of new material. Our a posteriori error procedure is applied to linear elasticity for the first time. The complexity of the geometry studied also requires a full geometry affine variation that is applied to (i) the finite element method (ii) the reduced-basis method (iii) the output bounds method. The combination of applying the output bounds method to linear elasticity *and* a complex geometry leads to the development of a more efficient output bounds method. A

"relaxed" error estimation procedure is therefore detailed herein.

Chapter 2 describes the microtruss plate that will be our test case during this study. The microarchitecture properties are explained and the need for an advanced mathematical model is presented. Chapter 3 gives the details of the mathematical model — linear elasticity — and reaches a non-dimensional weak form. A reference-domain formulation is presented in Chapter 4 in order to use a single approximation space. The finite element method is applied in Chapter 5 to the non-dimensional reference-domain weak form reached earlier; some convergence results and a comparison to a commercial FEM code are given. Chapter 6 then introduces the reduced-basis theory, proves the optimality of the solution in the low order space and goes through the details of the off-line / online implementation; some results stating the accuracy and the convergence of the output solutions are given. The a posteriori error estimation procedures are detailed in Chapter 7, including the "relaxed" procedure; we also show results confirming the bounding properties and discussing the effectivity of the bounds. The reader interested in the trade-off analysis can refer to Chapter 8, the only material required is the problem statement of Chapter 2. An interactive equation environment is available with the PDF version of this thesis in order to test any microtruss configuration with the reduced-basis method and the output bounds. Issues that will be considered in future research are given in Chapter 9.

Chapter 2

Microtruss Plate

2.1 Introduction

Cellular materials exhibit property profiles that favor their implementation as multi functional materials [1]. Their use as a core between panels and shells achieves lower structural weights than competing materials. At the same time, these structures provide more flexibility than conventional honeycomb or stringer stiffened structures for their use in heat dissipation devices or devices that store liquid or pressurized gas.

Cellular materials can be categorized as either random micro structures or periodic micro structures. Common forms of the later can be micro-truss assemblies, referred to as lattice materials, or two-dimensional periodic channels, referred to as prismatic materials.

The performance of all cellular materials is highly dependent on their topology and geometry. In particular, once a certain type of microarchitecture has been chosen for the cells, an extensive trade-off analysis is necessary to choose the best geometry.

2.2 Parameters and Properties

2.2.1 Geometry

The microtruss plate geometry shown in Figure 2-1 falls into the prismatic materials category. The structure is designed to provide lightweight support for a large transverse load with minimum deflection. The core consists of thirteen two-dimensional core trusses. Two solid sheet faces on top of and below the core provide impact resistance and isolation. On the side of the microtruss plate are two solid sheet sides.

The microtruss plate is submitted to a constant shear load: the left solid sheet side, $\tilde{\Gamma}_E$, is fixed vertically and horizontally and the right solid sheet side, $\tilde{\Gamma}_N$, is submitted to a vertical load distributed along the whole side.

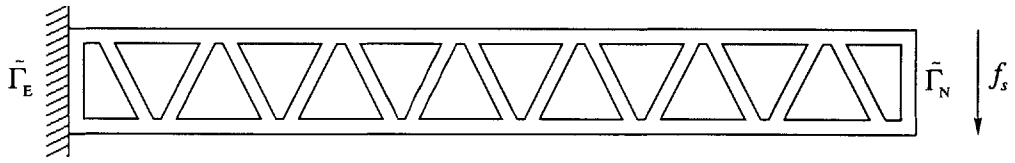

Figure 2-1: Plate structure with core trusses and solid sheet faces. The left end $\tilde{\Gamma}_E$ is fixed and the right end $\tilde{\Gamma}_N$ is submitted to a constant shear force $\tilde{f}_s$.

2.2.2 Design Parameters

The geometry and the material properties of the plate,shown in Figure 2-2, form a six–component parameter vector, or "input," $\mu = (\mu_1, \mu_2, \ldots, \mu_6)$, where $\mu_1 = \alpha$, $\mu_2 = \tilde{t}_{\text{truss}}$, $\mu_3 = \tilde{S}_{\text{y}}$, $\mu_4 = \tilde{t}_{\text{top}}$, $\mu_5 = \tilde{t}_{\text{bot}}$, $\mu_6 = E_{\text{frame}}/E_{\text{truss}}$. Here α is the angle in radians of the core trusses from a transverse position, $\tilde{t}_{\text{truss}}$ is the thickness of the core trusses in the direction parallel to the top and bottom faces, $\tilde{S}_{\text{y}}$ is the spacing between the top and the bottom sheet faces, $\tilde{t}_{\text{top}}$ and $\tilde{t}_{\text{bot}}$ are the thickness of the top and bottom sheet faces, and $E_{\text{frame}}/E_{\text{truss}}$ is the ratio of the trusses Young moduli over the sheets Young moduli. The structure contains 13 trusses; the thickness $\tilde{t}_{\text{side}}$ of the side sheets does not vary; the distance $\tilde{S}_{\text{x}}$ between the midpoints of successive trusses is fixed at

21 times $\tilde{t}_{\text{side}}$; the total length $\tilde{L}$ of the plate is 295 times $\tilde{t}_{\text{side}}$. The Poisson coefficient of the sheets and trusses is fixed at $\nu = 0.2$.

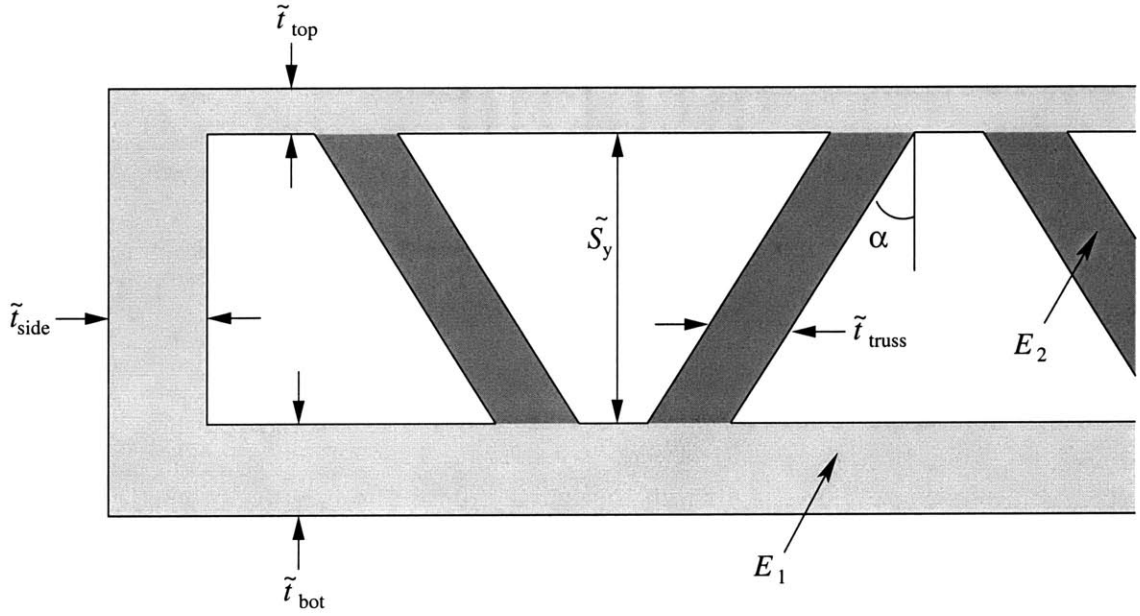

Figure 2-2: Design parameters of the truss plate.

For our parameter space $\mathcal{D}$ we define

$$\mathcal{D}_0 = [0.2, 1.1] \times [0.4, 4.0] \times [4, 60] \times [0.4, 4.0]^2 \times [0.05, 50] \tag{2.1}$$

and we make sure that the vertical spacing $\tilde{S}_{\text{y}}$ is restricted in function of the core trusses angle α and thickness $\tilde{t}_{\text{truss}}$ so that the core trusses do not intersect,

$$\mathcal{D} = \{\mu \in \mathcal{D}_0 \,|\, \tilde{S}_{\text{y}} \leq (\tan\alpha)^{-1}(\tilde{S}_{\text{x}} - \tilde{t}_{\text{truss}} - \tilde{S}_0)\}\ ; \tag{2.2}$$

where $\tilde{S}_0$ – the minimum spacing that we require between the extremities of two contiguous core trusses – is fixed to 0.5.

The acceptable values for a parameter set μ are summed up by the following set

of equations:

$$0.2 \leq \quad \mu_1 \equiv \alpha \quad \leq 1.1 \tag{2.3}$$

$$0.4 \leq \quad \mu_2 \equiv \tilde{t}_{\text{truss}} \quad \leq 4.0 \tag{2.4}$$

$$4.0 \leq \quad \mu_3 \equiv \tilde{S}_{\text{y}} \quad \leq \min\left(60, (\tan\alpha)^{-1}(\tilde{S}_{\text{x}} - \tilde{t}_{\text{truss}} - \tilde{S}_0)\right) \tag{2.5}$$

$$0.4 \leq \quad \mu_4 \equiv \tilde{t}_{\text{top}} \quad \leq 4.0 \tag{2.6}$$

$$0.4 \leq \quad \mu_5 \equiv \tilde{t}_{\text{bot}} \quad \leq 4.0 \tag{2.7}$$

$$0.05 \leq \quad \mu_6 \equiv \frac{E_{\text{truss}}}{E_{\text{frame}}} \quad \leq 50\,. \tag{2.8}$$

2.2.3 Examples of Geometric Parameters

The geometric parameters $\mu_1, \ldots, \mu_5$ define the micro architecture of the truss plate. It is obvious that increasing the thickness or the Young moduli of any of the members will increase the overall stiffness, however design constraints other than minimum weight might have to be considered.

Solid Sheet Face Thickness

Figure 2-3 shows how increasing the top solid sheet face thickness, $\tilde{t}_{\text{top}}$, affects the external geometry by augmenting the total thickness of the microtruss plate. This design parameter does not modify the topology of the structure core.

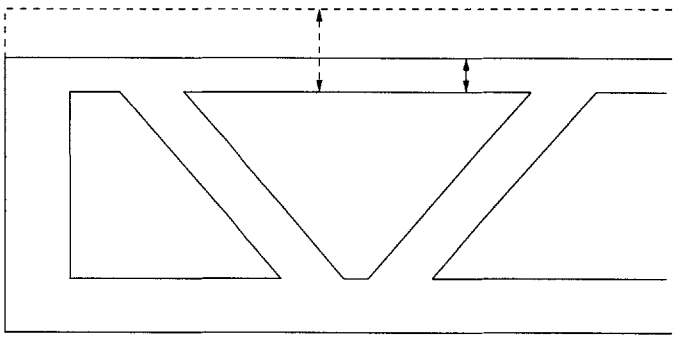

Figure 2-3: Effect of the parameter $\mu_4 \equiv \tilde{t}_{\text{top}}$ on the geometry of the truss plate.

Core Trusses Angles

Figure 2-4 shows how the core trusses angle, α, affects only the geometry of the structure core, leaving the external geometry of the microtruss plate unchanged. In fact, variations of the core trusses angles will only affect the structural properties of the plate, leaving unchanged the amount of intervening space that can be used to enable other functionalities such as heat transfer and gas or liquid storage. [1]

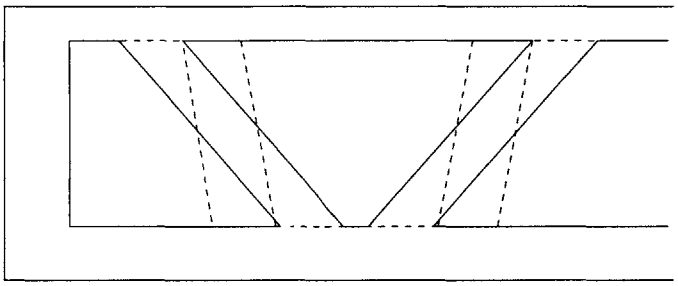

Figure 2-4: Effect of the parameter $\mu_1 \equiv \alpha$ on the geometry of the truss plate.

Vertical Spacing

Figure 2-5 exposes the effect of the vertical spacing, $\tilde{S}_y$, on both the core architecture and the external geometry. An increase in the vertical spacing creates a thicker plate relative to the fixed length $\tilde{L}$ and a more open micro architecture of the core. It is also obvious from Figure 2-5 that the vertical spacing must have an upper bound function of the core trusses angles and thicknesses in order to keep consecutive trusses from intersecting.

[1] Note that a variation in the angle of the core trusses α does not modify the plate volume. The core trusses thickness $\tilde{t}_{truss}$ is the thickness in the direction parallel to the bottom and top side faces; figure 2-2 shows that when α increases and $\tilde{t}_{truss}$ stays constant, the core trusses real thickness decreases, leaving the volume unchanged.

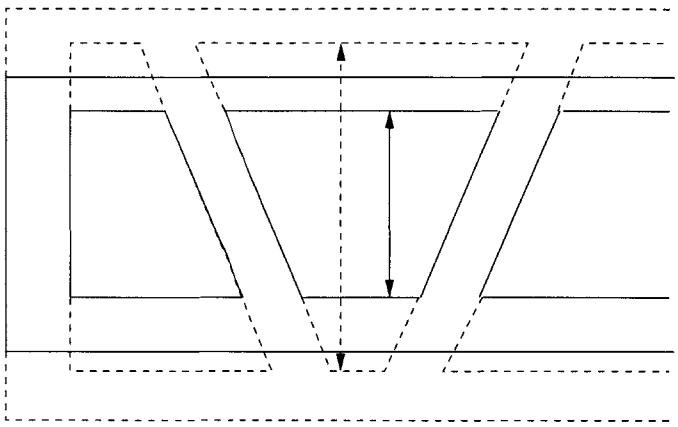

Figure 2-5: Effect of the parameter $\mu_3 \equiv \tilde{S}_y$ on the geometry of the truss plate.

2.3 Quantities of Interest

Some of the usual performance metrics for structures such as the microtruss plate are the maximum moment per unit length, the maximum shear force per unit length and the buckling resistance of the core trusses. Based on those failure modes, [12] shows how truss plates compare well to honeycomb structures.

2.3.1 Outputs Considered

Minimum weight designs can be found by identifying the failure modes of the microtruss plate and then varying the geometry parameters and material properties to determine the lowest weight for some given constraints. For our purposes, the weight is considered directly related to the volume of the microtruss plate and the constraint is a specified tip deflection which cannot be exceeded. In effect, this is equivalent to looking for the geometry that will yield the lowest weight for a given stiffness.

Chapter 9 describes other constraints and failure modes that will be modeled in future work.

Truss Plate Volume

In order to perform a trade off analysis in chapter 8, the total volume of the microtruss plate is required. In effect, given a certain material we wish to work with, the weight of the structure is directly related to the total volume $V(\mu)$ of the microtruss plate geometry described in section 2.2.1

$$V(\mu) = (2 \times \tilde{t}_{\text{side}} \times \tilde{S}_{\text{y}}) + \tilde{L} \times (\tilde{t}_{\text{top}} + \tilde{t}_{\text{bot}}) + 13 \times \tilde{S}_{\text{y}} \times \tilde{t}_{\text{truss}} \ , \tag{2.9}$$

where $\tilde{S}_{\text{y}}, \tilde{t}_{\text{top}}, \tilde{t}_{\text{bot}}, \tilde{t}_{\text{truss}}$ are the design parameters exposed in Section 2.2.2; $\tilde{t}_{\text{side}}$ is the fixed side thickness and $\tilde{L}$ is the total length equal to 295 times $\tilde{t}_{\text{side}}$.

Tip Deflection

Optimal design takes into account various failure modes or constraints relevant to the use of the microtruss plate, however this thesis focuses on a single design constraint – in addition to the volume – in order to lay the foundation for use of the reduced-basis methods in a complete design process.

In many applications, structures must be designed to withstand a specified shear force while minimizing the deflection of the structure. For the microtruss plate case described in Figure 2-1, the shear force is distributed along the right boundary $\tilde{\Gamma}_N$ and can be written as

$$\frac{1}{W} \int_{\tilde{\Gamma}_N} \tilde{f}_s \, dS \ ; \tag{2.10}$$

here W is the total length of $\tilde{\Gamma}_N$, i.e. the width of the plate.

We now define $\ell^O(v)$ as the functional that gives the average displacement along the structure tip $\tilde{\Gamma}_N$, given a displacement field v,

$$\ell^O(v) \equiv \frac{1}{W} \int_{\tilde{\Gamma}_N} v \, dS \ . \tag{2.11}$$

The displacement field in the structure resulting from the shear force $\tilde{f}_s$ is called $\tilde{u}$. The output of interest — the tip deflection — is then $s = \ell^O(\tilde{u})$.

In Chapter 7, we will use the similarity between the weak form of equation (2.10) and the output functional $\ell^O(v)$ to present a simplified version of the output bounds, referred to as the compliant case.

2.4 Physical Model

Various mathematical models are available to predict the structural behavior of the microtruss plate. Truss and beam analysis provides computationally inexpensive results that are used in many truss designs for qualitative purposes. Linear elasticity is a more complex and computationally more expensive modeling tool that will give good quantitative results and use non-restrictive assumptions. Finally, Non-linear elasticity is a model with very realistic assumptions but much more expensive computational requirements.

More specifically, linear elasticity offers a greater accuracy than truss analysis in which the truss joints are idealized as pin joints offering no rotational resistance from core trusses to sheet faces. For that reason, linear elasticity will give better prediction of failure modes such as the buckling resistance of the core trusses which truss analysis underestimates.

When opposed to non-linear elasticity, the linear elasticity model is usually considered valid for shear forces up to 0.5% of the Young modulus and geometric deformations with rotations of up to 10 degrees. For design purposes, plastic deformations are considered to be a failure mode, and the above limitations on shear forces and geometric deformation are incorporated in the constraints of the problem.

Choosing linear elasticity as the physical model allows to obtain good theoretical results for most failure modes while staying within reasonable complexity and cost for the discrete approximation.

2.4.1 Periodicity Properties

For numerical simulations as well as for manufacturing processes, periodicity often constitutes an opportunity to achieve better performance with fewer means. The

microtruss plate has a periodic microarchitecture; each cell consists of two trusses and the parts of the top and bottom sheet faces between the two trusses and before the next truss. The periodic cell is shown in Figure 2-6.

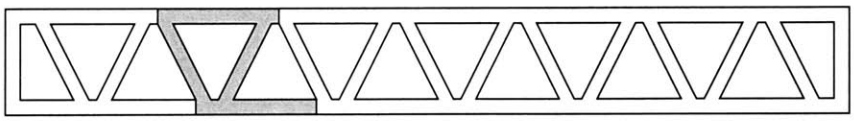

Figure 2-6: The periodic cell of the truss plate geometry.

Chapter 3

Mathematical Model

3.1 Introduction

Linear elasticity has been identified in section 2.4 as the physical model that gives a very good compromise between realistic assumptions for design purposes and moderate complexity for implementation purposes. This chapter describes in details the equations of linear elasticity that are used to model the structural behavior of the microtruss plate submitted to a constant shear force.

The notations and the dimensions used for the different quantities are introduced in Section 3.2.1. The governing equations of linear elasticity are derived from the equilibrium conditions in Section 3.2.2. We manipulate the governing equations to reach the variational formulation in Section 3.2.3. Non-dimensional quantities are introduced in Section 3.2.4 and a non-dimensionalized variational formulation is exposed.

3.2 Equations of Linear Elasticity

3.2.1 Notations for Dimensional Variables

This chapter will first derive the linear elasticity equations in dimensional form and later convert them in non-dimensional form.

Here we introduce the notations used for the dimensional quantities of Section 3.2.2 and 3.2.3. The coordinate in each dimension is $\tilde{x}_i,\ i = 1, 2$; the displacement in each dimension is $\tilde{u}_i,\ i = 1, 2$; the stress tensor is $\tilde{\sigma}_{ij},\ i, j = 1, 2$; the elasticity tensor is $\tilde{E}_{ijkl},\ i, j, k, l = 1, 2$; the Lamé constants are $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$; the shear force along the side sheet $\tilde{\Gamma}_N$ is $\tilde{f}_s$; the traction resulting from the shear force is $\tilde{g}_N^t = \frac{\tilde{f}_s}{W}$, where W is the length of $\tilde{\Gamma}_N$. The dimension units are as follows :

$$[\tilde{u}_j] = [\tilde{x}_j] = [\mathrm{m}] \tag{3.1}$$

$$[\tilde{g}_E^n] = [\tilde{g}_E^t] = [\mathrm{m}] \tag{3.2}$$

$$\left[\tilde{E}_{ijkl}\right] = \left[\tilde{\lambda}_i\right] = [\tilde{\sigma}_{ij}] = \left[\frac{\mathrm{N}}{\mathrm{m}^2}\right] \tag{3.3}$$

$$\left[\tilde{f}_s\right] = \left[\frac{\mathrm{N}}{\mathrm{m}}\right] \tag{3.4}$$

$$[\tilde{g}_N^t] = \left[\frac{\mathrm{N}}{\mathrm{m}^2}\right] . \tag{3.5}$$

3.2.2 Strong Form

The case studied is a two-dimensional microtruss plate $\tilde{\Omega}$ with a homogeneous Dirichlet boundary condition along the left side $\tilde{\Gamma}_E$ and a homogeneous Neumann boundary condition along the right side $\tilde{\Gamma}_N$. The body forces are not accounted for in this derivation. An average deflection s is induced on the structure tip $\tilde{\Gamma}_N$ where the shear force is applied.

Equilibrium Conditions

The corresponding equation is the equilibrium condition of a body $\tilde{\Omega}$ that requires the internal stresses to balance the external force:

$$-\frac{\partial}{\partial \tilde{x}_j}\tilde{\sigma}_{ij}(\tilde{u}) = \tilde{f}_s \text{ , in } \tilde{\Omega} \text{ ;} \tag{3.6}$$

where the indices i, j refer to the spatial coordinates, $\tilde{\sigma}_{ij}$ is the stress tensor, $\tilde{u}$ is the displacement field and $\tilde{f}_s$ is the external force. Note that in equation (3.6), the displacement $\tilde{u}$ is independent of the depth and equal to 0 along x_3. This represents the

plane strain approximation which is the adequate physical model since the microtruss plate has infinite depth.

The stress tensor $\tilde{\sigma}_{ij}$ for a linear elastic isotropic solid is given by

$$\tilde{\sigma}_{ij}(\tilde{u}) = \tilde{\lambda}_2 \left(\frac{\partial \tilde{u}_i}{\partial \tilde{x}_j} + \frac{\partial \tilde{u}_j}{\partial \tilde{x}_i} \right) + \tilde{\lambda}_1 \delta_{ij} \sum_{k=1}^{2} \frac{\partial \tilde{u}_k}{\partial \tilde{x}_k} \, . \tag{3.7}$$

In two dimensions, indices take values from 1 to 2. δ_{ij} is the Kronecker-delta symbol. $\tilde{\lambda}_1 > 0$ and $\tilde{\lambda}_2 > 0$ are the Lamé coeficients; they are related to the Young modulus $\tilde{E}$ and the Poisson coefficient ν by

$$\tilde{\lambda}_1 = \frac{\tilde{E}\nu}{(1+\nu)(1-2\nu)} \tag{3.8}$$

and

$$\tilde{\lambda}_2 = \frac{\tilde{E}}{2(1+\nu)} \quad . \tag{3.9}$$

Note that for an imcompressible solid, we have $\nu \to \frac{1}{2}$, i.e. $\lambda_1 \to \infty$.

For convenience during the implementation, we introduce the elasticity tensor

$$\tilde{E}_{ijkl} = \tilde{\lambda}_2(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \tilde{\lambda}_1 \delta_{ij}\delta_{kl} \, , \quad i,j,k,l = 1,2 \, , \tag{3.10}$$

in terms of which we can then express the stress tensor of equation (3.7) as

$$\tilde{\sigma}_{ij} = \tilde{E}_{ijkl} \frac{\partial \tilde{u}_k}{\partial \tilde{x}_l} \, , \quad i,j,k,l = 1,2 \, , \tag{3.11}$$

with the summation over repeated indices (i.e. the right hand side of (3.11) is a sum of 4 terms: for $k = 1,2$ and for $l = 1,2$).

We can now plug the expression (3.10) for the elasticity tensor into the equilibrium equation (3.6) to obtain the strong form of linear elasticity

$$-\frac{\partial}{\partial \tilde{x}_j} \tilde{E}_{ijkl} \frac{\partial \tilde{u}_k}{\partial \tilde{x}_l} = \tilde{f}_s \, , \text{ in } \tilde{\Omega} \, . \tag{3.12}$$

Boundary Conditions

We study the steady case, therefore there is no initial state condition. We only need to specify the boundary conditions in order to completely determine the problem.

At each point on the boundary, me must specify either the normal displacement or the normal traction:

$$\tilde{u}_i \hat{n}_i = \tilde{g}_E^n \tag{3.13}$$

$$\hat{n}_i \tilde{\sigma}_{ij}(\tilde{u}) \hat{n}_j = \tilde{g}_N^n \tag{3.14}$$

and either the tangential displacement or the tangential traction:

$$\tilde{u}_i \hat{t}_i = \tilde{g}_E^t \tag{3.15}$$

$$\hat{t}_i \tilde{\sigma}_{ij}(u) \hat{n}_j = \tilde{g}_N^t \tag{3.16}$$

where $\hat{n}_i$ and $\hat{t}_i$ are the vectors normal and tangential to the boundary. As exposed in Section 2.2.1, the boundary conditions for the microtruss plate are:

- On the Dirichlet boundary $\tilde{\Gamma}_E$: $\tilde{g}_E^n = \tilde{g}_E^t = 0$, i.e. null displacement in the tangential and in the normal directions.
- On the Neumann boundary $\tilde{\Gamma}_N$: $\tilde{g}_N^t = \frac{\tilde{f}_s}{W}$ and $\tilde{g}_N^n = 0$, i.e. a tangential traction — the shear force $\tilde{f}_s$ divided by the length of the boundary — and a null normal traction.
- On the upper and lower sides: $\tilde{g}_N^t = \tilde{g}_N^n = 0$, i.e. null traction in the tangential and in the normal directions.

3.2.3 Variational Formulation

We have now completely defined the linear elasticity model through mathematic equations. The equations of the previous section are called the strong form of the problem. The point of departure of the finite element method is a transformation of those equations into a new form, called the variationnal formulation (or weak form).

We multiply the strong form equation (3.12) by a test function $\tilde{v}$ that satisfies the essential boundary conditions (3.13) and (3.15). We obtain

$$\int_{\tilde{\Omega}} -\tilde{v}_i \frac{\partial}{\partial \tilde{x}_j} \tilde{E}_{ijkl} \frac{\partial \tilde{u}_k}{\partial \tilde{x}_l} \, d\tilde{V} = \int_{\tilde{\Gamma}_N} \tilde{v}_i \tilde{g}_N^t \, d\tilde{S} \; , \quad \forall \tilde{v} \in \tilde{X} \; , \tag{3.17}$$

where $\tilde{X}$ is the product of two Hilbert spaces $H^1(\tilde{\Omega})$ – since $\tilde{v}$ has two dimensions – with some restrictions to enforce the essential boundary conditions

$$\tilde{X} = \{\tilde{v} \in (H^1(\tilde{\Omega}))^2 | \tilde{v}_i \hat{n}_i|_{\Gamma_E^n} = \tilde{v}_i \hat{t}_i|_{\Gamma_E^t} = 0\} \; . \tag{3.18}$$

The next few steps are described in details in [8]. We integrate the left hand side of equation (3.17) by parts, we apply the divergence theorem and we simplify the integrals with the boundary conditions imposed on the test function $\tilde{v}$ by the space $\tilde{X}$; the result is the weak statement of equation (3.6)

$$\tilde{a}(\tilde{v}, \tilde{u}) = \tilde{\ell}(\tilde{v}) \; , \quad \forall \tilde{v}, \tilde{u} \in \tilde{X}^2 \; , \tag{3.19}$$

with the linear elasticity operator

$$\tilde{a}(\tilde{v}, \tilde{u}) = \int_{\tilde{\Omega}} \frac{\partial \tilde{v}_i}{\partial \tilde{x}_j} \tilde{E}_{ijkl} \frac{\partial \tilde{u}_k}{\partial \tilde{x}_l} d\tilde{V} \; , \tag{3.20}$$

and the Neumann boundary contribution to the right hand side

$$\tilde{\ell}(\tilde{v}) = \int_{\tilde{\Gamma}_N} \tilde{v}_i \tilde{g}_N^t \, d\tilde{S} \; . \tag{3.21}$$

We note that the stress boundary conditions are natural – they have been included in the inhomogeneity (3.21). This will be a great simplification in the next chapter when the finite element approximation is applied to the weak form of the linear elasticity model.

3.2.4 Non-dimensionalization of the Linear Elasticity Equations for the Truss Plate Problem

The results that we will obtain by solving equation (3.19) for the displacement $\tilde{u}$ can be generalized to a wider class of parameters μ if they are expressed in non-dimensional form.

Non-Dimensionalized Quantities

The parameter $\mu_1 \equiv \alpha$ is already a non-dimensional quantity. We non-dimensionalize the geometric parameters $\mu = \{\mu_2, \ldots, \mu_5\}$ with respect to $\tilde{t}_{\text{side}}$. The parameter μ_6 defined by equation (2.8) is already non-dimensionalized with respect to the young modulus of the solid sheets faces and sides, E_{frame}.

We now non-dimensionalize every quantity of section 3.2.3 with respect to $\tilde{t}_{\text{side}}$ and E_{frame}. We use the notation $\bar{y}$ for the non-dimensional version of a quantity $\tilde{y}$. We get

$$\bar{u}_j = \frac{1}{\tilde{t}_{\text{side}}}\tilde{u}_j \tag{3.22}$$

$$\bar{g}_E^{n,t} = \frac{1}{\tilde{t}_{\text{side}}}\tilde{g}_E^{n,t} \tag{3.23}$$

$$\bar{E}_{ijkl} = \frac{1}{E_{\text{frame}}}\tilde{E}_{ijkl} \tag{3.24}$$

$$\bar{\sigma}_{ij} = \frac{1}{E_{\text{frame}}}\tilde{\sigma}_{ij} \tag{3.25}$$

$$\bar{f}_s = \frac{1}{\tilde{t}_{\text{side}} E_{\text{frame}}}\tilde{f}_s \tag{3.26}$$

$$\bar{g}_N^t = \frac{1}{E_{\text{frame}}}\tilde{g}_N^t \,. \tag{3.27}$$

Non-Dimensional Equations

The non-dimensional governing equation is then expressed over a non-dimensional body $\bar{\Omega}$

$$-\frac{\partial}{\partial \bar{x}_j}\bar{\sigma}_{ij}(\bar{u}_i) = \bar{f}_s \quad , \text{in } \bar{\Omega} \,, \tag{3.28}$$

where

$$\bar{\sigma}_{ij} = \bar{E}_{ijkl}\frac{\partial \bar{u}_k}{\partial \bar{x}_l}, \quad i,j,k,l = 1,2\,. \tag{3.29}$$

The boundary conditions have the same form as equations (3.13)- (3.16)

$$\bar{u}_i \hat{n}_i = \bar{g}_E^n \quad \text{on } \bar{\Gamma}_E\,, \tag{3.30}$$

$$\bar{u}_i \hat{t}_i = \bar{g}_E^t \quad \text{on } \bar{\Gamma}_E \tag{3.31}$$

and

$$\hat{t}_i \bar{\sigma}_{ij}(u)\hat{n}_j = \bar{g}_N^t \quad \text{on } \bar{\Gamma}_N\,. \tag{3.32}$$

The approximation space $\bar{X}$ is now defined over the body $\bar{\Omega}$

$$\bar{X} = \{\bar{v} \in (H^1(\bar{\Omega}))^2 | \bar{v}_i \hat{n}_i|_{\bar{\Gamma}_E^n} = \bar{v}_i \hat{t}_i|_{\bar{\Gamma}_E^t} = 0\}\,. \tag{3.33}$$

And we get to a weak form similar to equation (3.19), but using non-dimensionalized variables

$$\bar{a}(\bar{v},\bar{u}) = \bar{\ell}(\bar{v}) \tag{3.34}$$

with the operator

$$\bar{a}(\bar{v},\bar{u}) = \int_{\bar{\Omega}} \frac{\partial \bar{v}_i}{\partial \bar{x}_j}\bar{E}_{ijkl}\frac{\partial \bar{u}_k}{\partial \bar{x}_l} d\bar{V},\ \forall \bar{v},\bar{u} \in \bar{X}^2, \tag{3.35}$$

and the natural boundary contribution to the right hand side

$$\bar{\ell}(\bar{v}) = \int_{\bar{\Gamma}_N} \bar{v}\bar{g}_N^t\, dS\,. \tag{3.36}$$

Calcul of Dimensional Quantities

We first compute the non-dimensional displacement $\bar{u}$ from the weak form (3.34) for a fixed non-dimensional load $\bar{F}_o$, i.e. a non-dimensional traction $\bar{g}_N^t = \bar{F}_0/\bar{W}$. Then we can compute the dimensional displacement for any dimensional load $\tilde{F}_{\text{actual}}$ under the condition that the geometric proportions are the same for the dimensional domain $\tilde{\Omega}$

and for the non-dimensional domain $\bar{\Omega}$. When a total force per unit depth $\tilde{F}_{\text{actual}}$ is given (in N/m), we have the corresponding non-dimensional force

$$\bar{F}_{\text{actual}} = \frac{\tilde{F}_{\text{actual}}}{E_{\text{frame}}\,\tilde{t}_{\text{side}}} \,. \tag{3.37}$$

Using the linearity of operator (3.34), the corresponding non-dimensional displacement is

$$\bar{u}_{\text{actual}} = \frac{\bar{F}_{\text{actual}}}{\bar{F}_0} = \frac{\tilde{F}_{\text{actual}}}{E_{\text{frame}}\,\tilde{t}_{\text{side}}\,\bar{F}_0}\,\bar{u} \,. \tag{3.38}$$

Remember that the displacement has been non-dimensionalized with respect to $\tilde{t}_{\text{side}}$, so the dimensional displacement is

$$\tilde{u}_{\text{actual}} = \bar{u}_{\text{actual}}\,\tilde{t}_{\text{side}} = \frac{\tilde{F}_{\text{actual}}}{E_{\text{frame}}\,\tilde{t}_{\text{side}}\,\bar{F}_0}\,\bar{u}\,\tilde{t}_{\text{side}} = \frac{\tilde{F}_{\text{actual}}}{E_{\text{frame}}\,\bar{F}_0}\,\bar{u} \,. \tag{3.39}$$

We simplify the re-dimensionalization by taking $\bar{F}_0 = 1$ during the implementation, therefore the dimensional displacement is given by

$$\tilde{u}_{\text{actual}} = \frac{\tilde{F}_{\text{actual}}}{E_{\text{frame}}}\,\bar{u} \,. \tag{3.40}$$

Chapter 4

A Reference-Domain Formulation

4.1 Motivation

Until now, we have assumed that the domain $\bar{\Omega}$ – over which the linear elasticity form (3.19) is applied – has exactly the same geometric proportions as the microtruss plate we want to model. This is actually a very restrictive assumption since it means that for each new geometry, we need to define a new space $\tilde{X}$ in which the linear elasticity model is applied.

It does not have to be that way. The microtruss geometry allows to use a *single approximation space* in conjunction with *piecewise continuous affine mappings* to model all possibles configurations. We call this the reference-domain formulation. One can already devise that this formulation would provide a much more flexible implementation of the model. Indeed, in the context of the discrete approximation of chapter 5, a new set of geometric parameters is then just another input; it does not require the costly discretization process of a new space.

Furthermore, the reduced-basis approximation exposed in chapter 6 cannot be applied over multiple approximation spaces. Therefore, we will use the reference-domain formulation to obtain the displacement field for any configurations of the microtruss.

We start this chapter by giving an example of affine mapping in Section 4.2. The mechanics of piecewise continuous affine mappings are presented in section 4.3. The

particular reference domain used for the microtruss plate architecture is explained in section 4.4. The piecewise continuous affine mapping method is applied to the linear elasticity weak form in section 4.5.

4.1.1 Notations for the Reference Domain and the Mapped Domain

This time we use two different notations to differentiate the (non-dimensional) variables $\bar{x}_i, \bar{u}_i, \bar{E}_{ijkl}, \bar{\sigma}_{ij}, \bar{f}_i$ in the *mapped* domain $\bar{\Omega}$ and the (non-dimensional) variables $x_i, u_i, E_{ijkl}, \sigma_{ij}, f_i$ in the *reference* domain Ω that we will introduce shortly.

4.2 Example of Affine Mapping

As an introductory example, we show the affine mapping used for a truss in the microtruss plate. Consider the domain $\bar{\omega}$ representing a truss with an angle that we want to map to the "reference-domain" ω representing a vertical truss, as shown in Figure 4-1.

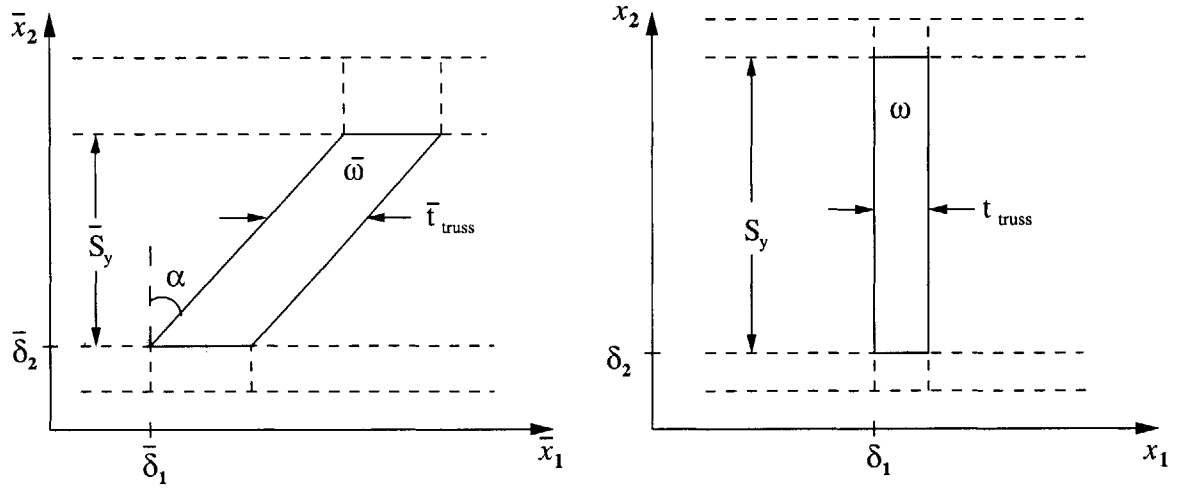

Figure 4-1: Affine mapping of a core truss to the reference domain.

The affine mapping from the deformed domain $\bar{\omega}$ to the reference domain ω is

given by

$$x_1 = \underbrace{\frac{t_{\text{truss}}}{\bar{t}_{\text{truss}}}}_{\alpha_{11}} \bar{x}_1 - \underbrace{\frac{t_{\text{truss}}}{\bar{t}_{\text{truss}}}(\tan\alpha)}_{\alpha_{12}} \bar{x}_2 + \underbrace{(\delta_1 - \bar{\delta}_1)}_{\beta_1}$$

$$x_2 = \underbrace{\frac{S_{\text{y}}}{\bar{S}_{\text{y}}}}_{\alpha_{22}} \bar{x}_2 + \underbrace{(\delta_2 - \bar{\delta}_2)}_{\beta_2} .$$

Note that the coefficients $\alpha_{ij},\ i,j = 1,2$ define the *linear* part of the mapping whereas β_1 and β_2 are constants, part of the *affine* mapping.

Using affine mapping, we can transform easily parallelograms, including some angle effects as in the truss case. This will b every appropriate for the truss plate problem. More generally, affine mappings can transform triangles and therefore be used for a very wide range of geometries.

4.3 Piecewise Continuous Affine Mapping

We consider a piecewise continuous affine mapping G from a deformed domain $\bar{\Omega}$ to a reference domain Ω. The mapping G and its inverse G^{-1} are shown in Figure 4-3.

Each region of the deformed domain $\bar{\Omega}$ has to be submitted to a different affine mapping

$$x_1 = \alpha_{11}\bar{x}_1 + \alpha_{12}\bar{x}_2 + \beta_1 \tag{4.1}$$

$$x_2 = \alpha_{21}\bar{x}_1 + \alpha_{22}\bar{x}_2 + \beta_2 , \tag{4.2}$$

that transforms it into the corresponding region in the reference domain Ω. Although the constants β_1 and β_2 are different for each region, the linear transformation coefficients $\alpha_{ij},\ i,j = 1,2$ are in fact the same for sets of regions. For example, all core trusses inclined to the right have the same *linear* mapping. We call a *parameter region* a set of regions that have the same linear mapping, i.e. the same linear coefficients $\alpha_{ij},\ i,j = 1,2$ in their affine mapping. The total number of parameter regions is denoted R. The parameter regions are presented in more details in Section 4.4.

For each parameter region, the associated linear mapping is $\mathcal{L}^r$, $r = 1, \ldots, R$ defined by

$$x_1 = \alpha^r_{11}\bar{x}_1 + \alpha^r_{12}\bar{x}_2 \tag{4.3}$$

$$x_2 = \alpha^r_{21}\bar{x}_1 + \alpha^r_{22}\bar{x}_2 \,, \tag{4.4}$$

which has the equivalent matrix form

$$x_i = \mathcal{L}^r_{ij}\bar{x}_j \,, \quad i,j = 1,2 \,, \quad r = 1, \ldots, R \,, \tag{4.5}$$

with summation over the repeated index $j = 1, 2$. From the previous equation, we see that for each parameter region, the matrix $\mathcal{L}^r$ is given by

$$\mathcal{L}^r = \begin{bmatrix} \frac{\partial x_1}{\partial \bar{x}_1} & \frac{\partial x_1}{\partial \bar{x}_2} \\ \frac{\partial x_2}{\partial \bar{x}_1} & \frac{\partial x_2}{\partial \bar{x}_2} \end{bmatrix} \tag{4.6}$$

Note that the matrices $\mathcal{L}^r$ contain informations about the geometry of $\bar{\Omega}$ and are therefore a function of the parameter vector μ.

4.4 Reference Domain and Parameter Regions

Figure 4-2 shows the reference domain we chose to use to model the microtruss plate described in section 2.2; it corresponds to $\mu = \{0, 1, 15, 1, 1, \mu_6\}$. It has of course composed of the same topology as the microtruss plate: thirteen core trusses, two faces and two sides. As mentioned earlier, the ratio of the side thickness to the total

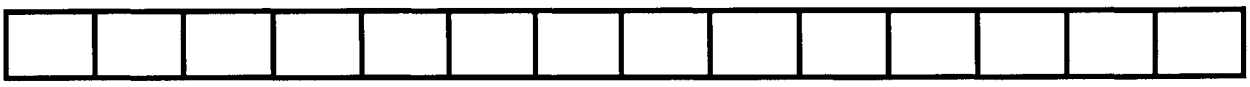

Figure 4-2: Reference domain Ω of the microtruss plate.

length is kept fixed to 1/295 through the transformation G. All other geometric

quantities defined in the parameter vector μ are allowed to vary through G.[1]

The transformation G from the mapped domain to the reference domain is shown in Figure 4-3 for a microtruss structure with five core trusses. Observing the figure

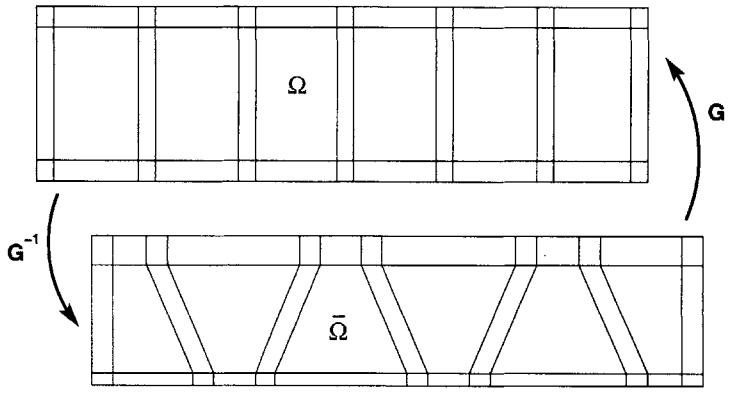

Figure 4-3: Transormation G from a deformed domain $\bar{\Omega}$ to a reference domain Ω.

reveals how we must use different linear mappings $\mathcal{L}^r$ for different regions of the structure. It is also obvious that a number of regions can benefit from the same linear mapping, e.g. all trusses inclined in the same way.

We have defined a parameter region as the union of all the regions of the reference domain Ω that use the same linear mapping $\mathcal{L}^r$ to obtain the required deformed domain $\bar{\Omega}$. There are a total of $R = 15$ different parameter regions in the microtruss plate problem of section 2.2; those are shown in figure 4-4.

4.5 Mapping Applied to the Bi-Linear Elasticity Form

The bi-linear form (3.35) can be decomposed in a sum over all the parameter regions

$$\bar{a}(\bar{u},\bar{v}) = \sum_{r=1}^{R} \int_{\bar{\Omega}^r} \frac{\partial \bar{u}_i}{\partial \bar{x}_j} \bar{E}^r_{ijkl} \frac{\partial \bar{v}_k}{\partial \bar{x}_l} d\bar{V}, \; \forall \bar{u},\bar{v} \in \bar{X}^2 \; . \tag{4.7}$$

[1]Note that although the structure members shown in Figure 4-2 have a high aspect ratio, we will still model them as members with a finite thickness.

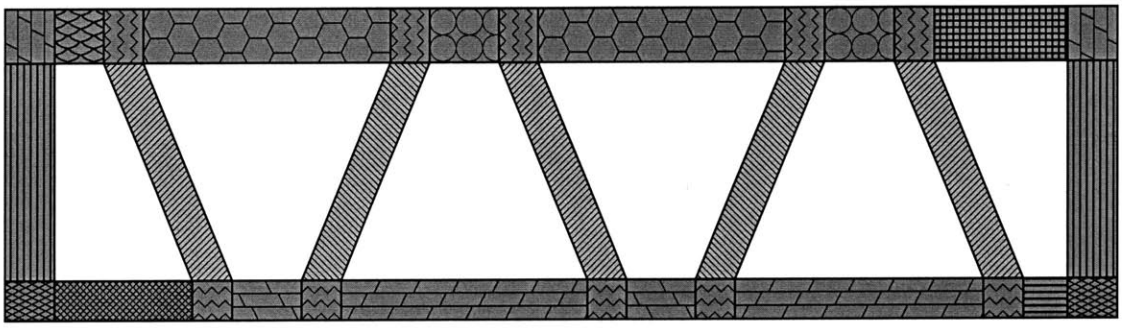

Figure 4-4: Parameter regions of a truss plate with two periods. The figure shows each of the 15 parameter regions with a different pattern

We then use the piecewise continuous affine mapping G to rewrite the bi-linear form over the reference domain Ω

$$a(u, v; \mu) = \sum_{r=1}^{R} \int_{\Omega^r} \frac{\partial u_i}{\partial x_j} \frac{\partial x_j}{\partial \bar{x}_{j'}} \bar{E}^r_{ij'kl'} \frac{\partial v_k}{\partial x_l} \frac{\partial x_l}{\partial \bar{x}_{l'}} J^r(\mu)\, dV \ , \quad \forall v, u \in X^2 \ , \tag{4.8}$$

where $J^r(\mu)$ is the Jacobian of the transformation $(\mathcal{L}^r)^{-1}$ and the approximation space is now defined as

$$X = \{v \in (H^1(\Omega))^2 | v_i \hat{n}_i|_{\Gamma^n_E} = v_i \hat{t}_i|_{\Gamma^t_E} = 0\} \ . \tag{4.9}$$

Now we introduce for each parameter region a "modified" elasticity tensor that includes the previous elasticity tensor (3.24) and the inverse linear mapping (4.6)

$$E^r_{ijkl}(\mu) = J^r(\mu) \mathcal{L}^r_{jj'} \bar{E}^r_{ij'kl'} \mathcal{L}^r_{ll'} \ , \quad r = 1, \ldots, R \ . \tag{4.10}$$

It is essential to note that the "modified" elasticity tensor $E^r_{ijkl}(\mu)$ now contains all the information of the input vector μ. Indeed the elasticity tensor $\bar{E}^r_{ij'kl'}$ contains the material properties of the microtruss plate, and $\mathcal{L}^r$ contains the geometric properties.

Finally, we can rewrite the weak form of linear elasticity over the reference domain Ω,

$$a(u, v; \mu) = \sum_{r=1}^{R} \int_{\Omega^r} \frac{\partial u_i}{\partial x_j} E^r_{ijkl}(\mu) \frac{\partial v_k}{\partial x_l}\, dV \ , \quad \forall v, u \in X \ . \tag{4.11}$$

The operator $a(u, v; \mu)$ has the property that the parameter set μ only appears in the modified elasticity tensor. This will allow us to perform in section 6.1.2 an affine decomposition of great importance for the reduced-basis method.

Chapter 5

Review of Finite Element Method for Linear Elasticity

5.1 Introduction

In Chapter 3, we have described linear elasticity, i.e. the mathematical model to study the structural behavior of the microtruss. We have expanded the linear elasticity model to a reference-domain formulation in Chapter 4, in order to use a single approximation space for all the configurations studied. We now need to choose a numerical method in order to compute the discrete solutions of the reference-domain formulation of linear elasticity.

Finite element analysis is by far the most established numerical method in structural mechanics, making it our prime choice. The reader will find in this chapter a detailed description of the *implementation* of the Finite Element Method for linear elasticity. Very little of the FEM *theory* is treated here; the reader should then refer to appropriates books and publication such as [8, 4].

Section 5.2 goes through the finite element method implementation used throughout this work. The results are presented and discussed in Section 5.3.

5.2 FEM applied to Linear Elasticity theory

Reaching the weak form (3.19) of the linear elasticity equations was in fact the first step of the finite element method. We now describe the proper two-dimensional implementation of the Galerkin method starting from the reference-domain weak form (4.11).

5.2.1 Abstract FEM Framework

The domain is divided in triangular elements forming a non-structured mesh called the discretized reference domain Ω_h. The number of nodes is $\mathcal{N}/2$. In a 2D elasticity problem, the discrete solution (displacement) at each node has 2 degrees of freedom. To refer to those components, we will use the indices $i, j = 1, 2$. The total number of nodes-components in our discretization is then $\mathcal{N}$.

Reference Element

The reference element $\hat{T}$ is the usual triangular first order element:

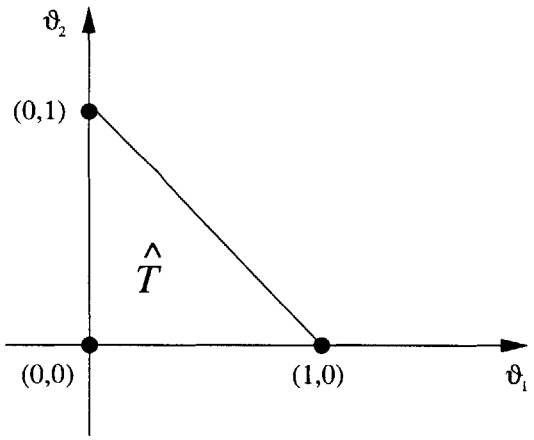

Figure 5-1: Reference element $\hat{T}$.

The basis functions associated to this element are H_1, H_2, H_3

$$H_1(\vartheta) = \vartheta_1 \,, \tag{5.1}$$

$$H_2(\vartheta) = \vartheta_2 \,, \tag{5.2}$$

$$H_3(\vartheta) = 1 - \vartheta_1 - \vartheta_2 \,. \tag{5.3}$$

In equation (5.13), we will have to compute the integral of a product of basis functions derivatives. It is straightforward to compute the derivatives in our reference element space:

$$D_q H_m \equiv \frac{\partial H_m}{\partial \vartheta_q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \,, \tag{5.4}$$

as well as the integral of their product, giving the result

$$B_{qmq'n} \equiv \int_{\hat{T}} \frac{\partial H_m}{\partial \vartheta_q} \frac{\partial H_n}{\partial \vartheta'_q} = \frac{1}{2}(D_q H_m)(D_{q'} H_n) \qquad m,n = 1,2,3 \quad q,q' = 1,2 \,; \tag{5.5}$$

where $D_q H_m$ is simply the entry (q,m) of the matrix (5.4).

Affine Mapping

The affine mapping F_k is the transformation from the reference element $\hat{T}$ to the mesh element T_k. We call x_n^k the coordinate of the node n of element k

$$x_n^k = F_k(\vartheta_n) \,, \quad n = 1,2,3 \,. \tag{5.6}$$

During the implementation, we will use the inverse affine mapping F_k^{-1}: that transformation from the mesh element to the reference element can be expressed in matrix form $\mathbb{T}_k^{-1}$

$$\mathbb{T}_k^{-1} = \frac{1}{2\text{Area}(T^k)} \begin{bmatrix} (y_2^k - y_3^k) & (x_3^k - x_2^k) \\ (y_3^k - y_1^k) & (x_1^k - x_3^k) \end{bmatrix} \tag{5.7}$$

Basis Functions

We can now introduce the basis functions of the discretized reference domain. A global basis function restricted to element k is:

$$\varphi_{k,m|T^k} = H_m \circ F_k^{-1} \tag{5.8}$$

The expression above describes a basis function restricted to an element: i.e. it has a local support. The basis function $\varphi_{n,i}$ we will use also has a local support, but it is defined by a node number n and a component number i. The basis function $\varphi_{n,i}$ is in fact the hat function, i.e. the sum of all basis functions $\varphi_{k,m|T^k}$ non-null at the global node n; there are two basis function per node, one for each component.

The span of the basis functions defines our finite element approximation space

$$X_h = \text{span}\{\varphi_{n,i},\ i = 1, 2,\ n = 1, \ldots, \mathcal{N}/2\}\ . \tag{5.9}$$

The dimension of X_h is thus $\mathcal{N}$.[1]

Galerkin Method

To obtain the discrete form of the operator $a(.,.;\mu)$ given by equation (4.11), we write the discretized solution u_h as a weighted sum of basis functions

$$u_h = \sum_{n=1}^{\mathcal{N}/2} \sum_{i=1}^{2} u_{n,i}\, \varphi_{n,i}\ ; \tag{5.10}$$

where $u_{n,i}$ is the coefficient of the basis function $\varphi_{n,i}$ corresponding to node n and component i. In the Galerkin method, the test functions of the operator $a(.,.;\mu)$ are chosen equal to the basis functions

$$v_{m,j} = \varphi_{m,j}\ , \quad j = 1, 2,\ m = 1, \ldots, \mathcal{N}/2\ . \tag{5.11}$$

[1] The essential boundary conditions must be enforced in the approximation space, so X_h does not contains the nodes-components submitted to Dirichlet boundary conditions.

We substitute (5.10) and (5.11) in the weak form (4.11):

$$\sum_{n=1}^{\mathcal{N}/2}\sum_{i=1}^{2} a(\varphi_{m,j},\varphi_{n,i};\mu)\,u_{n,i} = \ell(\varphi_{m,j}),\quad \forall\, j\in\{1,2\},\ m\in\{1,\ldots,\mathcal{N}/2\}\,. \tag{5.12}$$

The solution vector is then $u(\mu)\equiv u_{n,i}$ which has dimension $\mathcal{N}$. The expanded form of the right hand side operator is

$$\sum_{n=1}^{\mathcal{N}/2}\sum_{i=1}^{2} a(\varphi_{m,j},\varphi_{n,i};\mu) = \sum_{n=1}^{\mathcal{N}/2}\sum_{i=1}^{2}\left(\sum_{r=1}^{R}\int_{\Omega^r}\frac{\partial\varphi_{m,j}}{\partial x_{k'}}E^r_{ik'jk}(\mu)\frac{\partial\varphi_{n,i}}{\partial x_k}dV\right)\,, \tag{5.13}$$

and the expanded form of the left hand side inhomogeneity is

$$\ell(\varphi_m) = \int_{\Gamma^n_N}\varphi_m\hat{n}_i g^n_N dS + \int_{\Gamma^t_N}\varphi_m\hat{t}_i g^t_N dS\,. \tag{5.14}$$

The indices i,j refer to the components and the indices m,n refer to the nodes in the mesh.

5.2.2 Implementation Considerations

The Galerkin result (5.12) needs to be discretized in order to form the finite element matrix. The quantities needed for the implementation have been defined. The details of the discretization process are given in [8] and also numerous other publications.

Index Mappings

One last concept must be introduced: in the matrix form each node-component of an element will correspond to a line in the global matrix. To get to this form, we define two index mappings. The first one, $\theta_{(n,i)}$, merges the nodes indices and the component indices into a single index and wil be used in the elemental matrices. The second mapping uses $\theta_{(n,i)}$ and the element index k to associate a line number in the global matrix to a particular combination of element, local node and component. This local to global component-node mapping is called $\theta(k,n,i)$.

Mesh Generation

We use a mesh over the reference domain Ω to perform the computations. Each parameter region must be the union of whole elements; Figure 5-2 shows this at the base of the truss where the placement of the elements clearly defines a rectangle.[2]

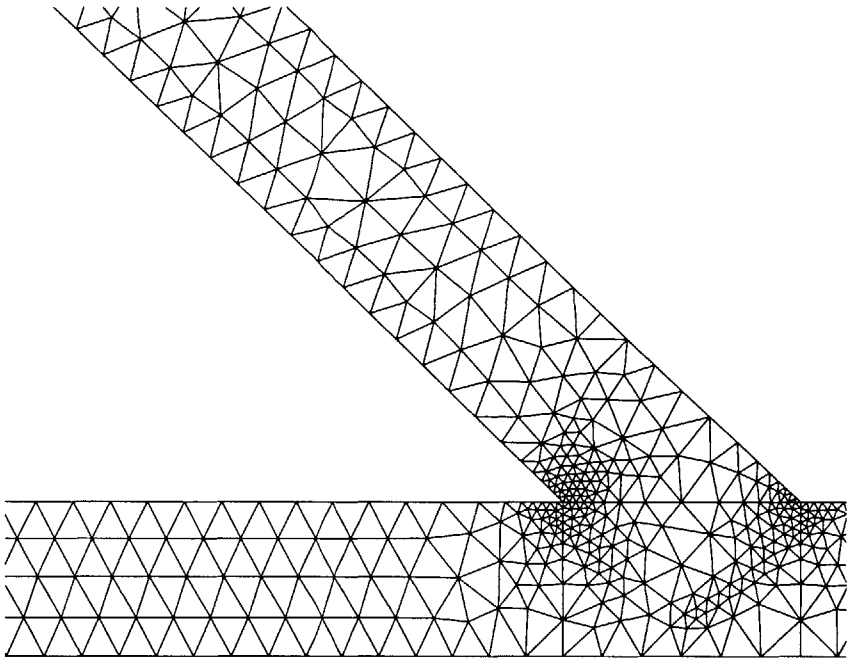

Figure 5-2: Detail of the mesh. The regions are constituted of whole elements; observe the rectangle defined by the elements under the truss base.

The use of the reference-domain procedure to compute results for a mapped domain $\bar{\Omega}$ from the mesh over Ω allows to compute results for a many geometries from a single discretization. The drawback is that we expect an over-distortion of the elements when some regions have a very different aspect ratio in $\bar{\Omega}$ and Ω.

Elemental Quantities

Now that all required quantities have been introduced, the implemented form of equation (4.11) is given under the form of the so called elemental stiffness matrix. Its size is the number of nodes-components of an element. Our 2D first order triangular

[2]Note that the figure shows the result on the deformed domain $\bar{\Omega}$.

elements have three nodes with 2 components each, resulting in a 6×6 elemental matrix.

Following the procedure described in [8], the final result for the stiffness matrix of an element in the parameter region r ($r = 1, \ldots, R$) is given by

$$(A^k)_{\theta_{(n,i)}\,\theta_{(m,j)}} = 2\text{Area}(T_h^k) E^r_{iljl'}(\mu) \sum_{q=1}^{2} (\mathbb{T}_k^{-1})_{ql} \sum_{q'=1}^{2} (\mathbb{T}_k^{-1})_{q'l'} B_{qmq'n} \ , \quad m,n = 1,...,3 \ , \ i,j = 1,2 \ . \tag{5.15}$$

In equation (5.15), the modified elasticity tensor $E^r_{iljl'}$ is given by equation (4.10). Remember that $E^r_{iljl'}$ depends on which parameter region r the element k belongs to. [3]

Assembly

For each element, the elemental matrix $\underline{A}^k$ is computed and it is used to fill up the finite element global matrix $\underline{A}^h$ through the usual "direct stiffness" assembly algorithm.

```
// loop over all the elements
for ( k = 1; k <= K; k++) {
  // loop over the nodes of the element
  for (m = 1; m <= 3; m++) {
    // loop over the components of the node
    for (j = 1; j <= 2; i++) {
      // retrieve the position in the global matrix
      int l = teta(k,m,j);
      // idem as previous loops
      for (int n = 1; n <= 3; n++) {
        for (i = 1; i <= 2; i++) {
          int c = teta(k,n,i);
```

[3] Also we observe, like in Section 4.5, that $E^r_{iljl'}$ is the only quantity in equation (5.15) that includes the information from the input vector μ.

```
            // put the entry in the main matrix
            A(l,c) = A(l,c) + A^k( teta(m,j), teta(n,i) );
}}}}}
```

Boundary Conditions

For the right hand side, we give the implemented result of a Neumann boundary condition, i.e. the inhomogeneity value for a global node-component $\theta(k,n,j)$ on the boundary,

$$F^h_{\theta(k,n,j)} = g^j_N h_{\theta(k,n,*)} \; ; \tag{5.16}$$

where we define $h_{\theta(k,n,*)}$ as the length of the boundary covered by the node n of element k (mean distance between that node and the two contiguous nodes) divided by the total length of the boundary. We will denote $\underline{F}^h \in I\!R^{\mathcal{N}}$ the vector of nodal values of the inhomogeneity, which entries are the $F^h_{\theta(k,n,j)}$ coefficients.

Other equations would be necessary to take into account the other types of inhomogeneities such as the body forces included in (5.14), or Robin boundary conditions.

5.3 Results and Discussion

The finite element code written during this thesis is named EBELAS (Error Bounds for ELASticity). An example of the displacement of a truss plate is given in Figure 5-3. Note, the figure does not show that the computations are actually performed on the reference-domain mesh, which has a different geometry.

In order to validate the numerical results obtained by EBELAS, we give some quick convergence rate results – as opposed to an in depth convergence rate study involving the error in the energy norm of the finite element matrix – and we then compare the results from EBELAS to results from ADINA, a commercial finite element code.

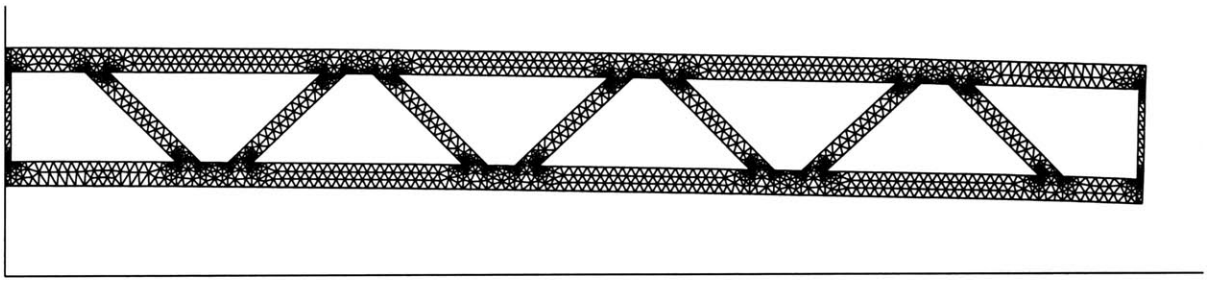

Figure 5-3: FEM approximation of Linear Elasticity in a 2D Truss plate.

5.3.1 Convergence Rate

The convergence results are computed with the finite element method presented hitherto, including of course the reference-domain formulation. The truss plate geometry used for this test is similar to Figure 4-2 but comprises only three core trusses. The discretization is refined at the singularities.

We consider several discretizations, each an exact refinement of the previous one, as shown in figure 5-4. If we define h as the longest element side in a mesh, then the longest element side in the exactly refined mesh is $h/2$.

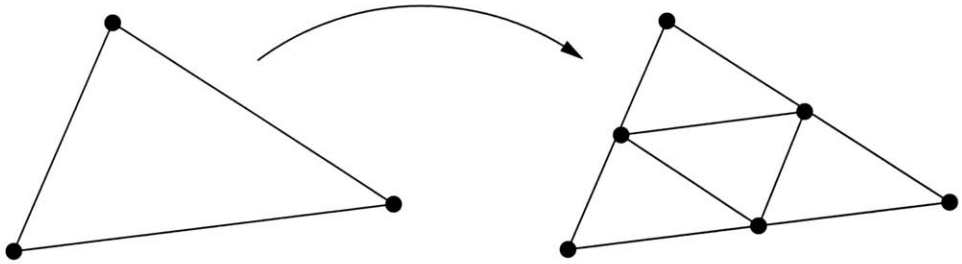

Figure 5-4: Exact refinement of a linear triangular element.

The convergence rate for our output s (the deflection at the tip Γ_N) is defined as p satisfying

$$|s^h - s^{exact}| = Ch^p \; ; \tag{5.17}$$

where s^{exact} is the "true" value of the output and s^h is the value of the output for a certain discretization Ω^h with longest element side h. We can infer the convergence

rate from

$$p = \frac{\log\left(|s^h - s^{exact}|\right)}{\log(h)} \,. \tag{5.18}$$

Approximating s^{exact} by the output obtained with our finest mesh, we can now plot the logarithm of the error $|s^h - s^{exact}|$ in function of the logarithm of the mesh size h. The slope of the curve (fig. 5-5) is then a good approximation for our convergence rate. We obtain $p = 1.85$. Since $u^h \in H^2(\Omega^h)$, we would expect the convergence to quadratic, i.e. $p \geq 2$. One must consider that the many singularities present in the microtruss plate geometry, in conjunction with the over-distortion of some elements (particularly the core trusses elements) can lead to a slight deterioration of the convergence properties.

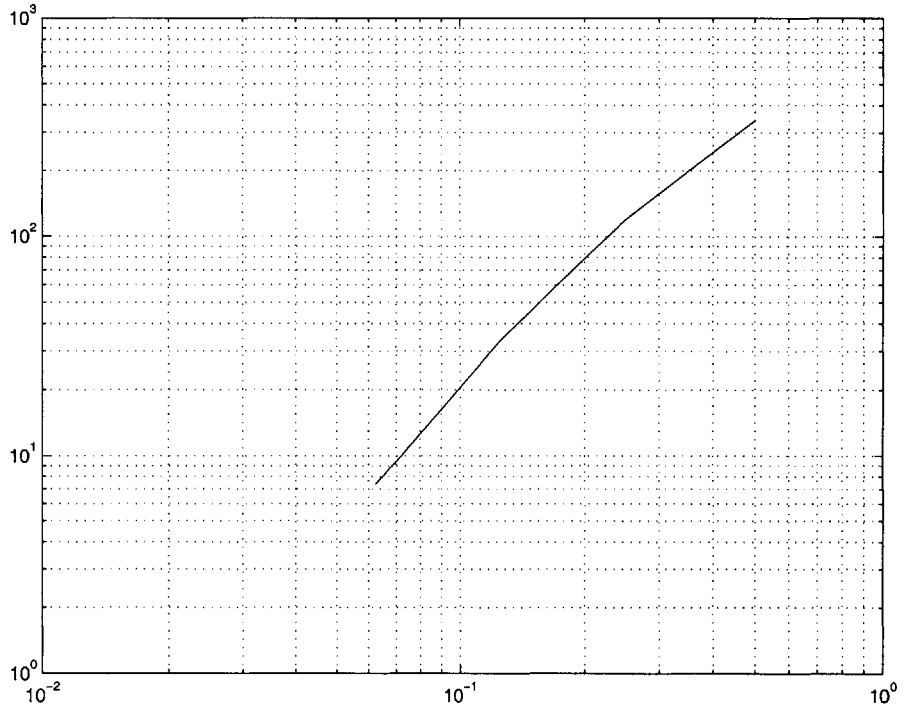

Figure 5-5: Convergence of the finite element code for a truss plate geometry with one periodic set

5.3.2 Comparison with ADINA

The convergence rate study of Section 5.3.1 gives an indication of the accuracy of our solution in function of the discretization refinement. We use a commercial finite element code, ADINA, to *validate* the results obtained by EBELAS.

The results are computed for the truss plate that we will study in the trade-off analysis of Chapter 8, i.e. a truss plate with 13 core trusses. The input vector is $\mu^{\text{test}} = \{0.49, 1, 15, 4.5, 3.5, 1.5\}$. These parameters imply some moderate distortion of the elements through the reference-domain formulation.[4]

Quadratic quadrangular elements are used and we configure ADINA to use the same linear elasticity plane strain approximation as we use in Chapter 3. The results are shown in Table 5.1. [5]

h	$s^h(\mu^{\text{test}})$
0.5	11.156
0.25	11.164
0.125	11.168

Table 5.1: ADINA – Tip deflection in function of the mesh size for quadratic quadrangular elements in a mesh not refined locally.

The results for the same test case computed with EBELAS are shown in Table 5.2. [6] Linear triangular elements are used and the mesh is not refined around the singularities in order not to include refinement issues in the comparison. Note that EBELAS uses a discretization of the reference-domain Ω in conjunction with a piecewise continuous affine mapping (see Chap. 4), whereas ADINA computes the result on a discretization of the actual domain $\bar{\Omega}$.

We see that ADINA converges to a value of about 11.17 for $s^h(\mu^{\text{test}})$. The results given by EBELAS are within 2% of ADINA for sufficiently fine discretizations. The

[4] A more complete study of the effects of over-distortion will be performed in [11].

[5] In fact, we need to give dimensional quantities to ADINA: the side thickness is set to 1 mm, the truss Young modulus is 10^{11}N/m^2 and the load per unit depth is 10^4N/m. Others quantities can be derived from μ^{test}. The results are given in mm.

[6] The non-dimensional parameters μ^{test} are given as an input. The result is dimensionalized as exposed in Section 3.2.4 and given in mm in the table.

h	$s^h(\mu^{\text{test}})$
0.5	10.373
0.25	10.846
0.125	11.103

Table 5.2: EBELAS – Tip deflection in function of the mesh size for linear triangular elements in a mesh not refined locally.

difference is due to the use of a superior type of elements in ADINA, namely the quadratic quadrangular elements. This issue might be addressed in future work, but at the first order triangular elements give an accuracy good enough for our computations to be realistic.

Furthermore, an adequate refinement of the mesh around the singularities will drastically improve the accuracy obtained with a first-order elements discretization, as shown in Table 5.3.

h	$s^h(\mu^{\text{test}})$
0.5	10.679
0.25	11.141
0.125	11.312

Table 5.3: EBELAS – Tip deflection in function of the mesh size for linear triangular elements in a mesh refined around the singularities. h is the length of the longest edge in the mesh.

Note that, in table 5.3, the mesh with $h = 0.25$ refined around the singularities will be used later in this thesis when we refer to the microtruss plate computations.

We now wish to put more emphasis on the use of the reduced-basis method to achieve very efficient computations of the output of interest.

Chapter 6

Reduced-Basis Approximations

The computations performed in chapter 5 to obtain the field of displacement in the truss plate are fairly expensive. By today's standard in computing power, simulation of structural behavior with finite element analysis is much more efficient than experimental approaches, but still not efficient enough to optimize the design of a structure with a large number of parameters.

The reduced-basis approximations presented in this chapter are very efficient methods for the prediction of linear functional outputs dependent on a large number of design parameters. In general, the reduced-basis method can be applied to any weighted-residual approximation; in this work, it is applied to the Galerkin approximation described in chapter 5.

The implementation of the reduced-basis method can be separated into two stages. The *off-line* stage computes parameter-independent quantities in the high-dimensional "truth" space and project those quantities onto a low-order space. The *on-line* stage uses the low-order parameter-independent operators from the on-line stage to reconstruct a low order operator for each evaluation. This approach is referred to as the *black-box* approach.

We will apply the reduced-basis method to an operator which properties are enumerated in Section 6.1. The quality and low computational cost of the reduced-basis approximation stems from a projection onto a "problem-specific" low order space described in section 6.2. These techniques enjoy an optimality property which ensures

rapid convergence even in high-dimensional parameter spaces; this is addressed in section 6.3.

6.1 Operator Properties

The elasticity operator enjoys certain properties that are exploited in our implementation of the reduced-basis and output bounds methods; those properties are summarized in the present section. References for extensions to more general problem are supplied.

6.1.1 General Properties

The most obvious properties of the linear elasticity operator (5.13) are symmetry

$$a(w, v; \mu) = a(v, w; \mu), \ \forall w, v \in X^2, \ \forall \mu \in \mathcal{D} , \tag{6.1}$$

and bi-linearity

$$a(u + \lambda v, w; \mu) = a(u, w; \mu) + \lambda a(v, w; \mu) \quad \lambda \in I\!R, \ \forall u, v, w \in X^3, \ \forall \mu \in \mathcal{D} . \tag{6.2}$$

In addition , we require uniform continuity

$$|a(w, v; \mu)| \leq \gamma \|w\|_Y \|v\|_Y, \quad \gamma > 0, \ \forall w, v \in X^2, \ \forall \mu \in \mathcal{D} , \tag{6.3}$$

and coercivity

$$\alpha \|v\|_Y^2 \leq a(v, v; \mu), \quad \alpha > 0, \ \forall v \in X, \ \forall \mu \in \mathcal{D} . \tag{6.4}$$

Extension to non-symmetric and/or non-coercive problems are discussed in [10, 13]. Note that the requirements above imply uniqueness and existence of the solution.

6.1.2 Separability of the Linear Elasticity Operator

In order to apply the *black-box* method to compute the outputs, the operator also has to be affinely dependent on the parameters of the problem. We use the affine dependence to obtain a decomposition with the general form

$$a(w, v; \mu) = \sum_{q=1}^{Q} \sigma^q(\mu) a^q(w, v), \ \forall w, v \in X^2, \ \forall \mu \in \mathcal{D} \ , \tag{6.5}$$

in which the operators $a^q(.,.)$ are independent of the problem parameters and the scalars $\sigma^q(\mu)$ contain the parameter dependence. The finite number Q is dependent on the original operator $a(.,.;\mu)$ as well as on the specific problem treated.

More specifically, the affine decomposition assumption can be verified for the linear elasticity operator $a(.,.;\mu)$ given by equation (5.13). The only quantity in $a(.,.;\mu)$ that depends on the parameter set μ is the modified elasticity tensor $E^r_{il'jl'}(\mu)$, which has $R \times 2^4$ entries [1]. Each entry corresponds to a scalar $\sigma^q(\mu)$ in equation (6.5). [2]

For simplicity, we introduce a mapping $\mathcal{Q} : \{1,2\}^4 \times \{1 \dots R\} \rightarrow \{1 \dots Q\}$ and we then have

$$a^{\mathcal{Q}(i,j,k,l,r)}(w, v) = \int_{\Omega^r} \frac{\partial w_i}{\partial x_j} \frac{\partial v_k}{\partial x_l} \ , \quad \forall w, v \in X \ , \quad r = 1, \dots, R \ , \quad i, j, k, l = 1, 2 \ . \tag{6.6}$$

The corresponding parameter-dependent coefficient is then the entry of the modified elasticity tensor

$$\sigma^{\mathcal{Q}(i,j,k,l,r)}(\mu) = E^r_{ijkl} \ , \quad r = 1, \dots, R \ , \quad i, j, k, l = 1, 2 \ . \tag{6.7}$$

Remember that the modified elasticity tensor contains all material *and* geometric properties, see Section 4.5.

Note that no other assumption is needed for the $a^q(.,.)$, $q = 1, \dots, Q$, than con-

[1] Remember that R is the total number of parameter regions of the microtruss problem, see section 4.4.

[2] Note also that the modified elasticity tensor has symmetries and (problem specific) zero entries that allow to considerably lower the number of independent entries for any parameter set $\mu \in \mathcal{D}$.

tinuity and bi-linearity.

6.2 Reduced-Basis Approximation

The finite element space X_h defined by (5.9) is a finite but very high dimensional subspace of the continuous space X. We choose a discretization refined enough to ensure that the solution in the high-dimensional "truth" space X_h is in good agreement with the exact solution in X.

The underlying idea of the reduced-basis method is a projection onto a low-order approximation space comprising solutions of the problem of interest at selected points in the parameter space.

6.2.1 Low-Order Space and Discrete Equations

We introduce a sample set $S^N = \{\mu^1, \ldots, \mu^N\}$ in the parameter space $\mathcal{D}$ defined in section 2.2.2. For each input vector in S^N, we calculate using the finite element method a solution $u_h(\mu^n)$ in the truth space X_h.

We then introduce the reduced-basis space as

$$W^N = \text{span}\{u_h(\mu^1), u_h(\mu^2), \ldots, u_h(\mu^N)\} \; . \tag{6.8}$$

To simplify the notation, we define $\zeta^n \in X_h$ as

$$\zeta^n \equiv u_h(\mu^n), \quad n = 1, \ldots, N;$$

we can then write $W^N = \text{span}\{\zeta^n,\ n = 1, \ldots, N\}$. The later means that W^N consists of all functions in X_h that can be expressed as a linear combination of the ζ^n; that is, any member v^N of W^N can be represented as

$$v^N = \sum_{n=1}^{N} \beta^n \zeta^n, \tag{6.9}$$

for some unique choice of $\beta^n \in I\!R$, $n = 1, \ldots, N$. We implicitly assume that the ζ^n, $n = 1, \ldots, N$, are linearly independent; it follows that W^N is an N-dimensional subspace of X.

In the reduced-basis approach we look for an approximation $u^N(\mu)$ to $u_h(\mu)$ in W^N — for our purposes here we presume that $u_h(\mu)$ is *sufficiently close* to $u(\mu)$. In particular, we express $u^N(\mu)$ as

$$u^N(\mu) = \sum_{n=1}^{N} u_n^N \zeta^n; \tag{6.10}$$

we denote by $\underline{u}^N(\mu) \in I\!R^N$ the coefficient vector $(u_1^N, \ldots, u_N^N)^T$. The premise is that we should be able to accurately represent the solution at some new point in parameter space, μ^d, as an appropriate linear combination of *solutions previously computed* at a small number of points in parameter space (the μ^n, $n = 1, \ldots, N$). [3]

The appropriate linear combination coefficients, $\underline{u}^N(\mu) \in I\!R^N$, are obtained by solving the linear elasticity problem in the space W^N. Again we apply the Galerkin projection to obtain the statement

$$a(u^N(\mu), v; \mu) = \ell(v) \;, \quad \forall v \in W^N \;. \tag{6.11}$$

6.2.2 Matrix Form

We can now choose the test functions to be the same as the basis functions

$$v = \zeta^n \;, \quad n = 1, \ldots, N \;, \tag{6.12}$$

and insert the expressions (6.10) and (6.12) into the reference-domain weak form (4.11) to obtain the algebraic representation of (6.11) that is used to calculate $\underline{u}^N(\mu) \in$

[3] Indeed, exponential convergence in N is obtained for sufficiently smoothed μ-dependence (e.g. [5, 9]).

$\mathbb{R}^N$,

$$\sum_{j=1}^{N} a(\zeta^j, \zeta^i; \mu)\, u_j^N = \ell(\zeta^i)\ , \quad \forall\, i = 1, \ldots, N. \tag{6.13}$$

Reduced-Basis Matrix

At this point, we can see that the entries of the reduced-basis matrix $\underline{A}^N(\mu) \in \mathbb{R}^{N\times N}$ can be obtained from

$$A^N_{\alpha\,\beta}(\mu) = a(\zeta^\beta, \zeta^\alpha; \mu)\ , \quad \forall \alpha, \beta \in \{1, \ldots, N\}. \tag{6.14}$$

Since ζ^α is the finite element solution for a particular configuration μ^α, we can write ζ^α in term of the nodal basis functions φ_i, $i = 1, \ldots, \mathcal{N}$ ($\mathcal{N}$ is the dimension of the finite element space),

$$\zeta^\alpha = \sum_{i=1}^{\mathcal{N}} \zeta_i^\alpha \varphi_i\ . \tag{6.15}$$

Then each element of the reduced-basis matrix $\underline{A}^N(\mu)$ is calculated by substituting (6.15) in expression (6.14)

$$A^N_{\alpha\,\beta}(\mu) = a(\sum_{i=1}^{\mathcal{N}} \zeta_i^\alpha \varphi_i, \sum_{j=1}^{\mathcal{N}} \zeta_j^\beta \varphi_j; \mu)\ , \quad \forall \alpha, \beta \in \{1, \ldots, N\}\ , \tag{6.16}$$

and using the bi-linearity of $a(.,.;\mu)$, we obtain

$$A^N_{\alpha\,\beta}(\mu) = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} \zeta_i^\alpha \zeta_j^\beta\, a(\varphi_i, \varphi_j; \mu)\ , \quad \forall \alpha, \beta \in \{1, \ldots, N\}\ . \tag{6.17}$$

To rewrite the above as a product of vectors and matrices, we define $\underline{\zeta}^\alpha$ as the vector of nodal values of the finite element solution for a parameter set μ^α. Also remember that $\underline{A}^h(\mu)$ is the finite element matrix generated in Section 5.2.2. Each element of the reduced-basis matrix is then computed as

$$A^N_{\alpha\,\beta}(\mu) = (\underline{\zeta}^\alpha)^T \underline{A}^h(\mu)\, (\underline{\zeta}^\beta)\ , \quad \forall\, \alpha, \beta \in \{1, \ldots, N\}\ . \tag{6.18}$$

The above can be written succinctly in terms of $\underline{Z}$, an $\mathcal{N} \times N$ matrix which columns are the vector of nodal values of the finite element solution $\underline{\zeta}^\alpha$, $\alpha = 1, \ldots, N$,

$$\underline{A}^N(\mu) = \underline{Z}^T \underline{A}^h(\mu) \underline{Z} \,. \tag{6.19}$$

For an efficient implementation of the reduced-basis solution, this operator will need to be formed using the black-box approach.

Black-Box Approach

Of course, the computation of the reduced-basis operator according to expression (6.19) would be very inefficient, since it requires the formation of a new parameter-dependent finite element matrix for each new evaluation in the parameter space. The big computational gain then comes from using the decomposition (6.5) to create a set of parameter-independent operators in the space W^N that are computed only once and can be reassembled to obtain the reduced basis operator.

We now want to use the separability of the linear elasticity operator, $a(.,.;\mu) = \sum_{q=1}^{Q} \sigma^q(\mu) a^q(.,.)$. First, we verify that a similar decomposition is valid for the corresponding finite element operator. Observe that

$$\begin{aligned} A^h_{ij}(\mu) &= a(\varphi_j, \varphi_i; \mu) && (6.20) \\ &= \sum_{q=1}^{Q} \sigma^q(\mu) a^q(\varphi_j, \varphi_i) && (6.21) \\ &= \sum_{q=1}^{Q} \sigma^q(\mu) A^{h,q}_{ij} \,; && (6.22) \end{aligned}$$

where $A^{h,q}_{ij}$ is an entry of the matrix $\underline{A}^{h,q}$ that is obtained by a direct stiffness assembly similar to the one used for $\underline{A}^h$ in Section 5.2.2. In the particular case of the microtruss plate, each matrix $\underline{A}^{h,q}$ $(q = 1, \ldots, Q)$ is formed by direct stiffness assembly restricted to the elements inside the parameter region q and according to $A^{h,q}_{ij} = a^q(\varphi_j, \varphi_i)$, with $a^q(\varphi_j, \varphi_i)$ defined by (6.6).

We use the separability of the linear elasticity operator, $a(.,.;\mu) = \sum_{q=1}^{Q} \sigma^q(\mu) a^q(.,.)$,

to decompose the algebraic expression (6.13) into the sum

$$\sum_{q=1}^{Q}\sum_{j=1}^{N}\sigma^q(\mu)\,a^q(\zeta^j,\zeta^i)\,u_j^N=\ell(\zeta^i)\;,\quad\forall\,i=1,\ldots,N. \tag{6.23}$$

Now we use the exact same steps as equations (6.14)-(6.18) applied to the decomposed algebraic form (6.23) to obtain the expressions for the black-box computation of the reduced-basis matrix entries

$$\begin{aligned}
A^N_{\alpha\beta}(\mu) &= \sum_{q=1}^{Q}\sigma^q(\mu)\,a^q(\zeta^\beta,\zeta^\alpha) && (6.24)\\
&= \sum_{q=1}^{Q}\sigma^q(\mu)\,a^q\Big(\sum_{i=1}^{\mathcal{N}}\zeta_i^\alpha\varphi_i,\sum_{j=1}^{\mathcal{N}}\zeta_j^\beta\varphi_j\Big) && (6.25)\\
&= \sum_{q=1}^{Q}\sigma^q(\mu)\sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\mathcal{N}}\zeta_i^\alpha\zeta_j^\beta\,a^q(\varphi_i,\varphi_j) && (6.26)\\
&= \sum_{q=1}^{Q}\sigma^q(\mu)(\underline{\zeta}^\alpha)^T\underline{A}^{h,q}(\underline{\zeta}^\beta)\;,\quad\forall\,\alpha,\beta\in\{1,\ldots,N\}\;. && (6.27)
\end{aligned}$$

Now, using the same matrix notations, we see that we must compute Q parameter-independent matrices $\underline{A}^{N,q}\in I\!R^{N\times N}$

$$\underline{A}^{N,q}=\underline{Z}^T\underline{A}^{h,q}\underline{Z}\quad q=1,\ldots,Q\;, \tag{6.28}$$

and we can then reconstruct the reduced-basis operator for any new parameter set by computing

$$\underline{A}^N(\mu)=\sum_{q=1}^{Q}\sigma^q(\mu)\underline{A}^{N,q}. \tag{6.29}$$

Reduced-Basis Inhomogeneity

A similar procedure can be used for the load vector F^N. The steps are outlined below,

$$\begin{aligned} F_\alpha^N &= \ell(\zeta^\alpha) && (6.30)\\ &= \ell(\sum_{i=1}^{\mathcal{N}} \zeta_i^\alpha \varphi_i) && (6.31)\\ &= \sum_{i=1}^{\mathcal{N}} \zeta_i^\alpha \ell(\varphi_i) && (6.32)\\ &= (\underline{\zeta}^\alpha)^T \underline{F}^h , \quad \alpha = \{1, \ldots, N\}. && (6.33) \end{aligned}$$

Therefore

$$\underline{F}^N = \underline{Z}^T \underline{F}^h . \tag{6.34}$$

Reduced-Basis System and Output

We can then write the matrix form of equation (6.13) as

$$\underline{A}^N(\mu)\, \underline{u}^N(\mu) = \underline{F}^N , \tag{6.35}$$

where $\underline{A}^N(\mu) \in \mathbb{R}^{N\times N}$ and $\underline{F}^N \in \mathbb{R}^N$. The matrix $\underline{A}^N(\mu)$ is dense and direct solution methods are employed. Finally, since for the compliance case, $\ell^O(v) = \ell(v)$, we have the output functional

$$\underline{L}^N = \underline{F}^N ,$$

and we get the reduced-basis output s^N as

$$s^N(\mu) = \underline{L}^{N^T} \underline{u}^N(\mu) . \tag{6.36}$$

6.2.3 Off-line / On-line Decomposition

It is computationally extremely efficient to obtain a reduced-basis solution from the system (6.35). The computational cost lies in forming the matrices $\underline{A}^{h,q}$; fortunately, they are parameter-independent and can therefore be computed only once and used

thereon to explore the whole parameter space $\mathcal{D}$. This leads to a two-stage computational approach:

- Off-line stage: compute N finite element solutions, one for each parameter set $\mu \in S^N$. Construct and store the parameter independent reduced-basis matrices $\underline{A}^{N,q} = \underline{Z}^T \underline{A}^{h,q} \underline{Z} \quad q = 1, \ldots, Q$. Construct and store the reduced-basis inhomogeneity $\underline{F}^N = \underline{Z}^T \underline{F}^h$.

- On-line stage: compute the parameter dependent coefficients $\sigma^q(\mu)$ with equation (6.7). Form the reduced-basis matrix $\underline{A}^N(\mu) = \sum_{q=1}^{Q} \sigma^q(\mu) \underline{A}^{N,q}$ and solve the system $\underline{A}^N(\mu)\, \underline{u}^N(\mu) = \underline{F}^N$. Obtain the reduced-basis output $s^N(\mu) = (\underline{L}^N)^T \underline{u}^N(\mu)$.

Note that the two processes are completely decoupled. The expensive off-line computation can be processed at an early stage and needs to be done only once. The efficient on-line computation can then be used for very fast evaluations of outputs at different point in the parameter space.

6.3 Optimality of the Reduced-Basis Approximation

6.3.1 Optimality of the Reduced-Basis Solution

The solution $u^N(\mu)$ obtained in the reduced-basis space W^N is optimal in the energy norm. The proof is similar to the classical proof of the optimality of the finite element solution in the energy norm [2]. In other words, given an input vector μ, the reduced-basis solution is the best possible linear combination of basis functions ζ_n, $n = 1, \ldots, N$ to approximate the exact solution $u(\mu)$ in the energy norm.

In the X-norm, we can prove that

$$\|u(\mu) - u^N(\mu)\|_X \leq \sqrt{\frac{\gamma}{\alpha}} \inf_{w^N \in W^N} \|u(\mu) - w^N\|_X \,, \tag{6.37}$$

which states that the error of the reduced-basis solution is bounded from above by a factor of the *optimal* solution in the space W^N in the X-norm.

Proof. For convenience during this proof, we omit the μ-dependence in the notation of some quantities: $u^N \equiv u^N(\mu)$ and $u \equiv u(\mu)$. The operator $a(.,.;\mu)$ with the properties described in section 6.1.1 can be written for $u - u^N - v^N \in X$ and using bi-linearity and symmetry

$$\begin{aligned} a(u-u^N-v^N, u-u^N-v^N;\mu) &= a(u-u^N, u-u^N;\mu) \\ &\quad -2a(v^N, u-u^N;\mu) + a(v^N, v^N;\mu)\,. \end{aligned}$$

At this point, the Galerkin orthogonality will be useful: the two equations

$$a(u, v;\mu) = \ell(v) \quad \forall v \in X \tag{6.38}$$

$$a(u^N, v;\mu) = \ell(v) \quad \forall v \in W^N \tag{6.39}$$

can be substracted in the space W^N, since $W^N \subset X$. The right hand side is then null and we use bi-linearity on the left hand side to obtain

$$a(u-u^N, v;\mu) = 0\,, \quad \forall v \in W^N\,. \tag{6.40}$$

We use equation (6.40) and the operator coercivity $a(v^N, v^N;\mu) \geq \alpha \|v^N\|_X^2 \geq 0$ to write

$$a(u-u^N-v^N, u-u^N-v^N;\mu) \geq a(u-u^N, u-u^N;\mu)\,. \tag{6.41}$$

For convenience, we use $w^N \equiv u^N + v^N$. Obviously, w^N can be any vector in the reduced-basis space W^N. We then apply continuity as defined by (6.3) to the left side of inequality (6.41) and coercivity to the right side

$$\gamma \|u-w^N\|^2 \geq a(u-w^N, u-w^N;\mu) \geq a(u-u^N, u-u^N;\mu) \geq \alpha \|u-u^N\|_X^2\,,$$

which shows

$$\|u(\mu) - u^N(\mu)\|_X \leq \sqrt{\frac{\gamma}{\alpha}} \inf_{w^N \in W^N} \|u(\mu) - w^N\|_X \ . \tag{6.42}$$

□

6.3.2 Optimality of the Reduced-basis Output

Similarly, the reduced-basis output is optimal in the energy norm. An a priori error estimate of the reduced-basis output also states that it is bounded from above, in the X-norm, by a factor of the optimal output in the reduced-basis space W^N.

Proof. For convenience during this proof, we omit the μ-dependence in the notation of some quantities: $s \equiv s(\mu)$, $s^N \equiv s^N(\mu)$, $e^N \equiv e^N(\mu)$, $u^N \equiv u^N(\mu)$ and $u \equiv u(\mu)$. Consider the difference between the "truth"output and the reduced basis output; by linearity, we have

$$s - s^N = \ell^O(u) - \ell^O(u^N) = \ell^O(u - u^N) \ . \tag{6.43}$$

We also use the fact that our output is compliant[4] , i.e. it is the same linear functional as the inhomogeneity of our problem, to obtain

$$s - s^N = \ell(u - u^N) = a(u, u - u^N; \mu) \ . \tag{6.44}$$

We use the bi-linearity of $a(.,.;\mu)$ to rewrite the previous equation

$$s - s^N = a(\underbrace{u - u^N}_{e^N}, \underbrace{u - u^N}_{e^N}; \mu) + a(u^N, u - u^N; \mu) \ , \tag{6.45}$$

which last term is zero by virtue of the Galerkin orthogonality property (6.40) and the fact that the operator is symmetric and that we have homogeneous Dirichlet boundary conditions. Therefore, we have

$$s(\mu) = s^N(\mu) + a\left(e^N(\mu), e^N(\mu); \mu\right) \ . \tag{6.46}$$

[4] *Non compliant* linear output functionals are treated in [7].

The corollary of the above result, continuity (6.3) and reduced-basis solutions optimality relation (6.42) is that the reduced-basis output is also bounded from above,

$$|s(\mu) - s^N(\mu)| \leq \frac{\gamma^2}{\alpha} \left(\inf_{w^N \in W^N} \|u(\mu) - w^N\|_X \right)^2 . \tag{6.47}$$

□

6.4 Performance of the Reduced-basis Method

To consider the reduced-basis method performance, we focus on the computational cost on one hand and on the accuracy of the results compared to the results obtained in the "truth" space on the other hand.

6.4.1 Computational Cost

The computational cost consists of a fixed cost, the off-line part, and a variable cost for any subsequent evaluation, the on-line part. To evaluate the computational costs, we will use three numbers:

- $\mathcal{N}$: the dimension of the truth space, typically $O(10^4)$ for the two-dimensional microtruss plate problem.
- N: the reduced-basis order, i.e. the number of "truth" space solutions used for the creation of the reduced-basis; typically $O(10)$.
- Q: the number of parameter independent operators. For the microtruss problem, Q is $O(10^2)$.

Fixed cost

In the off-line part, we have to compute N fine solutions; for each solution, $O(\mathcal{N}^\kappa)$ operations are required; where $\kappa \leq 3/2$ for the conjugate gradient method without pre-conditioner. We then have to form Q parameter-independent operators $a^q(.,.)$

and multiply them by the matrix $\underline{Z} \in I\!R^{\mathcal{N} \times N}$ which contains the solutions vectors; this has a cost of $O(Q\mathcal{N}N^2)$.

The total "fixed" cost in the off-line part is $O(\mathcal{N}^\kappa N + Q\mathcal{N}N^2)$.

Variable cost

During the on-line computation, we just need $O(N^2Q)$ operations to add together the Q parameter-independent reduced-basis matrices of size $N \times N$. We then perform $O(N^3)$ operations to invert the resulting matrix in order to obtain the reduced-basis solution.

The total variable cost for each new result is very efficient: $O(N^2Q + N^3)$. This represents an extremely low cost for each new evaluation in the parameter space.

6.4.2 Convergence Rate

By convergence of the reduced-basis, we mean how close the reduced-basis output value is to the "truth" space output value, as a function of the reduced-basis dimension N.

The exponential convergence rate for the output, mentioned in earlier works [10], is observed in Figure 6-1. Note that, although the heat transfer fin problem treated in [2, 7] has the same number of input parameters as the microtruss plate problem, the later requires a higher reduced-basis dimension to reach a satisfying accuracy.

A crucial aspect of reduced-basis methods is that although the error in the output has an exponential convergence with the reduced basis dimension N, it is also highly dependent on the choice of S^N during the creation of the space W^N. [5] In effect, some of the parameter sets chosen for Figure 6-1 show a much more important relative error than others. The error and the rate of convergence in our results is of course dependent on how close the evaluation parameter set is to some of the parameter sets used for the basis functions. It is interesting to notice that the biggest errors are obtained near the lower boundaries of the parameter space: the set μ^9 is very close

[5] Presently, the points are selected randomly in the parameter space. Studies of the distribution of the sample parameter sets are discussed in Chapter 9.

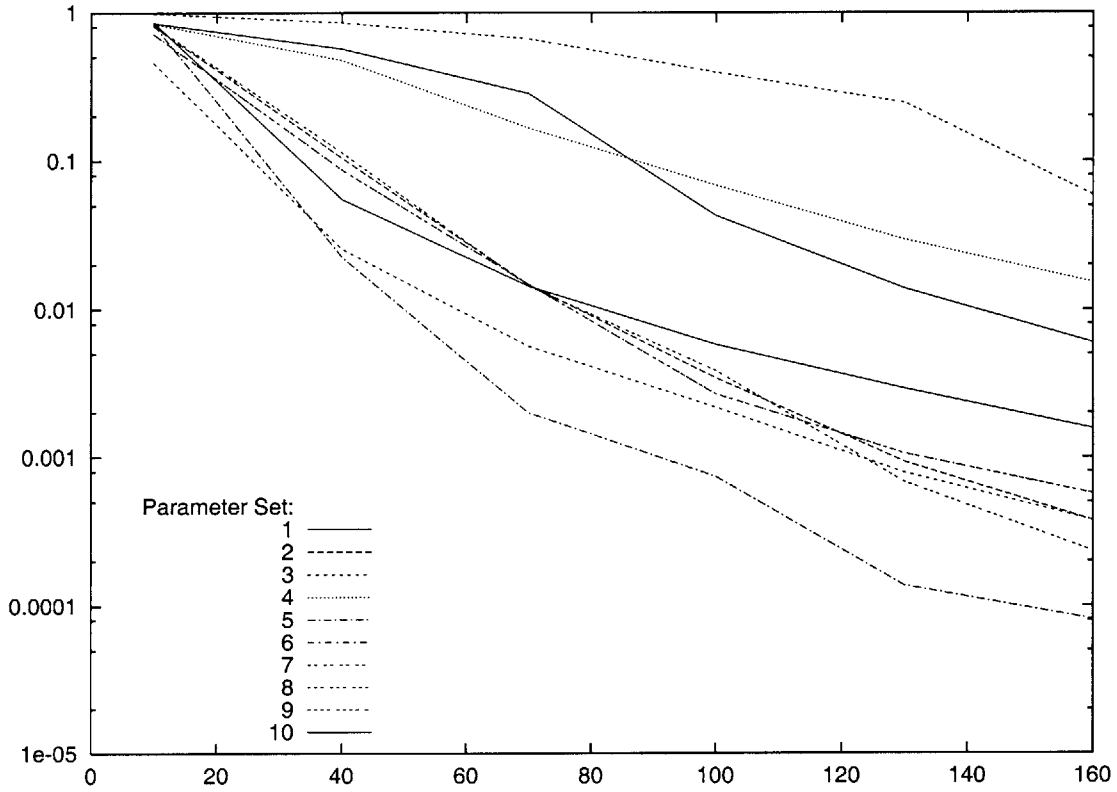

Figure 6-1: Relative error induced by the reduced-basis on the output, for 10 parameter sets, as a function of the reduced-basis order, N=10,...,160.

	μ^1	μ^2	μ^3	μ^4	μ^5	μ^6	μ^7	μ^8	μ^9	μ^{10}
α	0.3	0.7	0.1	1.1	0.35	0.5	0.4	0.9	1.1	0.35
t_{truss}	2.3	1.2	2.8	2.3	0.82	1.3	1.2	2.8	2.3	3.2
S_{y}	25.6	12.5	7.0	4.8	5.1	5.6	12.5	7.0	7.8	5.1
t_{top}	0.5	2.1	3.0	0.5	4.0	1.5	2.1	3.0	2.5	4.0
t_{bot}	2.5	1.9	4.0	2.5	4.0	2.5	5.9	0.8	0.5	4.0
$E_{\text{frame}}/E_{\text{truss}}$	2.3	2.0	12	0.3	34	4.5	45	23	0.09	1

Parameter sets used in Figure 6-1.

to the lowest admissible values for t_{bot} (0.4) and $E_{\text{frame}}/E_{\text{truss}}$ (0.05) and has a much slower convergence rate than other sets. The log-random distribution implemented in [11] will address this issue in more details.

Obtaining a good estimate of the error for each evaluation would allow to insert relevant additional modes in the reduced-basis in order to get sharp results for every estimation.

The computation of the error itself is of course a costly process since it requires a X^h-space approximation. For this reason, we develop in the next chapter an *a posteriori* error estimation procedure that will give rigorous bounds for the output of interest.

Chapter 7

Output Bounds Methods

7.1 Motivation

We have applied in Chapter 6 the blackbox reduced-basis method to the microtruss plate problem. In a design context, where many output estimations in the parameter space are needed, the reduced-basis method is an extremely efficient approximation with optimality properties in the reduced-basis space.

However, the solutions accuracy relative to the high dimensional finite element space is highly dependent on the choice of the reduced-basis, i.e. its dimension and the choice of the sample set S^N in the parameter space.

In order to use the reduced-basis method for design purposes, we present in this chapter an a posteriori error estimation procedure — output bounds — that is both reliable and efficient. With this new tool, multi-parameter designs can be performed using the reduced-basis method in the most efficient manner and with a certified accuracy of the results.

In Section 7.2, we introduce the operator properties that will be needed for the error estimation procedure. Section 7.3 presents the $O(Q^2)$ output bounds method, our most rigorous method. In Section 7.4, we relax some of our requirements on the operator properties and we obtain a very efficient error estimation procedure, the $O(Q)$ method.

7.2 Operator and Output Properties

We require that our operator satisfies the same properties described in Section 6.1. Namely, the operator $a(.,.;\mu)$ is symmetric (6.1), bi-linear (6.2), continuous (6.3) and coercive (6.4).

7.2.1 Lower Bound Operator

In addition to the properties above, we will require for the $O(Q^2)$ output bounds method that a lower bound operator can be identified. In particular, we want to identify a parameter-independent operator $\hat{a}(.,.)$ and a parameter-dependent scalar $g(\mu)$ such that

$$c\|v\|_X^2 \leq g(\mu)\hat{a}(v,v) \leq a(v,v;\mu), \quad \forall v \in X,\ \forall \mu \in \mathcal{D}\ . \tag{7.1}$$

We require that the operator $\hat{a}(.,.)$ is coercive, symmetric and bi-linear.

In this work, we will present results for $\hat{a}(.,.) = a(.,.;\hat{\mu})$, where $\hat{\mu} \in \mathcal{D}$ is a fixed set of parameters.

Note that an alternative statement to define the scalar $g(\mu)$ is obtained by identifying the ratio $\frac{a(v,v;\mu)}{\hat{a}(v,v)}$ as a generalized Rayleigh quotient. The factor $g(\mu) \leq \inf_{v\in X} \frac{a(v,v;\mu)}{\hat{a}(v,v)}$ is then in fact a lower bound to the minimum eigenvalue $\lambda_{\hat{a}}^1(\mu)$ of the generalized eigenvalue problem

$$a(u^\lambda(\mu),v;\mu) = \lambda(\mu)\hat{a}(u^\lambda(\mu),v),\ \forall v \in X. \tag{7.2}$$

7.2.2 Compliant Output

We also use some property related to the inhomogeneity and the output. In this chapter, we derive the a posteriori error estimator for the *compliant* case, i.e. an output functional $\ell^O(v)$ equal to the inhomogeneity $\ell(v)$ defined in (3.19).

The results extend to the general case with the introduction of an adjoint (dual) problem, as exposed in [7].

7.3 $O(Q^2)$ Method

The output error bounds most rigorous method is called the $O(Q^2)$ method. We first present in Section 7.3.1 the method used to compute the output error bounds, without giving the proofs at that stage. The analysis of the error bounds is then detailed in Section 7.3.2.

7.3.1 Low Order Space and Discrete Equations

Of course, we still need a sample set of parameters $S^N = \{\mu^1, \ldots, \mu^N\}$ in the parameter space $\mathcal{D}$ defined in Section 2.2.2. The low order space we will use in the $O(Q^2)$ method is defined by a reduced-basis that comprises the truth space solution $u(\mu^n),\ n = 1, \ldots, N$ and the truth space eigenfunctions $u^\lambda(\mu^n),\ n = 1, \ldots, N$ corresponding to the minimum eigenvalue of the problem (7.2). This low order space is denoted W^{2N},

$$W^{2N} = \text{span}\{u(\mu^n), u^\lambda(\mu^n),\ n = 1, \ldots, N\} \equiv \text{span}\{\zeta^n,\ n = 1, \ldots, 2N\}\ . \tag{7.3}$$

We start by obtaining the reduced-basis approximation presented in Section 6.2.1. This means we apply the Galerkin projection in the space W^{2N} to obtain the statement

$$a(u^{2N}(\mu), v; \mu) = \ell(v)\ , \quad \forall v \in W^{2N}\ , \tag{7.4}$$

and we solve for $u^{2N}(\mu)$ the above reduced-basis linear elasticity problem.

We then want to compute $g(\mu)$ defined in equation (7.34). In our case, $g(\mu)$ has to be approximated[1] for every new parameter set $\mu \in \mathcal{D}$. For each new $\mu \in \mathcal{D}$, the reduced-basis approximation of $g(\mu)$ can be obtained by computing the minimum eigenvalue $\lambda_{\hat{a}}^{1,2N}(\mu)$ of the generalized eigenvalue problem

$$a(u^{\lambda,2N}(\mu), v) = \lambda_{\hat{a}}^{2N}(\mu)\, \hat{a}(u^{\lambda,2N}(\mu), v)\,, \quad \forall v \in W^{2N}. \tag{7.5}$$

[1] For some simple problems, $g(\mu)$ is available analytically, e.g. [2]

Note that the minimum eigenvalue is fact the infimum of the Rayleigh quotient $\frac{a(v,v;\mu)}{\hat{a}(v,v)}$. This means that as we compute the minimum eigenvalue in a reduced space $W^{2N} \subset X$, we will approach from above the minimum eigenvalue $\lambda_{\hat{a}}^1(\mu)$ computed in the truth space,

$$\lambda_{\hat{a}}^{1,2N}(\mu) = \inf_{v \in W^{2N}} \frac{a(v,v;\mu)}{\hat{a}(v,v)} \geq \inf_{v \in X} \frac{a(v,v;\mu)}{\hat{a}(v,v)} = \lambda_{\hat{a}}^1(\mu) \ . \tag{7.6}$$

Fortunately, the convergence of the problem (7.5) is exponential and therefore ensures that $\lambda_{\hat{a}}^{1,2N}(\mu)$ has very good accuracy, even for a small dimension of the reduced-basis space, see [3, 5, 9, 10, 6]. Since we need our approximation $g^{2N}(\mu)$ to be a lower bound to $\lambda_{\hat{a}}^1(\mu)$, we will take

$$g^{2N}(\mu) = \beta \lambda_{\hat{a}}^{1,2N}(\mu) \ , \tag{7.7}$$

with β a safety factor. Empirically, taking $\beta = 0.5$ ensures that $g^{2N}(\mu) \leq \lambda_{\hat{a}}^1(\mu)$ will always hold.

Finally, in order to estimate the error of the reduced-basis approximation relative to the truth space, we compute $\hat{e}(\mu) \in X$, solution of

$$g^{2N}(\mu)\, \hat{a}(\hat{e}(\mu), v) = \ell(v) - a(u^{2N}(\mu), v; \mu) \ , \quad \forall v \in X \ , \tag{7.8}$$

where $g^{2N}(\mu)\, \hat{a}(\hat{e}(\mu), v)$ is the lower bound operator described in Section 7.2.1. We then evaluate the lower bound

$$s_-^{2N}(\mu) = s^{2N}(\mu) \tag{7.9}$$

and the upper bound

$$s_+^{2N}(\mu) = s^{2N}(\mu) + \Delta^{2N}(\mu) \ , \tag{7.10}$$

where $\Delta^{2N}(\mu)$ is called the bound gap and has the following value:

$$\Delta^{2N}(\mu) = g^{2N}(\mu)\, \hat{a}(\hat{e}(\mu), \hat{e}(\mu)) \ . \tag{7.11}$$

We can then show that the truth space output, $s(\mu)$, lies between our output bounds,

$$s_-^{2N}(\mu) \leq s(\mu) \leq s_+^{2N}(\mu) \; . \tag{7.12}$$

We will prove this result in Section 7.3.2 and we will show in Section 7.3.3 that $\Delta^{2N}(\mu)$ can be computed using quantities in the reduced-basis space W^{2N} only.

7.3.2 Analysis of Reduced-Basis Output Bounds

Reconstructed Error

In order to estimate the error of the reduced-basis approximation relative to the truth space, we observe in that space the residual of the reduced-basis solution.

We define the reduced-basis residual

$$R(v;\mu) \equiv \ell(v) - a(u^{2N}(\mu), v; \mu), \quad \forall v \in X \; ; \tag{7.13}$$

in which $u^{2N}(\mu)$ is the reduced-basis approximation defined by equation (6.10), i.e. the sum of the reduced-basis coefficient times the truth space solutions that constitute the reduced-basis.

The test function v in the residual (7.13) can be any function in the space X. In particular, we can take v as the error between the truth space solution and the reduced-basis approximation, $e^{2N}(\mu) = u(\mu) - u^{2N}(\mu)$:

$$R(e^{2N}(\mu);\mu) = \ell(e^{2N}(\mu)) - a(u^{2N}(\mu), e^{2N}(\mu); \mu) \; . \tag{7.14}$$

We then appeal to $\ell(e^{2N}(\mu)) = a(u(\mu), e^{2N}(\mu); \mu)$ and bi-linearity to get

$$R(e^{2N}(\mu);\mu) = a(e^{2N}(\mu), e^{2N}(\mu); \mu) \; , \tag{7.15}$$

with $e^{2N}(\mu) \in X$.

This is an important intermediate result: in the compliant case, the quantity (7.15) is precisely the error induced by the reduced-basis into the output approximation,

according to (6.46), i.e. we have

$$R(e^{2N}(\mu);\mu) = s(\mu) - s^{2N}(\mu)\ . \tag{7.16}$$

Output bounds

The bound gap $\Delta^{2N}(\mu) = g^{2N}(\mu)\hat{a}(\hat{e}(\mu),\hat{e}(\mu))$ is in fact an efficient way to compute a tight upper bound to $R(e^{2N}(\mu);\mu)$, and therefore provides the output bounds we look for.

Proof of Bounding properties. For convenience during this proof, we omit the μ-dependence in the notation of some quantities: $e^{2N} \equiv e^{2N}(\mu)$ and $\hat{e} \equiv \hat{e}(\mu)$. In equation (7.12), the relation $s_-^{2N}(\mu) \leq s(\mu)$ is obvious from equation (6.46) and the coercivity of the operator $a(.,.;\mu)$.

To prove the right inequality $s(\mu) \leq s_+^{2N}(\mu)$, we start by using bi-linearity

$$\begin{aligned}
\Delta^{2N}(\mu) &\equiv g^{2N}(\mu)\hat{a}(\hat{e},\hat{e}) = g^{2N}(\mu)\hat{a}(\hat{e},\hat{e}-e^{2N}) + g^{2N}(\mu)\hat{a}(\hat{e},e^{2N}) \\
&= g^{2N}(\mu)\hat{a}(\hat{e}-e^{2N},\hat{e}-e^{2N}) + g^{2N}(\mu)\hat{a}(\hat{e},e^{2N}) + g^{2N}(\mu)\hat{a}(e^{2N},\hat{e}-e^{2N}) \\
&= g^{2N}(\mu)\hat{a}(\hat{e}-e^{2N},\hat{e}-e^{2N}) + g^{2N}(\mu)\hat{a}(\hat{e},e^{2N}) + g^{2N}(\mu)\hat{a}(e^{2N},\hat{e}) - g^{2N}(\mu)\hat{a}(e^{2N},e^{2N})
\end{aligned}$$

and symmetry to get

$$\Delta^{2N}(\mu) = g^{2N}(\mu)\hat{a}(\hat{e}-e^{2N},\hat{e}-e^{2N}) + 2g^{2N}(\mu)\hat{a}(\hat{e},e^{2N}) - g^{2N}(\mu)\hat{a}(e^{2N},e^{2N}) \tag{7.17}$$

Now note that if we choose $v = e^{2N}$ in (7.8), we obtain from equation (7.15) that

$$g^{2N}(\mu)\hat{a}(\hat{e},e^{2N}) = a(e^{2N},e^{2N};\mu)\ , \tag{7.18}$$

so that equation (7.17) can be rewritten as

$$\Delta^{2N}(\mu) = g^{2N}(\mu)\hat{a}(\hat{e}-e^{2N},\hat{e}-e^{2N}) + 2a(e^{2N},e^{2N};\mu) - g^{2N}(\mu)\hat{a}(e^{2N},e^{2N})\ . \tag{7.19}$$

We also use the coercivity of the $\hat{a}(.,.)$ operator on the first term and we use on the last term the lower bound property (7.1) of the $g^{2N}(\mu)\hat{a}(.,.)$ operator (for $g^{2N}(\mu)$ close enough to $g(\mu)$). We get

$$\Delta^{2N}(\mu) \geq a(e^{2N}, e^{2N}; \mu) \equiv R(e^{2N}; \mu) \; . \tag{7.20}$$

And since $a(e^{2N}, e^{2N}; \mu) = s(\mu) - s^{2N}(\mu)$ according to equation (6.46), we obtain

$$s^{2N}_{+}(\mu) \equiv s^{2N}(\mu) + \Delta^{2N}(\mu) \geq s(\mu) \; . \tag{7.21}$$

□

Effectivity of the Output Bounds

In order for the output bound $\Delta^{2N}(\mu)$ to be useful, we also need to ensure that it is not only *greater than* but also *sufficiently close to* the actual error $s(\mu) - s^{2N}(\mu) = R(e^{2N}(\mu); \mu)$. We define the effectivity $\eta(\mu)$ as the ratio of the bound gap to the actual error

$$\eta(\mu) = \frac{\Delta^{2N}(\mu)}{s(\mu) - s^{2N}(\mu)} \; . \tag{7.22}$$

We can prove that the effectivity is bounded

$$\eta(\mu) \leq \frac{\gamma}{c} \; ; \tag{7.23}$$

where γ is the continuity constant defined in equation (6.3) and c is the coercivity constant of the operator $g(\mu)\hat{a}(.,.)$ defined in equation (7.1).

Proof. For convenience during this proof, we omit the μ-dependence in the notation of some quantities: $s \equiv s(\mu)$, $s^N \equiv s^N(\mu)$, $e^N \equiv e^N(\mu)$, $u^{2N} \equiv u^{2N}(\mu)$, $e^{2N} \equiv e^{2N}(\mu)$ and $\hat{e} \equiv \hat{e}(\mu)$. Consider equation (7.8) in conjunction with the residual definition (7.13)

$$g^{2N}(\mu)\hat{a}(\hat{e}, v) = \ell(v) - a(u^{2N}, v; \mu) \, , \quad \forall v \in X \; , \tag{7.24}$$

where the inhomogeneity value is $\ell(v) = a(u(\mu), v; \mu)$ for all $v \in X$. We use bilinearity of the $a(.,.;\mu)$ operator and we take $v = \hat{e}$ to get

$$g^{2N}(\mu)\hat{a}(\hat{e},\hat{e}) = a(e^{2N},\hat{e};\mu) \; . \tag{7.25}$$

Now we introduce two norms: $\|.\|_{\hat{a}}$ is the norm induced by the SPD operator $\hat{a}(.,.)$ and $\|.\|_a$ is the norm induced by the SPD operator $a(.,.;\mu)$. We apply those norms to equation (7.25) and use the Cauchy-Schwartz inequality on the right hand side to obtain

$$g^{2N}(\mu)\|\hat{e}\|_{\hat{a}}^2 \leq \|e^{2N}\|_a \|\hat{e}\|_a \; , \tag{7.26}$$

which we rewrite

$$\|\hat{e}\|_{\hat{a}} \leq \frac{\|e^{2N}\|_a \|\hat{e}\|_a}{g^{2N}(\mu)\|\hat{e}\|_{\hat{a}}} \; . \tag{7.27}$$

We can modify (7.27) using the continuity property (6.3) in the form $\|\hat{e}\|_a \leq \gamma^{\frac{1}{2}}\|\hat{e}\|_X$ and the coercivity of operator $\hat{a}(.,.)$ — i.e. the right hand inequality in (7.1) for $g^{2N}(\mu)$ close enough to $g(\mu)$ — in the form $\|\hat{e}\|_{\hat{a}} \geq \left(\frac{c}{g^{2N}(\mu)}\right)^{\frac{1}{2}} \|\hat{e}\|_X$ to obtain

$$\|\hat{e}\|_{\hat{a}} \leq \left(\frac{\gamma}{g^{2N}(\mu)c}\right)^{\frac{1}{2}} \|e^{2N}\|_a \; . \tag{7.28}$$

The result (7.28) constitutes our proof. We just have to rewrite it as

$$g^{2N}(\mu)\|\hat{e}\|_{\hat{a}}^2 \leq \frac{\gamma}{c}\|e^{2N}\|_a^2 \; ,$$

remember that the bound gap is defined by (7.11) as $\Delta^{2N}(\mu) = g^{2N}(\mu)\|\hat{e}\|_{\hat{a}}^2$ and use the norm definition $\|e^{2N}\|_a^2 = a(e^{2N}, e^{2N}; \mu)$ to obtain

$$\Delta^{2N}(\mu) \leq \frac{\gamma}{c}\, a(e^{2N}, e^{2N}; \mu) \; . \tag{7.29}$$

To get the upper bound on the effectivity as stated by equation (7.23), simply remember that the error induced by the reduced-basis on the output is $|s(\mu) - s^{2N}(\mu)| = a(e^{2N}, e^{2N}; \mu)$ according to (6.46). □

7.3.3 Numerical Algorithm: Black Box Approach

For the type of operator described in Section 7.2, both the reduced-basis and the output bounds methods can be decomposed into two stages. The *off-line* stage computes all the computationally expensive quantities — i.e. the quantities related to the truth space X. We make sure that all the quantities computed off-line are parameter-independent by using the operator affine dependence on the parameters described in Section 6.1.2. The *on-line* stage then computes the output and error bounds for each new desired design point μ^d. The quantities used in the on-line stage are related exclusively to the low order space W^{2N} and therefore induce only very inexpensive computations. This dual stage approach is called the *black-box* approach.

We will now give a clearer idea of the discrete quantities and then describe step by step the on-line and the off-line procedures. The discrete form of the bound gap, $\Delta^{2N}(\mu)$ given by (7.11), is obtained by putting together all the information we already have. The first step is to expand the terms of equation (7.8) by using the operator decomposition (6.5) and the reduced-basis approximation equation (6.10),

$$\hat{a}(\hat{e}(\mu), v) = \frac{1}{g^{2N}(\mu)} \left(\ell(v) - \sum_{q=1}^{Q} \sum_{j=1}^{2N} \sigma^q(\mu) u_j^{2N}(\mu) a^q(\zeta^j, v) \right), \quad \forall v \in X . \tag{7.30}$$

From linear superposition, we obtain

$$\hat{e}(\mu) = \frac{1}{g^{2N}(\mu)} \left(\hat{z}_0 + \sum_{q=1}^{Q} \sum_{j=1}^{2N} \sigma^q(\mu) u_j^{2N}(\mu) \hat{z}_j^q \right) ; \tag{7.31}$$

where $\hat{z}_0$ satisfies

$$\hat{a}(\hat{z}_0, v) = \ell(v) , \quad \forall v \in X , \tag{7.32}$$

and $\hat{z}_j^q$ satisfies

$$\hat{a}(\hat{z}_j^q, v) = -a^q(\zeta^j, v) \quad \forall v \in X, \, j = 1, \ldots, 2N, \, q = 1, \ldots, Q . \tag{7.33}$$

Finally, the discrete expression for the bound gap just comes from substituting the

expression (7.31) for $\hat{e}(\mu)$ in the bound gap expression $\Delta^{2N}(\mu) = \hat{a}\left(\hat{e}(\mu), \hat{e}(\mu)\right)$:

$$\Delta^{2N}(\mu) = \frac{1}{g^{2N}(\mu)} \left[\underbrace{\hat{a}(\hat{z}_0, \hat{z}_0)}_{c_0} + 2 \sum_{q=1}^{Q} \sum_{j=1}^{2N} \sigma^q(\mu) u_j^{2N}(\mu) \underbrace{\hat{a}(\hat{z}_0, \hat{z}_j^q)}_{\Lambda_j^{2N,q}} + \right.$$
$$\left. \sum_{q=1}^{Q} \sum_{q'=1}^{Q} \sum_{j=1}^{2N} \sum_{j'=1}^{2N} \sigma^q(\mu) \sigma^{q'}(\mu) u_j^{2N}(\mu) u_{j'}^{2N}(\mu) \underbrace{\hat{a}(\hat{z}_j^q, \hat{z}_{j'}^{q'})}_{\Gamma_{jj'}^{2N,qq'}} \right]. \tag{7.34}$$

Note that in equation (7.34), the terms c_0, $\Lambda_j^{2N,q}$ and $\Gamma_{jj'}^{2N,qq'}$ are all parameter independent and can be computed during the same off-line stage that computes the reduced-basis quantities. The online assembly of the bound gap according to (7.34) is then inexpensive. More details about the computational cost are given in Section 7.3.4.

From Section 6.2.2, we are now used to recognize the implemented matrix form of the operators defined above. To clarify, the notations z and $\underline{z}$ will be used; z will denote a function in the space X and $\underline{z}$ will be the vector of the function coefficients in the basis of the space X^h. For our finite element approximation, $\underline{z}$ is the vector of nodal values for the discretization Ω^h used. In particular, the vector $\underline{\hat{z}}_0$ contains the nodal values of the functions $\hat{z}_0$ of (7.32); we define the $\mathcal{N} \times N$ matrices $\underline{\hat{Z}}^q$, $q = 1, \ldots, Q$, which j^{th} column is the vector $\underline{\hat{z}}_j^q$ that contains the nodal values of $\hat{z}_j^q$ solution of (7.33). Also remember that $\underline{A}^h(\mu)$, $\underline{\hat{A}}^h$ and $\underline{A}^{h,q}$, $q = 1, \ldots, Q$, are the $\mathcal{N} \times \mathcal{N}$ finite element matrices corresponding to $a(.,.;\mu)$, $\hat{a}(.,.)$ and $a^q(.,.)$, respectively.

We will now sum up the implementation of the off-line and on-line stages.

Off-line Stage

1. Calculate the finite element solutions $u(\mu^i), i = 1, \ldots, N$ from equation (3.19) and $u^\lambda(\mu^i), i = 1, \ldots, N$ from equation (7.2) to form the reduced-basis space $W^{2N} = \text{span}\{u(\mu^n), u^\lambda(\mu^n),\ n = 1, \ldots, N\} \equiv \text{span}\{\zeta^n,\ n = 1, \ldots, 2N\}$.

2. Form the matrix $\underline{Z}$ which columns are the vector of nodal values of the reduced-basis functions, $\underline{\zeta}^n$, $n = 1, \ldots, 2N$. Compute and store the parameter-independent

matrices $\underline{A}^{2N,q} = \underline{Z}^T \underline{A}^{h,q} \underline{Z}$, $q = 1, \ldots, Q$.

3. Compute and store $\underline{F}^{2N} = \underline{Z}^T \underline{F}^h$.

4. Compute and store the matrix $\hat{\underline{A}}^{2N} = \underline{Z}^T \hat{\underline{A}}^h \underline{Z}$.

5. Compute $\hat{\underline{z}}_0 \in I\!R^{\mathcal{N}}$ from equation (7.32), and $\hat{\underline{z}}_j^q \in I\!R^{\mathcal{N}}$, $q = 1, \ldots, Q$, from equation (7.33). Form the matrices $\hat{\underline{Z}}^q$, $q = 1, \ldots, Q$ which columns are the vectors $\hat{\underline{z}}_j^q$, $j = 1, \ldots, 2N$.

6. Compute and store the scalar c_0, the Q vectors $\underline{\Lambda}^{2N,q}$, and the Q^2 matrices $\underline{\Gamma}^{2N,pq}$ introduced in equation (7.34),

$$
\begin{aligned}
c_0 &= \hat{\underline{z}}_0^T \hat{\underline{A}}^h \hat{\underline{z}}_0 \,, \\
\underline{\Lambda}^{2N,q} &= \hat{\underline{z}}_0^T \hat{\underline{A}}^h \hat{\underline{Z}}^q \,, \quad q = 1, \ldots, Q \,, \\
\underline{\Gamma}^{2N,pq} &= (\hat{\underline{Z}}^p)^T \hat{\underline{A}}^h \hat{\underline{Z}}^q \,, \quad p, q = 1, \ldots, Q \,.
\end{aligned}
$$

On-line Stage

For each new desired design point $\mu \in \mathcal{D}$ we then compute the reduced-basis prediction and the output bounds based on the quantities computed in the off-line stage:

1. Form the parameter-dependent reduced-basis matrix $\underline{A}^{2N}(\mu) = \sum_{q=1}^{Q} \sigma^q(\mu) \underline{A}^{2N,q}$.

2. Compute the reduced-basis solution $\underline{u}^{2N}(\mu) \in I\!R^{2N}$ from $\underline{A}^{2N}(\mu)\, \underline{u}^{2N}(\mu) = \underline{F}^{2N}$.

3. Calculate the output $s(\mu) = (\underline{L}^{2N})^T \underline{u}^{2N}(\mu)$; where $\underline{L}^{2N} = \underline{F}^{2N}$ since this is a compliance case implementation.

4. Compute $g^{2N}(\mu) = \beta \lambda_{\hat{a}}^{1,2N}(\mu)$ from $\underline{A}^{2N}(\mu) \underline{u}^{2N,\lambda}(\mu) = \lambda_{\hat{a}}^{1,2N}(\mu) \hat{\underline{A}}^{2N} \underline{u}^{2N,\lambda}(\mu)$.

5. Evaluate the bound gap $\Delta^{2N}(\mu)$ from (7.34) and the bounds $s_-^{2N}(\mu) = s^{2N}(\mu)$ and $s_+^{2N}(\mu) = s^{2N}(\mu) + \Delta^{2N}(\mu)$.

7.3.4 Results and Issues

Computational Cost

Off-line. The complete off-line stage described step by step in Section 7.3.3 requires $(2NQ+2N+1)$ X^h-linear system solves; $(4N^2(Q^2+1)+2NQ+1)$ $\hat{a}$-inner products; and $2N$ evaluations of linear functionals. It is important to note that the stand-alone reduced-basis method — without the output bounds — accounts for only a small part of the off-line computational cost, namely (N) X^h-linear system solves and (NQ) $\hat{a}$-inner products.

On-line. The complete on-line stage described step by step in Section 7.3.3 requires $O(Q(2N)^2)$ *operations* to reconstruct the parameter-dependent reduced-basis matrix $\underline{A}^{2N}(\mu)$, $O((2N)^3)$ operations to invert it and the bound gap $\Delta^{2N}(\mu)$ is obtained in $O(Q^2(2N)^2)$ operations.

In past references, the $O(Q^2)$ method has been applied to heat transfer problems in which $Q \sim O(10)$, see [2, 10, 7]. However, for reasonably complex structures, the linear elasticity operator decomposition (6.5) requires at least $O(10^2)$ parameter-independent operators — i.e. $Q \sim O(10^2)$. In short, providing output bounds makes us solve $2Q$ times more systems in the high-dimensional space X^h than the stand-alone reduced-basis method. The on-line part also requires $O(Q)$ times more operations than the stand alone reduced-basis method. This computational cost constitutes the first impediment to the use of the $O(Q^2)$ output bounds method for linear elasticity problems.

Test Case

For the $O(Q^2)$ method, the results are computed for a simple structure composed of two trusses fixed at one end and submitted to a shear force at the other end; the beam is shown in Figure 7-1. The parameters are the structure vertical thickness and the trusses angle with the horizontal; $\mu^{\text{beam}} = \{t^{\text{beam}}, \alpha^{\text{beam}}\}$. The projected length

of each truss is kept fixed to 1. This case is much simpler than the microtruss plate described in Chapter 2, however it is sufficient to address the limitations of the $O(Q^2)$ output bounds method for linear elasticity problems.

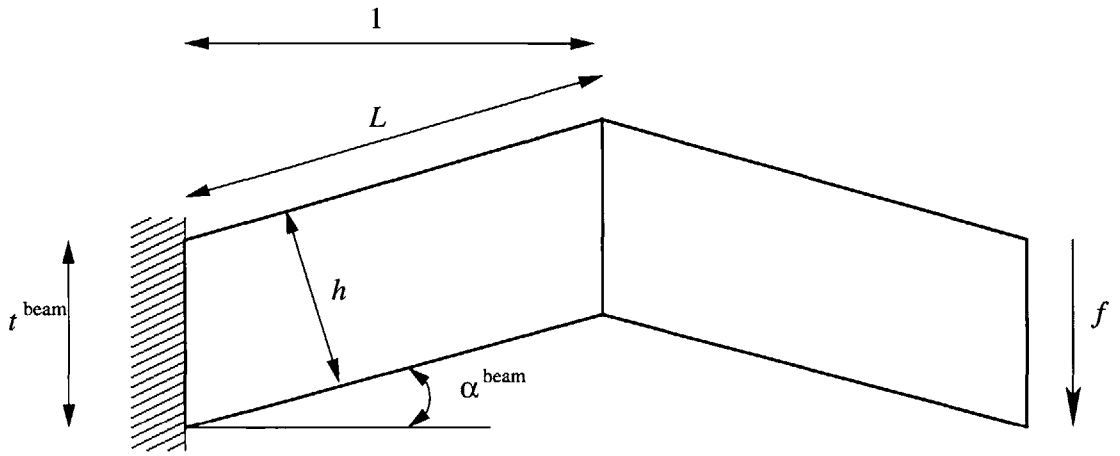

Figure 7-1: Simple structure fixed at one end and submitted to a shear force at the other end.

Effectivity of the Output Bounds

Table 7.1 shows some typical results for the simple beam case. The reduced-basis dimensions chosen 15 and 20, which is enough to get a good accuracy of the output for such a simple case. The effectivities observed are similar for other dimensions of the reduced-basis.

In addition to the effectivities, Table 7.1 gives the aspect ratio of the trusses to the fourth power

$$\left(\frac{L}{h}\right)^4 = \left(\frac{1}{t^{\text{beam}}(\cos\alpha^{\text{beam}})^2}\right)^4 . \tag{7.35}$$

The table also shows the value given by the reduced-basis for the minimum eigenvalue $\lambda_{\hat{a}}^1$ of the generalized eigenvalue problem (7.2). The accuracy of the reduced-basis for $\lambda_{\hat{a}}^1$ has been checked by solving the eigenvalue problem in the high-dimensional space. The results are excellent: within 1% and within 0.1% for the reduced-basis of order 15 and 20 respectively.

The effectivity of the output bounds is not acceptable for case with high aspect ratios. We find a very high correlation between the small value of $\lambda_{\hat{a}}^1$ and the overshooting of the effectivities.

$\mu^{\text{beam}} = \{t^{\text{beam}}, \alpha^{\text{beam}}\}$	$(L/h)^4$	$\lambda_{\hat{a}}^1$	η for $N = 15$	η for $N = 20$
$\mu_1^{\text{beam}} = \{1.42, 0.09\}$	0.25	0.176	3	3
$\mu_2^{\text{beam}} = \{1.81, 0.15\}$	0.09	0.049	8	8
$\mu_3^{\text{beam}} = \{0.77, 0.25\}$	3.62	0.083	11	11
$\mu_4^{\text{beam}} = \{0.83, 0.5\}$	5.92	0.027	60	64
$\mu_5^{\text{beam}} = \{0.55, 0.9\}$	490	0.005	1144	1207

Table 7.1: Effectivity of the $O(Q^2)$ Method for 5 parameter sets. Test case with Dirichlet on one boundary and Neumann on all others.

Note that in cases with all-Dirichlet boundary conditions, the minimum eigenvalue is close to unity and the effectivities are therefore unaffected. In fact, Fourier analysis and variational arguments suggest that the coercivity constant c is order unity for Dirichlet boundary conditions, whereas for Neumann boundary conditions, the distinct rotations induced away from the Dirichlet boundary by $a(.,.;\mu)$ and $\hat{a}(.,.)$ can make the coercivity constant very small [11].

One issue remaining is the choice of the lower bound operator. The results presented above use the operator $\hat{a}(.,.) = a(.,.;\hat{\mu})$ with $\hat{\mu} = \{1.0, 0.3\}$. Various test have been perform to find a more adequate operator, such as other values for $\hat{\mu}$, or use of the Laplacian operator — a special case of the linear elasticity operator. The conclusion was that the choice of operator very unlikely to lower the effectivities by the orders of magnitudes needed for the worst cases.

It appears that for problems with a potentially very small coercivity constant c, the $O(Q^2)$ method rigorous approach — in the sense that the bounds never underestimates the error — can lead to an unacceptable overshooting of the error, indeed equation (7.23) states that the effectivity is only bounded by $\frac{\gamma}{c}$.

7.4 $O(Q)$ Method

The application of the $O(Q^2)$ output bounds method to real-world elasticity problems is obviously compromised by the high computational cost relative to the stand alone reduced-basis method and the high effectivity of the resulting bounds for certain types of boundary conditions.

We now present a relaxation of the $O(Q^2)$ method that lacks its mathematical rigor but performs extremely well in practice. In the $O(Q)$ method, the computational cost of the output bounds is of the same order of magnitude as the reduced-basis computational cost. In addition the effectivity of the output bounds is now bounded from above.

7.4.1 Low Order Space and Discrete Equations

A Reduced Basis for the Error

As in Section 6.2.1, we choose a sample set S^N of parameters $\mu^n \in \mathcal{D}$, $n = 1, \ldots, N$ and construct a low-order space W^N comprising the finite element solutions at the chosen design parameters $W^N = \text{span}\{\zeta^n \equiv u(\mu^n), n = 1, \ldots, N\}$. Again, we compute all the projections in the low-order space W^N of the parameter-independent operators, i.e. $a^q(\zeta^i, \zeta^j)$, $i, j = 1, \ldots, N$ and $\ell(\zeta^i)$, $i = 1, \ldots, N$.

We also choose a second set with *different* sample parameters, $S^{M,err} = \{\mu^{m,err}; m = 1, \ldots, M\}$ in the parameter space $\mathcal{D}$. We then apply *during the off-line stage* the reduced-basis procedure to compute a reduced-basis solution for each parameter vector $\mu^{err} \in S^{M,err}$, i.e we assemble the operator

$$a(\zeta^i, \zeta^j; \mu^{m,err}) = \sum_{q=1}^{Q} \sigma^q(\mu^{m,err})\, a^q(\zeta^i, \zeta^j)\,, \quad i, j = 1, \ldots, N\,, \tag{7.36}$$

and we solve for $u^N(\mu^{m,err})$ the systems

$$a(\zeta^i, \zeta^j; \mu^{m,err})\, u_j^N = \ell(\zeta^i)\,, \quad \forall\, i = 1, \ldots, N\,, \quad m = 1, \ldots, M\,. \tag{7.37}$$

We then construct off-line a reduced-basis space Y^M for the error

$$Y^M = \text{span}\{\xi_m; m = 1, \ldots, M\} \tag{7.38}$$

where $\xi_m \in X$ satisfies the error equation

$$a(\xi_m, v; \mu^{m,err}) = R\left(u^N(\mu^{m,err}), v; \mu^{m,err}\right) , \quad \forall\, v \in X , \quad m = 1, \ldots, M , \tag{7.39}$$

with $R(.;\mu)$ the residual defined by equation (7.13).

As in the case of the solution reduced-basis, we need to compute all the projections in the low-order space Y^M of the parameter-independent quantities, i.e. we compute for each parameter region the quantity

$$a^q(\xi_i, \xi_j) , \quad i, j = 1, \ldots, M , \quad q = 1, \ldots, Q , \tag{7.40}$$

and we compute the reduced-basis right hand side

$$\ell(\xi_i) , \quad i = 1, \ldots, M . \tag{7.41}$$

Output Bounds

During the on-line stage, for each input vector μ^d, we first obtain the reduced-basis solution $u^N(\mu^d)$. In order to compute bounds, we then compute $\check{e}(\mu^d) \in Y^M$, solution of

$$a(\check{e}(\mu^d), v; \mu^d) = R(v; \mu^d) , \quad \forall v \in Y^M . \tag{7.42}$$

We evaluate the lower bound as

$$s_-^{N,M}(\mu^d) = s^N(\mu^d) , \tag{7.43}$$

and the upper bound as

$$s_+^{N,M}(\mu^d) = s^N(\mu^d) + \Delta^{N,M}(\mu^d) , \tag{7.44}$$

with the bound gap

$$\Delta^{N,M}(\mu^d) = \frac{a(\check{e}(\mu^d), \check{e}(\mu^d); \mu^d)}{\beta} \; ; \tag{7.45}$$

where β a security coefficient lesser than 1; typically $\beta = 0.5$.

7.4.2 Analysis of the $O(Q)$ Output Bounds Method

Note that the idea of the $O(Q)$ method come from the rigorous $O(Q^2)$ method of Section 7.3. In fact, the idea is that the reconstructed error $\hat{e}(\mu)$ is replaced by an approximately computed error $\check{e}(\mu)$. This $\check{e}(\mu)$ is obtained by solving the error equation (7.8) in the space Y^M. Following the optimality considerations of the previous chapter, we expect exponential convergence of $\check{e}(\mu)$ to $\hat{e}(\mu)$, i.e. for sufficiently large dimensions M of Y^M we expect that $\check{e}(\mu)$ (and relatedly $\hat{a}(\check{e}(\mu), \check{e}(\mu))$) will be very close to $\hat{e}(\mu)$ (and relatedly to $\hat{a}(\hat{e}(\mu), \hat{e}(\mu))$). Therefore, the bounds $g(\mu)\hat{a}(\hat{e}(\mu), \hat{e}(\mu))$ to the output error $s(\mu) - s^N(\mu)$ should be obtained asymptotically.

This presents an opportunity: choosing $\hat{a}(.,.) \equiv a(.,.;\mu)$, the method can simplify considerably. The operator decomposition property can now be used for the solution of the error equation. Moreover, the small values of $g(\mu)$ that were deteriorating the effectivity no longer exist, since $g(\mu)$ is identically equal to 1 for that particular choice of $\hat{a}(.,.)$.

Effectivity of the Output Bounds

Because of this "relaxation", we cannot prove that the bound gap $\Delta^{N,M}$ will be greater than the error. In fact, we can prove the opposite, i.e. the bound gap approaches the error asymptotically from below — which is why we need the security coefficient β in equation (7.45).

The upside is that we can now prove[2] that the effectivity $\eta^{N,M}$ is bounded from above by a (small) constant;

$$\eta^{N,M} \equiv \frac{\Delta^{N,M}(\mu)}{s(\mu) - s^N(\mu)} \leq \frac{1}{\beta} \, . \tag{7.46}$$

[2]Presently, this proof has only been done for the case of a compliant output.

Proof. As usual we will omit during the proof the μ-dependence of $e^N(\mu)$ and $\check{e}(\mu)$. Remember from equation (6.46) that $s(\mu) - s^N(\mu) = a(e^N, e^N; \mu)$ for the compliant case. We use this in the denominator of (7.46) and we substitute $\Delta^{N,M}(\mu)$ from equation (7.45) in the numerator,

$$\eta^{N,M} = \frac{1}{\beta} \frac{a(\check{e}, \check{e}; \mu)}{a(e^N, e^N; \mu)} \ . \tag{7.47}$$

From equation (7.42) and Cauchy-Schwartz, we get $a(\check{e}, \check{e}; \mu) \leq a(e^N, e^N; \mu)$ and therefore inequality (7.46) holds.

□

7.4.3 Black Box Approach

Using the operator decomposition (6.5) and the reduced-basis approximation equation (6.10), we can rewrite the reduced-basis space error equation (7.42) as

$$\sum_{q=1}^{Q} \sigma^q(\mu) a^q(\check{e}(\mu), v) = \ell(v) - \sum_{q=1}^{Q} \sum_{n=1}^{N} \sigma^q(\mu) u_n^N a^q(\zeta^n, v) \ , \quad \forall v \in Y^M \ , \tag{7.48}$$

which is the system we need to solve for $\check{e}(\mu)$.

Again, we use the matrix form introduced in Section 6.2.2. In particular, we use again the $\mathcal{N} \times N$ matrix $\underline{Z}$ which columns contain the vectors $\underline{\zeta}_j$, $j = 1, \ldots, N$; we define the $\mathcal{N} \times M$ matrix $\underline{\mathcal{X}}$ which columns contains the vectors $\underline{\xi}_j$, $j = 1, \ldots, M$. Also remember that $\underline{A}^h(\mu)$ and $\underline{A}^{h,q}$, $q = 1, \ldots, Q$, are the $\mathcal{N} \times \mathcal{N}$ finite element matrices corresponding to $a(.,.;\mu)$ and $a^q(.,.)$, respectively.

We now sum up the two stages of the black-box approach.

Off-line Stage

1. Calculate the finite element solutions $\zeta^i \equiv u(\mu^i), i = 1, \ldots, N$ from equation (3.19) to form the output reduced-basis space W^N. Form the matrix $\underline{Z} \in I\!R^{\mathcal{N} \times N}$ which n^{th} column is $\underline{\zeta}^n$.

2. Compute and store the matrices $\underline{A}^{N,q} \in I\!R^{N\times N}$ as $\underline{A}^{N,q} = \underline{Z}^T \underline{A}^{h,q} \underline{Z}$, $q = 1,\ldots,Q$.

3. Using the output reduced-basis, calculate $\xi^i, i = 1,\ldots,M$ from equation (7.39) to form the error reduced-basis space. Form the matrix $\underline{\mathcal{X}} \in I\!R^{\mathcal{N}\times M}$ which n^{th} column is $\underline{\xi}^n$.

4. Compute and store $\breve{\underline{A}}^{M,q} \in I\!R^{M\times M}$ as $\breve{\underline{A}}^{M,q} = \underline{\mathcal{X}}^T \underline{A}^{h,q} \underline{\mathcal{X}}$, $q = 1,\ldots,Q$.

5. Compute and store the matrices $\underline{\Upsilon}^{MN,q} \in I\!R^{M\times N}$ as $\underline{\Upsilon}^{MN,q} = \underline{\mathcal{X}}^T \underline{A}^{h,q} \underline{Z}$, $q = 1,\ldots,Q$.

6. Compute and store the vectors $\underline{F}^N = \underline{Z}^T \underline{F}^h$ and $\underline{F}^M = \underline{\mathcal{X}}^T \underline{F}^h$.

On-line Stage

For each new desired design point $\mu \in \mathcal{D}$ we then compute the reduced-basis prediction and error bound based on the parameter-independent quantities computed in the off-line stage:

1. Form $\underline{A}^N(\mu) = \sum_{q=1}^Q \sigma^q(\mu) \underline{A}^{N,q}$ and $\breve{\underline{A}}^M(\mu) = \sum_{q=1}^Q \sigma^q(\mu) \breve{\underline{A}}^{M,q}$.

2. compute the reduced-basis solution $\underline{u}^N(\mu) \in I\!R^N$ from $\underline{A}^N(\mu)\, \underline{u}^N(\mu) = \underline{F}^N$.

3. Calculate the output $s^N(\mu) = (\underline{L}^N)^T \underline{u}^N(\mu)$, with $\underline{L}^N = \underline{F}^N$ since this is a compliant case implementation.

4. Compute $\breve{\underline{e}}(\mu) \in I\!R^M$ from $\breve{\underline{A}}^M(\mu)\, \breve{\underline{e}}(\mu) = \underline{F}^M - \sum_{q=1}^Q \sigma^q(\mu)\, (\underline{u}^N(\mu))^T\, \underline{\Upsilon}^{MN,q}$.

5. Evaluate the bound gap $\Delta^{N,M}(\mu) = \frac{1}{\beta}\, \breve{\underline{e}}(\mu)^T \breve{\underline{A}}^M(\mu)\, \breve{\underline{e}}(\mu)$ and the bounds $s_-^N(\mu) = s^N(\mu)$ and $s_+^N(\mu) = s^N(\mu) + \Delta^{N,M}(\mu)$.

7.4.4 Results and Issues

The relative error induced by the reduced-basis on the output estimates has been studied in Section 6.4.2 (Fig. 6-1). Clearly, the reduced-basis output estimates can benefit from a method that certifies their accuracy in a sharp and efficient manner.

The rigorous $O(Q^2)$ method failed on both accounts for the microtruss plate problem. We will now review the results for the $O(Q)$ method.

Computational Cost

Off-line. The complete off-line stage described step by step in Section 7.4.3 requires $(N + M)$ X^h-linear system solves; $Q(N^2 + M^2 + NM)$ a-inner products; $N + M$ evaluations of linear functionals; and M reduced-basis computations. Given that M is roughly of order N, the $O(Q)$ methods allows to gain a factor Q on the off-line computational cost compared to the $O(Q^2)$ method. This is a very consequent gain in the case of the microtruss plate problem where $Q \sim O(10^2)$.

On-line. The on-line stage cost consists then of QN^2 and QM^2 *operations* to reconstruct the parameter-dependent operators A^N and $\breve{A}$, respectively; and N^3 and M^3 to inverse the dense matrices obtained. Therefore, we also gain a factor Q on the on-line computational cost, compared to the $O(Q)$ method.

To conclude, the $O(Q)$ output bound method computational cost — off-line and online — is very close to the reduced-basis cost. This is critical for practical implementations, where the choice is between getting a*certified accuracy* of the results vs gaining *overall accuracy* by introducing more modes in the basis.

Test Case

We test the $O(Q)$ method on an advanced structural mechanics problem, namely the two-dimensional microtruss plate presented in Chapter 2. We now give a few numbers relevant to the problem.

- $\mathcal{N} = 3.10^4$ — dimension of the X^h-space in order to get a good approximation of the exact solution (within 3%).
- $Q = 240$ — number of parameter-independent operators.

- $N, M \in [50; 200]$ — dimensions of the low-order spaces W^N and Z^N in order to get reasonable accuracies for the reduced-basis output and the output bounds.
- $\beta = \frac{1}{2}$ — the "security" coefficient. Also the upper bound to the effectivity: $\eta \leq 1/\beta$.

Effectivity of the Output Bounds

We will now study the bounds effectivity — the ratio of the bound gap to the actual error — in order to establish how sharp is the $O(Q)$ method.

Figure 7-2 shows the effectivities of the bounds in function of the reduced-basis dimension N for four design point in the parameter space $\mathcal{D}$. The dimension of the "error" low-order space is $M = N^{1.1}$. We observe the asymptotic behavior mentioned in Section 7.4.2: as the dimension N of the output reduced-basis space increases, so does the ratio M/N (since $M = N^{1.1}$) and the effectivity converges asymptotically to $1/\beta$. The downside of the $O(Q)$ method is the pre-asymptotic behavior observed for $N \leq 50$: an effectivity inferior to 1 indicates a bound gap inferior to the error, i.e. false.

The experimental relation between the output bounds effectivity and the dimension M of the "error" reduced-basis is studied in Table 7.2. The dimension of the low-order space Y^M is taken as a power p of the dimension of the output reduced-basis space W^N, $M = N^p$, $p \geq 1$; also, the security coefficient $1/\beta$ is fixed to 2 and the dimension of W^N is 80. We observe that to be on the safe side — and if we do not wish to augment the security coefficient $\frac{1}{\beta}$ — it is preferable to take $p = 1.1$ for the microtruss plate problem. By experience, the optimal power p is problem specific.

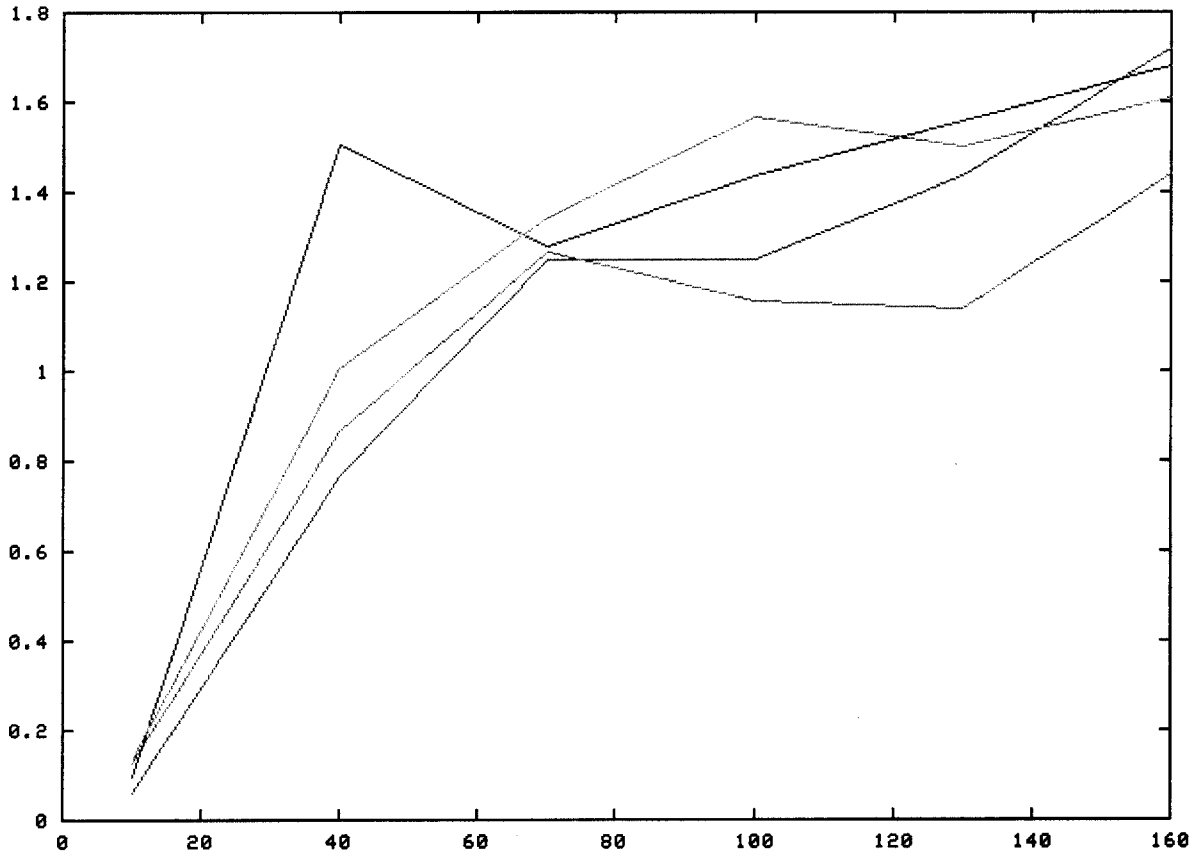

Figure 7-2: Effectivity of the $O(Q)$ Method for 4 parameter sets, as a function of N, $M = N^{1.1}$.

p	Exp(η)	Std(η)	$\eta \geq 1$
1.0	1.21	0.32	72.8 %
1.05	1.53	0.40	88.4 %
1.1	1.75	0.32	98.6 %
1.2	1.92	0.23	98.4 %

Table 7.2: Mean and standard deviation of the effectivity over 500 sets of parameter, for the pre-asymptotic case $N = 80$, $M = N^p, p \in \{1; 1.2\}$. The fourth column shows the percentage of output bounds with effectivities ≥ 1.

Chapter 8

Application: Trade Off Analysis

The reduced-basis method is a very powerful tool for engineering design. The quasi-instantaneous computation time of the outputs allows to explore many configurations in the parameter space.

An example of parameter space exploration for the microtruss plate problem is given in Figure 8-1. Computations have been performed for design configurations over all the parameter space $\mathcal{D}$. Each point shows the computed deflection (and volume) for a particular configuration. The lower left boundary is the Pareto curve on which a minimum structural volume is achieved for a given deflection (or inversely, a minimum deflection is achieved for a given volume).

8.1 Example: the Optimal Storage Structure

We now present a short example of optimal structure design. The case we study is a multi-functional application of the microtruss described in Chapter 2. The idea is to use the reduced-basis method very short computation time to evaluate many design configurations. In addition, the accuracy of the computations is certified with the output bounds $O(Q)$ method.

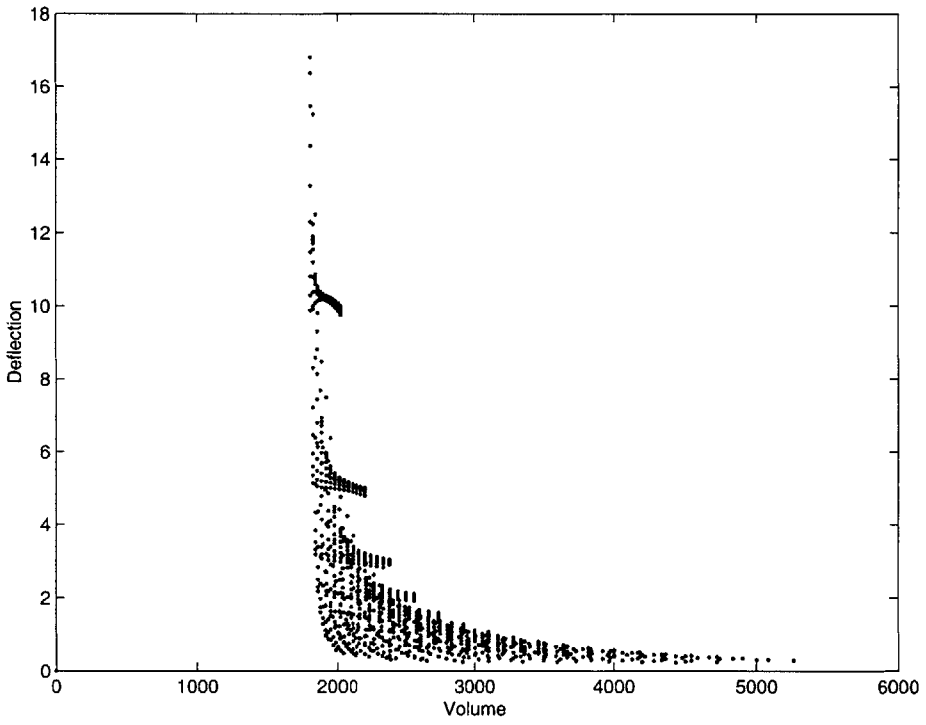

Figure 8-1: Deflection in function of the volume for all values in the parameter space $\mathcal{D}$. The lower left side is the Pareto curve for the truss plate problem.

8.1.1 Storage Structure

Design Requirements

As an example, we choose to design a multi-functional structure that must meet certain stiffness criteria and will be used to store a liquid. In particular, the storage volume inside the prismatic structure, given by $V_{\text{storage}} = S_{\text{y}} \times (L - 2 - t_{\text{side}}) - 13 \times t_{\text{truss}}$, is required to be 1800 units per unit depth. In addition, the structure must be able to carry a load applied along Γ_N without exceeding a maximum deflection. We want to optimize the design in order to meet the storage and deflection requirements while minimizing the structural weight (i.e. volume).

Design Parameters

The top and bottom sheet faces thicknesses are fixed to $t_{\text{top}} = t_{\text{bot}} = 3$ in order to provide a required impact resistance. Also, we take $t_{\text{side}} = 1$ and $\frac{E_{\text{frame}}}{E_{\text{truss}}} = 1.5$.

We vary two design parameters: the vertical spacing S_{y} and the core trusses angle α. Note that the core trusses thickness t_{truss} varies in function of S_{y} in order to keep a constant storage volume,

$$t_{\text{truss}} = \frac{1}{13}\left((L - 2 \times t_{\text{side}}) - \frac{V_{\text{storage}}}{S_{\text{y}}}\right) . \tag{8.1}$$

The parameter space of the storage structure problem is then defined by the intersection of the (more general) microtruss plate parameter space and the parameter variations described above, giving

$$4.0 \leq \quad S_{\text{y}} \quad \leq 60 \tag{8.2}$$

$$0.2 \leq \quad \alpha \quad \leq 1.2 \tag{8.3}$$

$$0.4 \leq \quad t_{\text{truss}} = \tfrac{1}{13}\left(295 - \tfrac{1800}{S_{\text{y}}}\right) \quad \leq 4.0 \, . \tag{8.4}$$

8.1.2 Optimal Design

Reduced-Basis Computations for Design

The reduced-basis very short computation time allows to explore the entire parameter space of the problem, as shown in Figure 8-2. The truss angle α and the vertical spacing S_y vary along the two lower axis. One output, the deflection, is the z-coordinate of the surface, and the other output, the structural volume, is the color of the surface. Once the structural behavior is known for all possible geometries of the

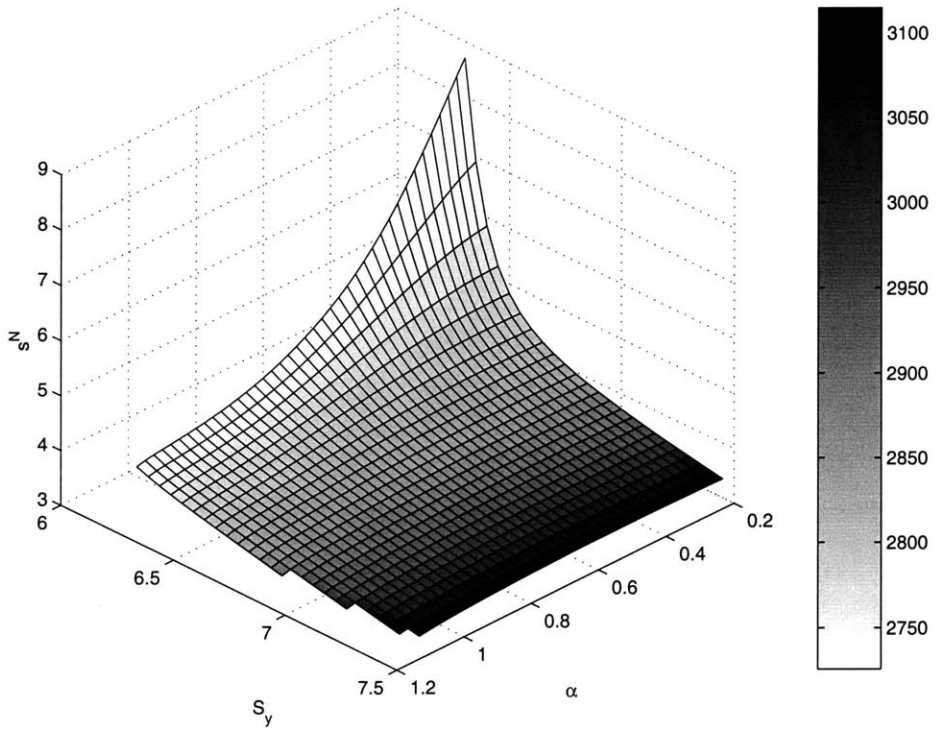

Figure 8-2: Deflection in function of the vertical spacing S_y and the truss angle α. The surface color gives the corresponding structure volume.

structure, an optimal design is easily reached.

Output Bounds Computations

In order to use the reduced-basis methods for design, it is also necessary to predict how accurate they are in comparison with the underlying "truth" space computations.

The output bounds developed in Chapter 7 provide the necessary tools.

In particular, observe Figure 8-3 in which the surface color is this time the error induced by the reduced-basis method into the deflection computation, as predicted by the (OQ) output bounds method. Our structure seems to have an interesting behavior for values of the truss angle α superior to 1.1 radian: the tip deflection starts decreasing again. In fact, the surface color indicates that the error in that region represents up to 15% of the deflection values. This is a non-negligible error that indicates the need for additional basis functions in the reduced space Z^M.

8.2 Actionable Equation

The PDF version of this thesis contains an interactive equation that allows to run the reduced-basis code over the web. To retrieve the PDF version, visit

`http://augustine.mit.edu/`

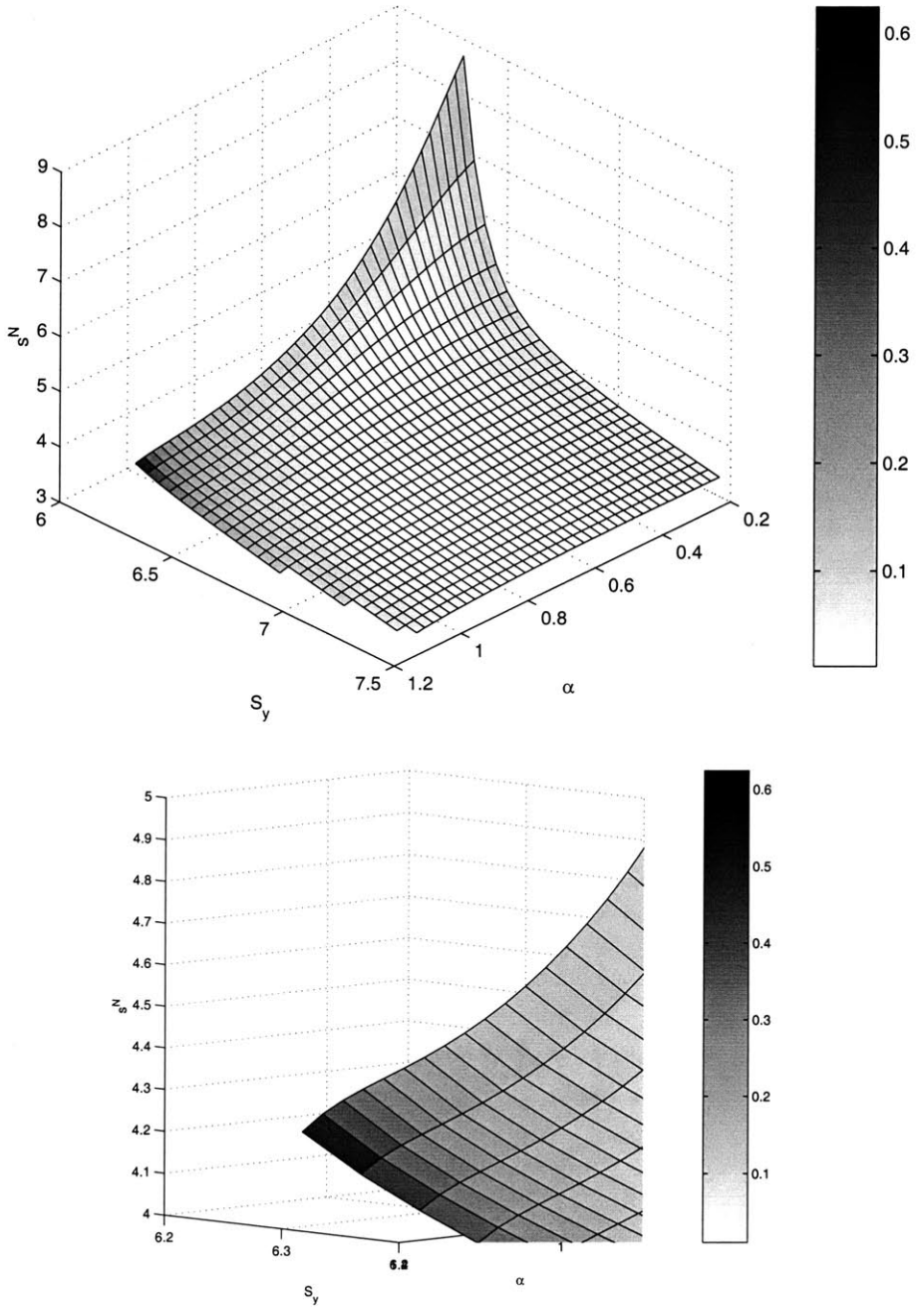

Figure 8-3: Deflection in function of the vertical spacing S_y and the truss angle α. The surface color gives the output error bound $\Delta^{N,M}(\mu)$. The lower figure is a close up of the surface leftmost corner.

Chapter 9

Future Work

9.1 Reduced Basis and Output Bounds Methods Theoretical Extensions

Non-compliant outputs. The output bounds theory presented herein for compliant outputs has already been extended to more general non-compliant outputs [7] for the $O(Q^2)$ method. The $O(Q)$ method is also being extended to non-compliant outputs in [11].

Sample parameter sets distribution. In this thesis, a uniform random distribution has been used for the sample parameter sets S^N; other distributions will be studied, such as log-random. Some theoretical work is in progress to predict an optimal distribution pattern.

Output bounds $O(Q)$ method variants. The $O(Q)$ method can still be improved to reduce the number of system solve off-line as well as the number of operations on-line.

9.2 Microtruss Plate Problem Extensions

Three-dimensional model. More general geometries than prismatic materials can be studied by implementing a three-dimensional model of the microtruss plate. The reduced-basis will be an extremely appropriate method since the number of degrees of freedom of the "truth" space will increase very rapidly for a three-dimensional problem.

Non-compliant outputs. The implementation of the output bounds methods for non-compliant outputs will allow to study more general failure modes in [11], such as:

- Stresses at the junction between the trusses and the sheet faces. This output is related to the fracture failure mode.
- Axial stress in the trusses. This output is related to buckling and yielding.
- Stress at the interface between the sheet faces and the clamped side sheet. For a downward load at the tip, the top sheet face is in tension and therefore the stress at the interface between the top sheet face and the clamped side sheet is related to yielding; the bottom sheet face is in compression and therefore the stress at the interface between the bottom sheet face and the clamped side sheet is related to buckling.

Multi-functionality. The microarchitecture of cellular materials allows for their use in multi-functional applications. Particular applications that will be modeled are:

- heat transfer applications in which a flow is conducted through the core of the structure.
- Actuator problems that rely on thermo-mechanical physics. Shape memory alloys deflect with changes in temperature and can therefore be used as actuators. The model will then involve some coupling between thermal and mechanical response.

Inverse problems. For each new output approximation, the reduced-basis has a very low computational cost. This makes it a very powerful method for multi-parametric problems which require many output evaluations such as design, control and inverse problems.

In particular, [11] will study inverse problems such as finding the position of a crack by analyzing the changes in the vibrations frequencies and eigenmodes of the structure. This problem is governed by the Helmholtz equations.

Bibliography

[1] N.A. Fleck M.F. Ashby A. G. Evans, J.W. Hutchinson and H.N.G. Wadley. The topological design of multifunctional cellular metals. *Progress in Materials Science*, 46:309–327, 2001.

[2] D. Rovas A. T. Patera and L. Machiels. Reduced-basis output-bound methods for elliptic partial differential equations. *SIAG/OPT Views-and-News*, April 2000.

[3] I. Babuska and J. Osborn. Eigenvalue problems. In *Handbook of numerical analysis*, volume II, pages 641–787. Elsevier, 1991.

[4] K. J. Bathe. *Numerical methods in finite element analysis.* Prentice-Hall, 1976.

[5] J. P. Fink and W. C. Rheinboldt. On the error behaviour of the reduced basis technique for nonlinear finite element approximations. *Z. Angew. Math. Mech.*, 63:21–28, 1983.

[6] 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*, to appear.

[7] Y. Maday, L. Machiels, A. T. Patera, and D. V. Rovas. Blackbox reduced-basis output bound methods for shape optimization. In *Proceedings* 12^{th} *International Domain Decomposition Conference*, Japan, 2000. To appear.

[8] A. T. Patera and E. M. Rønquist. *Introduction to Finite Element Method.* 1998. To appear.

[9] T. A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, October 1985.

[10] D. V. Rovas. An overview of blackbox reduced-basis output bound methods for elliptic partial differential equations. In *16th IMACS World Congress*, 2000.

[11] K. P. Veroy. A posteriori error estimation for the reduced basis method applied to problems in elasticity. PhD thesis, MIT, 2002. To appear.

[12] Nathan Wicks and John W. Hutchinson. Optimal truss plates. Technical report, Division of Engineering and Applied Sciences, Harvard University, 2000.

[13] A.T. Patera Y. Maday and D.V. Rovas. A blackbox reduced-basis output bounds approach for non-coercive linear problems. Technical report, MIT-FML, 02 2000. Also in Collège de France series.